4th IEEE Conference on Automation Science and Engineering
Key Bridge Marriott, Washington DC, USA
August 23-26, 2008

Synchronization Performance of the Precision Time Protocol in the Face of Slave Clock Frequency Drift

Ruxandra Lupas Scheiterer, Chongning Na, Dragan Obradovic, Günter Steindl, Franz-Josef Goetz

Abstract— **This paper studies the performance of the Precision Time Protocol (PTP) of the IEEE 1588 standard for drifting slave frequencies. The error expression for the master time estimate at the n^{th} slave is analytically derived and demonstrated in simulation runs. We show that single-slave frequency drift is very benign compared to master frequency drift, which is only matched by all slaves drifting.**

I. Introduction

Many Ethernet-based applications require the networked clocks to be precisely synchronized. The Standard Network Time Protocol (NTP) [1], [2], executed over Ethernet provides synchronization accuracy at the millisecond level, which is appropriate for processes that are not time critical. However, in many applications, for example base station synchronization or motion control, where only sub-microsecond level synchronization errors are allowed, a more accurate synchronization solution is needed. The Precision Time Protocol (PTP), delivered by the IEEE 1588 standard [3] published in 2002 constitutes a promising Ethernet synchronization protocol, in which messages carrying precise timing information, obtained by the hardware time stamping in the physical layer, are propagated in the network to synchronize the slave clocks to a master clock. Boundary clocks adjust their own clock to the master clock and then serve as masters for the next network segment. Authors of [4], [5] introduced the transparent clock (TC) concept, in which intermediate bridges are treated as network components with known delay. By doing this, no control loop in the intermediate element is needed for providing timing information to the next local clock and hence the synchronization at the time client is not dependent on the control loop design in the intermediate bridges. The transparent clock concept has been adopted in the new version of IEEE 1588 published in 2007 (http://ieee1588.nist.gov/: Balloting on IEEE 1588 version 2 began on July 5, 2007).

The current state of the art is to guarantee a synchronization precision of $1\mu s$ for topologies with no more than 30 consecutive slaves. To expand this limit it is important to study the factors that influence the quality of the synchronization process and to find out methods to minimize the effect of detrimental factors.

An important factor that affects the synchronization quality achievable by PTP is the stability of oscillators. Industrial environments are such that unpredictable and independent temperature changes at each node are commonly encountered, causing short-term frequency drifts, unless precluded by expensive temperature compensated (TCXO) or oven controlled (OCXO) crystal oscillators. This possibility is not addressed in the current PTP synchronization protocol, but can introduce non-negligible synchronization errors.

In this paper, we analytically derive the expression for the error introduced by slave frequency drift. The master frequency is assumed to be stable. This scenario is quite realistic, because often the Sync master is an oscillator which is considerably more expensive, in return for a much more stable frequency. In contrast, usually slaves have regular quartzes, which heat or cool with the environment. The formulas allow comparing the effects of pure slave-drift with those of pure master-drift derived in [6], and assist a decision on how to prioritize the deployment of more costly quartzes with improved frequency stability.

The paper is organized as follows. Section 2 introduces the system model and briefly describes the PTP protocol. Analysis of the influence of slave frequency drift on the synchronization accuracy is presented in Section 3. Simulation results are shown in Section 4.

II. Brief Description of PTP with Transparent Clocks

Since the standard doesn't specify the details, this section introduces the system model and notation. Fig. 1 shows a system with $N+1$ cascaded elements connected in a line topology. The PTP has a master/slave structure. The time server, called (grand)master, provides the reference time to the other elements, called slaves, via time-aware bridges (TCs). The master sends Sync messages every T seconds, which carry the counter state of the master clock M^i, stamped at sending, and are propagated along the network.

Quantities, certain or not, linked with the Sync message transmitted by the master at time t_i are labeled by the superscript i. The *line delay* LD_n^i, is the propagation time between the n^{th} slave and its uplink element, and is estimated

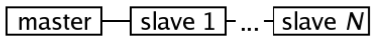
Manuscript received February 29, 2008.

R. L. Scheiterer, C. Na and *D. Obradovic* are with Siemens AG, Corporate Technology, Information and Communications, Munich, Germany (e-mail: {ruxandra.scheiterer, na.chongning.ext, dragan.obradovic }@siemens.com; *corresponding author* phone: +49-89-636 49499; fax: +49-89-636-49767).

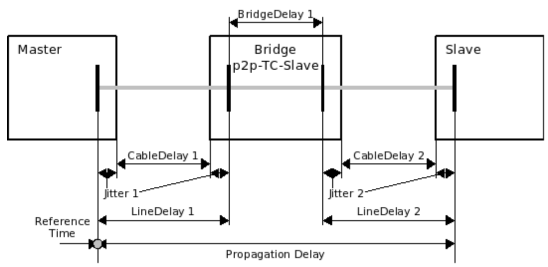
G. Steindl is with Siemens AG, Automation and Drives, Industrial Automation Systems, Amberg, Germany (e-mail: guenter.steindl @siemens.com).

F. J. Goetz is with Siemens AG, Automation and Drives, Advanced Technologies and Standards, Nürnberg, Germany (e-mail: franz-josef.goetz @siemens.com).

by using the line delay estimation process. The Sync message is forwarded after a *bridge delay* BD_n^i, which is recorded at each slave as the difference of the times stamped at reception and forwarding. A time labeled by S_n (resp. M) means "measured in the local time of slave n (resp. master time)"; a hat on a symbol means "estimate". E.g. $\hat{S}_n(BD_n^i)$ means "the bridge delay that affects the i[th] Sync message at slave n, estimated in terms of the own local free-running clock". We define LB_n^i as the sum of line delay plus bridge delay of Sync message i at slave n, and $\delta_{LB}^{i,n} = LB_n^i - LB_n^{i-1}$ to be the difference between the LB values that affected Sync messages i and $i-1$ at slave n. The *latency* L_n^i is the propagation time of the i^{th} Sync message from its initial transmission until its arrival at slave n, with $L_0^i = 0$.



(a) Network topology



(b) System parameters

Fig. 1: System Model

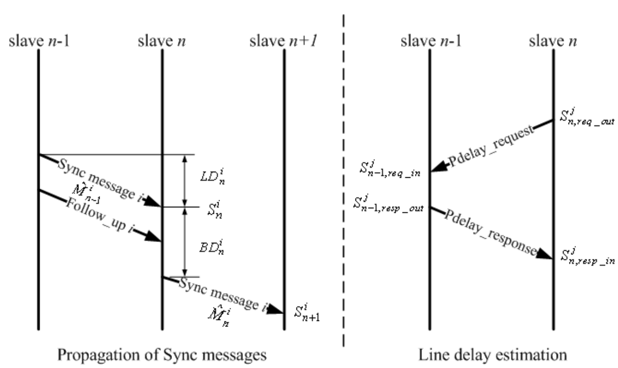


Fig. 2: PTP with transparent clocks

Fig. 2 illustrates the pillars of the time synchronization, the timing propagation and the line delay estimation processes. Slave n updates the (estimated) master counter value packaged in the received Sync message by augmenting it with its local delay, i.e. the sum of its line and bridge delays, translated to master time. Since both are computed in local time, each slave needs to determine its frequency offset to the master. The *rate compensation factor* (RCF, also "rate ratio", [7], [8]) is defined as the frequency ratio of two clocks. We use $RCF_{X/Y}$ to denote the frequency ratio between X and Y, i.e. ideally $RCF_{X/Y} = f_X / f_Y$. The rate offset to the master can be estimated via the master counter estimates in two Sync messages and the local counter values at arrival of these messages at slave n:

$$RCF_{M/S_n} = \frac{\hat{M}_{n-1}^i - \hat{M}_{n-1}^{i-1}}{S_n^i - S_n^{i-1}} \tag{1}$$

Slave n then translates to master time the delay measured in local time, by multiplying it by RCF_{M/S_n}, and updates the received estimated master counter value $\hat{M}_{n-1}^i$ according to:

$$\begin{cases} \hat{M}_n^i = \hat{M}_{n-1}^i + \left(\hat{S}_n(LD_n^i) + \hat{S}_n(BD_n^i)\right) \cdot RCF_{M/S_n}^i \\ \hat{M}_0^i = M^i = M(t_i) \end{cases} \tag{2}$$

The last ingredient necessary for being able to execute the update of eq. (2) is the estimation of the line delay to the predecessor, shown on the right in Fig. 2; j indexes the line delay computation. This process uses 4 time-stamps: with periodicity R, node n (the requestor) sends a delay request message to node $n-1$ and records its time of departure, S_{n,req_out}^j (1[st]). Node $n-1$ (the responder) reports in a delay response message the two time-stamps of receiving the request message and transmitting the reply: S_{n-1,req_in}^j and $S_{n-1,resp\_out}^j$ (2[nd] and 3[rd]). The *responder delay* of node $n-1$ is RD_{n-1}^j in absolute time, and is in local time:

$$S_{n-1,respD}^j := S_{n-1,resp\_out}^j - S_{n-1,req\_in}^j . \tag{3}$$

Node n records the time, $S_{n,resp\_in}^j$ (4[th]), when it receives the desired reply, after a *requestor delay* in node n time of:

$$S_{n,reqD}^j := S_{n,resp\_in}^j - S_{n,req\_out}^j . \tag{4}$$

To be able to subtract the time intervals of (3) and (4), each element maintains an "RCF peer" estimate, i.e. frequency ratio estimate to its predecessor, estimated via:

$$RCF_{S_n/S_{n-1}}^j = \frac{S_{n,req\_out}^j - S_{n,req\_out}^{j-1}}{S_{n-1,req\_in}^j - S_{n-1,req\_in}^{j-1}} , \tag{5}$$

Then the line delay can be estimated as:

$$\hat{S}_n(LD_n^j) = \frac{S_{n,reqD}^j - S_{n-1,respD}^j \cdot RCF_{S_n/S_{n-1}}}{2} \tag{6}$$

This concludes the description of the synchronization process. In the following section we will study its accuracy in the face of slave frequency drift, in the absence of other uncertainties. In our analytic work we adopt the usual isolation approach when it is desired to identify the effect due entirely to one specific cause, and therefore neglect jitters (random transmission and reception time noise). They will however be included in the presented simulations. Also, we assume zero delay skew, i.e. direction-independent cable run time. The latter is only a mild idealization, since the IEC61784-5-3 mandates stringent requirements for the

Delay Skew. E.g. for PROFINET it may not exceed 20ns/100m.

III. Analysis of Error Propagation: Effect of Slave Clock Drift in the Absence of Jitter

A. Scenario Description

In this section, we investigate the scenario where the master frequency stays constant, while the slave frequencies drift. The frequency of all elements is constant until t_0, after which some of the slaves' frequencies increase linearly. Short-term linear frequency drifts are typically temperature-induced, as can be verified from the corresponding characteristic curves. For situations with nonlinear frequency change, our analysis is a local first order approximation.

The Master's frequency follows $f_M(t_i) = f_M(t_{i-1}) = f_M$, while the frequency of slave n has slope Δ_n and follows:

$$f_n(t_i) = f_n(t_{i-1}) + \Delta_n \cdot (t_i - t_{i-1}), \quad t_i > t_{i-1} > t_0, \tag{7}$$

$$f_n(t_2)/f_n(t_1) = 1 + \Delta_n \cdot (t_2 - t_1)/f_n(t_1). \tag{8}$$

The counter value increase of each element over the time interval (t_{i-1}, t_i) is the integral over the element's frequency. For the constant-frequency master element this results in:

$$M(t_i) = M(t_{i-1}) + f_M \cdot (t_i - t_{i-1}) \tag{9}$$

For slave n the counter value increase is calculated as:

$$\begin{aligned} S_n(t_i) - S_n(t_{i-1}) &= \int_{t_{i-1}}^{t_i} f_n(t) \cdot dt = \\ &= f_n(t_{i-1}) \cdot (t_i - t_{i-1}) + \frac{\Delta_n}{2} \cdot (t_i - t_{i-1})^2. \end{aligned} \tag{10}$$

Due to the assumed linearity of the frequency change, we can rewrite (10) as the product of the frequency in the middle of the time interval times the length of the interval:

$$S_n(t_i) - S_n(t_{i-1}) = f_n\left(\frac{t_i + t_{i-1}}{2}\right) \cdot (t_i - t_{i-1}). \tag{11}$$

E.g., using the "latency" defined in II, the bridge delay is:

$$\begin{aligned} \hat{S}_n(BD_n^i) &= f_n(mid\ Bridge\ Delay) \cdot BD_n^i = \\ &= f_n(t_i + L_n^i + BD_n^i/2) \cdot BD_n^i \end{aligned} \tag{12}$$

B. Effect of frequency drift on the accuracy of the line delay estimate

In [10] the line delay estimate of slave n is derived to be:

$$\begin{aligned} \hat{S}_n(LD_n^j) &\approx f_n(t_i + L_n^i - A_n^{i,j(i)}) \cdot LD_n^j + \\ &+ RD_{n-1}^j \cdot \frac{R + RD_{n-1}^j}{4} \cdot (\Delta_n - \Delta_{n-1}), \end{aligned} \tag{13}$$

where $A_n^{i,j(i)}$ is the "age" of the line delay computation j valid for Sync Message i, i.e. the middle of the last line delay computation interval until the arrival time of the current Sync message in slave n, upper bounded by the requestor interval R introduced earlier.

C. Derivation of Synchronization Error Expression

In this part, the expression of the master counter estimation error at each element will be derived. The RCF value computed by Slave_1 is, using (9) and (11):

$$\begin{aligned} RCF_{M/S_1}^i &= \frac{M(t_i) - M(t_{i-1})}{S_1(t_i + LD_1{}^i) - S_1(t_{i-1} + LD_1{}^{i-1})} = \\ &= \frac{f_M \cdot T}{f_1\left(t_i + LD_1{}^i - \frac{T + \delta_{LD}^{i,1}}{2}\right) \cdot (T + \delta_{LD}^{i,1})} \end{aligned} \tag{14}$$

In the absence of jitter $\delta_{LD}^{i,n} = 0$, hence:

$$RCF_{M/S_1}^i = \frac{f_M}{f_1(t_i + LD_1{}^i - T/2)} \tag{15}$$

Slave_1 forwards the received Sync message at the time $t_i + LB_1^i$ and from (2) replaces the master estimate by:

$$\begin{aligned} \hat{M}_1^i &= M^i + \left(\hat{S}_1(LD_1^i) + \hat{S}_1(BD_1^i)\right) \cdot RCF_{M/S_1}^i = \\ &= M^i + f_M \cdot \frac{f_1(t_i + LD_1^i - A_1^{i,j(i)})}{f_1(t_i + LD_1{}^i - T/2)} \cdot LD_1^{j(i)} + \\ &+ f_M \cdot \frac{\Delta_1}{f_1(t_i + LD_1{}^i - T/2)} \cdot RD_M^{j(i)} \cdot \frac{R + RD_M^{j(i)}}{4} + \\ &+ f_M \cdot \frac{f_1(t_i + LD_1^i + \frac{BD_1^i}{2})}{f_1(t_i + LD_1{}^i - T/2)} \cdot BD_1^i. \end{aligned} \tag{16}$$

We have used (15), (13) and (12). We can use (8) to simplify the frequency quotients, to obtain:

$$\begin{aligned} \hat{M}_1^i &= M^i + f_M \cdot (LD_1^{j(i)} + BD_1^i) + \\ &+ \frac{f_M \cdot \Delta_1}{f_1(t_i + LD_1^i - \frac{T}{2})} \cdot \left(\begin{aligned} & LD_1^{j(i)} \cdot (T/2 - A_1^{i,j(i)}) + \\ & + RD_M^{j(i)} \cdot \frac{R + RD_M^{j(i)}}{4} + \\ & + BD_1^i \cdot (BD_1^i + T)/2 \end{aligned} \right) \end{aligned} \tag{17}$$

Looking at the denominator, we observe that

$$[f_n(t_i \pm \tau)]^{-1} = [f_n(t_i) \pm \Delta_n \cdot \tau]^{-1} \approx [f_n(t)]^{-1}[1 \mp \Delta_n \cdot \tau / f_n(t_i)]. \tag{18}$$

Practical values for the variables are a nominal frequency of 100 MHz, $75 Hz/s \le \Delta_n \le 300 Hz/s$, $\tau \approx 30 ms$, for which the 2nd term in (18) is $6 \cdot 10^{-8}$, hence negligible, and hence:

$$\begin{aligned} \hat{M}_1^i &= M^i + f_M \cdot (LD_1^{j(i)} + BD_1^i) + \\ &+ \frac{f_M \cdot \Delta_1}{f_1(t_i)} \cdot \left(\begin{aligned} & LD_1^{j(i)} \cdot (T/2 - A_1^{i,j(i)}) + \\ & + RD_M^{j(i)} \cdot \frac{R + RD_M^{j(i)}}{4} + \\ & + BD_1^i \cdot (BD_1^i + T)/2 \end{aligned} \right) \end{aligned} \tag{19}$$

Using (9) the master counter at this time is:

$$M\big|_{t_i + LB_1^i} = M(t_i) + f_M \cdot LB_1^i = M^i + f_M \cdot (LD_1^i + BD_1^i). \tag{20}$$

Therefore the estimation error of slave_1 is given by:

$$\begin{aligned} M - \hat{M}_1\big|_{t_i + LB_1^i} &= f_M \cdot (LD_1^i - LD_1^{j(i)}) - \\ &- \frac{f_M \cdot \Delta_1}{f_1(t_i)} \cdot \left(\begin{aligned} & -LD_1^{j(i)} \cdot (A_1^{i,j(i)} - T/2) + \\ & + RD_M^{j(i)} \cdot \frac{R + RD_M^{j(i)}}{4} + \\ & + BD_1^i \cdot (BD_1^i + T)/2 \end{aligned} \right). \end{aligned} \tag{21}$$

The 1st term comes from the difference of estimated and actual line delay, unless there are no jitters. Since the first component of the 2nd term is much smaller than the other two, $Slave_1$'s estimate of the Master is too high if his own frequency is increasing and too low if his own frequency is decreasing. The RCF value computed by $Slave_2$ is:

$$RCF_{M/S_2}^{i} = \frac{\hat{M}_1(t_i + LB_1^i) - \hat{M}_1(t_{i-1} + LB_1^{i-1})}{S_2(t_i + LB_1^i + LD_2^i) - S_2(t_{i-1} + LB_1^{i-1} + LD_2^{i-1})}. \quad (22)$$

Using (19) and (11), repeating the reasoning of (18), and ignoring the insignificant terms containing $\delta_{LD}^{j(i)}$, for the cases of "new line delay estimate in between: yes (lower in each bracket) and no (upper)" this computes to:

$$RCF_{M/S_2}^{i} = \frac{f_M \cdot T + f_M \cdot \delta_{BD}^{i,1} + \frac{f_M \cdot \Delta_1}{f_1(t_{i-1})} \cdot \left[LD_1^{j(i)} \cdot \begin{cases} -T \\ R-T \end{cases} + \frac{\delta_{BD}^{i,1}}{2} \cdot \left(T + BD_1^i + BD_1^{i-1}\right) + \begin{cases} 0 \\ \frac{\delta_{RD_M}^{j(i)}}{4} \cdot \left(R + RD_M^{j(i)} + RD_M^{j(i)-1}\right) \end{cases} \right]}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2}) \cdot (T + \delta_{BD}^{i,1})}$$
$$= \frac{f_M}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})} \cdot \left(1 + \frac{\Delta_1}{f_1(t_{i-1})} \cdot [...]\right) \quad (23)$$

For parameter values of practical interest the error by ignoring the 2nd term in the parenthesis in (23) is under 10^{-8}. Therefore we can safely approximate:

$$RCF_{M/S_2}^{i} \cong \frac{f_M}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})}. \quad (24)$$

$Slave_2$ forwards the received Sync message at the time $t_i + LB_1^i + LB_2^i$ and from (2) replaces the master estimate by:

$$\hat{M}_2\Big|_{t_i + LB_1^i + LB_2^i} = \hat{M}_1^i + \left(\hat{S}_2(LD_2^i) + \hat{S}_2(BD_2^i)\right) \cdot RCF_{M/S_2}^{i} =$$
$$\approx \hat{M}_1^i + f_M \cdot \frac{f_2(t_i + LB_1^i + LD_2^i - A_2^{i,j(i)})}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})} \cdot LD_2^j +$$
$$+ f_M \cdot \frac{\Delta_2 - \Delta_1}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})} \cdot \frac{RD_1^{j(i)}}{4} \cdot (R + RD_1^{j(i)}) + \quad (25)$$
$$+ f_M \cdot \frac{f_2(t_i + LB_1^i + LD_2^i + \frac{BD_2^i}{2})}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})} \cdot BD_2^i$$

We have used (24), (13) and (12). As we have done for $slave_1$, we can use (8) to simplify the frequency quotients:

$$\hat{M}_2^i = \hat{M}_1^i + f_M \cdot (LD_2^{j(i)} + BD_2^i) +$$
$$+ \frac{f_M \cdot \Delta_2}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})} \cdot \begin{pmatrix} LD_2^{j(i)} \cdot (\frac{T + \delta_{BD}^{i,1}}{2} - A_2^{i,j(i)}) + \\ + RD_1^{j(i)} \cdot \frac{R + RD_1^{j(i)}}{4} + \\ + BD_2^i \cdot (BD_2^i + T + \delta_{BD}^{i,1})/2 \end{pmatrix} + \quad (26)$$
$$- \frac{f_M \cdot \Delta_1}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})} \cdot \frac{RD_1^{j(i)}}{4} \cdot (R + RD_1^{j(i)}).$$

Using (9) the master counter at this time is:

$$M\Big|_{t_i + LB_1^i + LB_2^i} = M\Big|_{t_i + LB_1^i} + f_M \cdot (LD_2^i + BD_2^i). \quad (27)$$

Therefore the estimation error of $slave_2$ is given by:

$$M - \hat{M}_2\Big|_{t_i + LB_1^i + LB_2^i} = M - \hat{M}_1\Big|_{t_i + LB_1^i} + f_M \cdot (LD_2^i - LD_2^{j(i)}) -$$
$$- \frac{f_M \cdot \Delta_2}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})} \cdot \begin{pmatrix} LD_2^{j(i)} \cdot (\frac{T + \delta_{BD}^{i,1}}{2} - A_2^{i,j(i)}) + \\ + RD_1^{j(i)} \cdot \frac{R + RD_1^{j(i)}}{4} + \\ + BD_2^i \cdot (BD_2^i + T + \delta_{BD}^{i,1})/2 \end{pmatrix} + \quad (28)$$
$$+ \frac{f_M \cdot \Delta_1}{f_2(t_i + LB_1^i + LD_2^i - \frac{T + \delta_{BD}^{i,1}}{2})} \cdot \frac{RD_1^{j(i)}}{4} \cdot (R + RD_1^{j(i)})$$

We use the reasoning of (18) to simplify the denominators, and to extrapolate to slave N (inductive proof in appendix):

$$M - \hat{M}_N\Big|_{t_i + L_N^i + BD_N^i} = M - \hat{M}_{N-1}\Big|_{t_i + L_{N-1}^i + BD_{N-1}^i} + f_M \cdot (LD_N^i - LD_N^{j(i)})$$
$$+ \frac{f_M \cdot \Delta_{N-1}}{f_N(t_i)} \cdot \frac{RD_{N-1}^{j(i)}}{4} \cdot (R + RD_{N-1}^{j(i)}) -$$
$$- \frac{f_M \cdot \Delta_N}{f_N(t_i)} \cdot \begin{pmatrix} LD_N^{j(i)} \cdot (\frac{T + \delta_L^{i,N}}{2} - A_N^{i,j(i)}) + \\ + RD_{N-1}^{j(i)} \cdot \frac{R + RD_{N-1}^{j(i)}}{4} + \\ + BD_N^i \cdot (BD_N^i + T + \delta_L^{i,N})/2 \end{pmatrix} \quad (29)$$

This error has four components: the 1st term is the error handed down by the predecessor; the 2nd term is due to the estimated line delay being different from the actual incurred line delay, unless there are no jitters: the 3rd term is the own error even if the own frequency is constant, due to the error in line delay computation if the predecessor is drifting; and finally the 4th is the own error due to the own frequency drift, which also includes a clear line delay computation error. We see that each drifting slave leads to an additive error of identical structure, which will be passed on unchanged down the line. An additional smaller error, due to the error in the line delay estimation, is contributed to his successor. For every drifting slave these error terms get added both in the drifting slave and its successor, and are then percolated down the line together with the previous accumulated error.

IV. SIMULATION RESULTS

We have developed a MATLAB simulation tool to test and analyze the synchronization performance of IEEE 1588 in a line with cascaded bridges. We have used this tool to simulate PTP in PROFINET [9]. The model parameters, summarized in Table 1, are given by the Siemens Automation & Drive department. In the simulation performed for this paper, we define "heating" as temperature increase with a speed of 3K/s, resulting in a frequency drift of 3ppm/s. The temperature change starts at 20s, increases from 25°C to 85°C in the next 20s, and then stays constant again. We test PTP synchronization under different scenarios, i.e. “heating” at master or different slave elements, and compare the results.

All figures show the synchronization errors at slaves 1, 2, 32 and 49: Fig. 3 for the case that only the first slave is heating; Fig. 4 if exactly the first two slaves are heating identically; and Fig. 5 if all slaves are heating identically. Then, for comparison, we simulate the case where the master is heating while the slaves’ frequencies are constant. The result is shown in Fig. 6. Finally, it is also realistic that all slaves heat or cool with different speeds: the start time is still 20s, but then the gradient is a random number between -3 and 3 K/s, which is kept for the next 20s, after which it returns to zero. The simulation result of this scenario is shown in Fig. 7.

TABLE I
SIMULATION SETTINGS

Parameter	Value
Number of elements	50
Nominal Frequency	100MHz
Cable delay	100ns
Bridge delay	Uniform [5 15]ms
Temperature change	3K/s
Frequency Change	1ppm/K
Interval of Sync Message	32ms
Interval of Pdelay_request	8s
Interval of RCF calculation	200ms
Number of RCF averaging	7
Number of line delay averaging	8

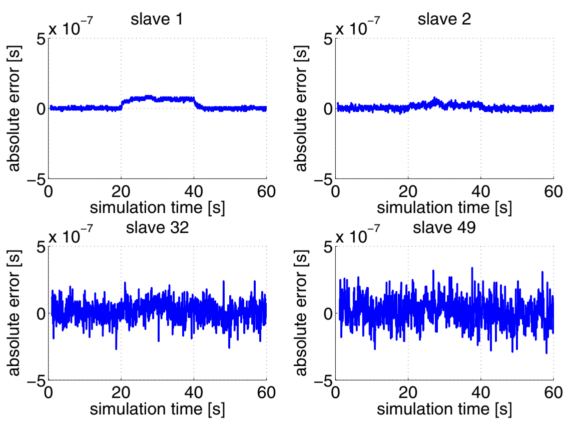


Fig. 3: Sync error at slave 1, 2, 32 and 49, if the 1st slave is “heating”

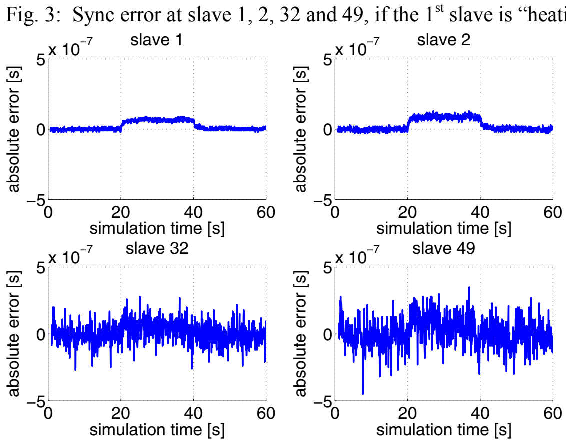


Fig. 4: Sync error at slave 1, 2, 32 and 49, if slaves 1 and 2 are “heating”

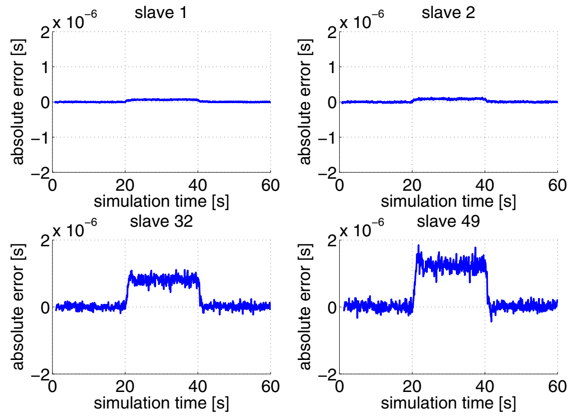


Fig. 5: Sync error at slave 1, 2, 32 and 49, if all slaves are “heating”

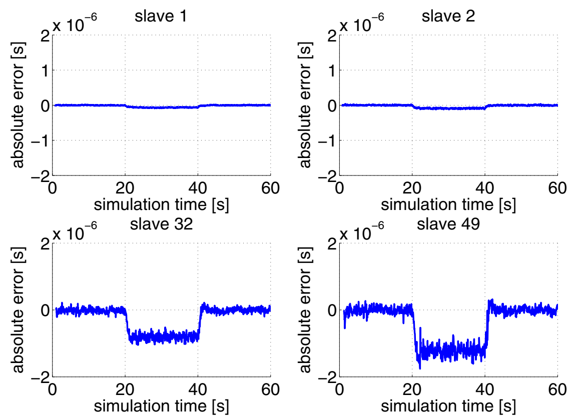


Fig. 6: Sync error at slave 1, 2, 32 and 49, if only the master is “heating”

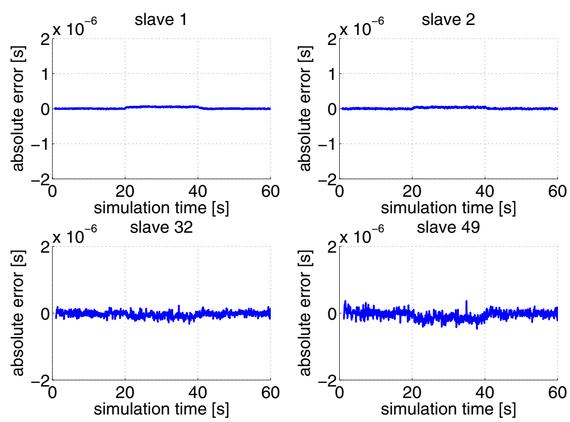


Fig. 7: Sync error at slave 1, 2, 32 and 49, if all slaves are heating or cooling with different speeds

These figures show the following: if only one element is heating, the error at the end of the line is by a factor of 10 larger if this element is the master and not a slave. The same relationship holds for the master versus two slaves heating. Only if all slaves exhibit identical non-zero frequency drift do they match the effect of “master only eating”. However, if the slaves have random gradients, the situation is benign again.

V. Conclusion and Outlook on Future Work

We can conclude that a costly master brings a disproportionately large synchronization benefit, as compared to allocating the corresponding cost fraction to improving each slave. Our next steps will be to expand the analytic synchronization error formula for the "master heating" scenario derived in [6] to include the line delay computation error found in [10], for analytic comparison. Then we intend to investigate further short-term frequency deviations, due to vibration and shocks, as well as the influence of quantization errors on synchronization accuracy.

Appendix

Induction hypothesis $H(N)$:

$$\hat{M}_N\Big|_{t_i+L_N^i+BD_N^i} = M^i + f_M \cdot (L_N^i + BD_N^i) - \sum_{k=1}^{N-1} \frac{f_M \cdot \Delta_k}{f_k(t_i)} \cdot \frac{RD_k^{j(i)}}{4} \cdot (R + RD_k^{j(i)}) + \sum_{k=1}^{N} \frac{f_M \cdot \Delta_k}{f_k(t_i)} \cdot \begin{pmatrix} LD_k^{j(i)} \cdot (\frac{T+\delta_L^{i,k}}{2} - A_k^{i,j(i)}) + \\ + RD_{k-1}^{j(i)} \cdot \frac{R + RD_{k-1}^{j(i)}}{4} + \\ + BD_k^i \cdot (BD_k^i + T + \delta_L^{i,k})/2 \end{pmatrix} \quad (30)$$

The RCF value computed by Slave N+1 is:

$$RCF_{M/S_{N+1}}^i = \frac{\hat{M}_N(t_i + L_N^i + BD_N^i) - \hat{M}_N(t_{i-1} + L_{N+1}^{i-1} + BD_N^{i-1})}{S_{N+1}(t_i + L_{N+1}^i) - S_{N+1}(t_{i-1} + L_{N+1}^{i-1})} \quad (31)$$

Using (30) and (11), in the absence of jitters dropping superscripts on the line delay and repeating the reasoning of (18), and assuming the same line delay estimation process for both consecutive Sync messages, this computes to:

$$RCF_{M/S_{N+1}}^i = \frac{f_M \cdot T + f_M \cdot (\delta_L^{i,N} + \delta_{BD}^{i,N}) + \sum_{k=1}^{N} \frac{f_M \cdot \Delta_k}{f_k(t_{i-1})} \cdot \begin{bmatrix} LD_k^{j(i)} \cdot \left(\delta_L^{i,k} - \delta_L^{i-1,k} - T\right) \\ + \frac{\delta_{BD}^{i,k}}{2} \cdot \begin{pmatrix} T + \delta_L^{i,k} + \\ BD_k^i + BD_k^{i-1} \end{pmatrix} \end{bmatrix}}{f_{S_2}(t_i + L_{N+1}^i - \frac{T + \delta_L^{i,N}}{2}) \cdot (T + \delta_L^{i,N+1})}$$

$$= \frac{f_M}{f_{S_2}(t_i + L_{N+1}^i - \frac{T + \delta_L^{i,N+1}}{2})} \cdot \left(1 + \sum_{k=1}^{N} \frac{\Delta_k}{f_k(t_{i-1})} \cdot \frac{\delta_{BD}^{i,k}}{2} \cdot \left(1 + \frac{BD_k^i + BD_k^{i-1}}{T + \delta_L^{i,N+1}}\right)\right) \quad (32)$$

For realistic parameter values the error by ignoring the summand in the parenthesis is, even for N=100, $O(10^{-6})$. Similar orders of magnitude arise in the case where a new line delay estimate is computed between the two consecutive Sync messages. Hence we can safely approximate:

$$RCF_{M/S_{N+1}}^i \cong \frac{f_M}{f_{N+1}(t_i + L_{N+1}^i - \frac{T + \delta_L^{i,N+1}}{2})}. \quad (33)$$

Slave N+1 forwards the received Sync message at the time $t_i + L_{N+1}^i$ and from (2) replaces the master estimate by:

$$\hat{M}_{N+1}\Big|_{t_i+L_{N+1}^i+BD_{N+1}^i} = \hat{M}_N^i + \left(\hat{S}_{N+1}(LD_{N+1}^i) + \hat{S}_{N+1}(BD_{N+1}^i)\right) \cdot RCF_{M/S_{N+1}}^i =$$

$$= \hat{M}_N^i + f_M \cdot \frac{f_{N+1}(t_i + L_{N+1}^i - A_{N+1}^{i,j(i)})}{f_{N+1}(t_i + L_{N+1}^i - \frac{T + \delta_L^{i,N+1}}{2})} \cdot LD_{N+1}^j + f_M \cdot \frac{\Delta_{N+1} - \Delta_N}{f_{N+1}(t_i + L_{N+1}^i - \frac{T + \delta_L^{i,N+1}}{2})} \cdot \frac{RD_N^{j(i)}}{4} \cdot (R + RD_N^{j(i)}) + f_M \cdot \frac{f_{N+1}(t_i + L_{N+1}^i + \frac{BD_{N+1}^i}{2})}{f_{N+1}(t_i + L_{N+1}^i - \frac{T + \delta_L^{i,N+1}}{2})} \cdot BD_{N+1}^i \quad (34)$$

We have used (30), (13) and (12). Using (8) to simplify the frequency quotients:

$$\hat{M}_{N+1}\Big|_{t_i+L_{N+1}^i+BD_{N+1}^i} = \hat{M}_N^i + f_M \cdot (LD_{N+1}^j + BD_{N+1}^i) - \frac{f_M \cdot \Delta_N}{f_{N+1}(t_i + L_{N+1}^i - \frac{T + \delta_L^{i,N+1}}{2})} \cdot \frac{RD_N^{j(i)}}{4} \cdot (R + RD_N^{j(i)}) + \frac{f_M \cdot \Delta_{N+1}}{f_{N+1}(t_i + L_{N+1}^i - \frac{T + \delta_L^{i,N+1}}{2})} \cdot \begin{pmatrix} LD_{N+1}^j \cdot (\frac{T + \delta_L^{i,N+1}}{2} - A_{N+1}^{i,j(i)}) \\ + \frac{RD_N^{j(i)}}{4} \cdot (R + RD_N^{j(i)}) + \\ + BD_{N+1}^i \cdot \frac{BD_{N+1}^i + T + \delta_L^{i,N+1}}{2} \end{pmatrix} \quad (35)$$

This is precisely $H(N+1)$. Q.e.d.

References

[1] D. L. Mills, "Internet time synchronization: The network time protocol", Network Working Group Request for Comments, 1989.

[2] D. L. Mills, "Precision synchronization of computer network clocks", ACM SIGCOMM Computer Communication Review, Vol. 24, pp. 28-43, 1994.

[3] IEEE, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE, New York. ANSI/IEEE Std 1588-2002, 2002.

[4] J. Jasperneite, K. Shehab, K. Weber, "Enhancements to the time synchronization standard IEEE-1588 for a system of cascaded bridges", in: Proc. of 2004 IEEE International Workshop on Factory Communication Systems, Vienna, 2004.

[5] J. Jasperneite, P. Neumann, "How to guarantee real-time behaviour using Ethernet", in: Proc. of 11th IFAC Symposium on Information Control Problems in Manufacturing (IN-COM2004), Salvador-Bahia, 2004.

[6] C. Na, D. Obradovic, R. L. Scheiterer, G. Steindl and F. J. Goetz, "Synchronization Performance of the Precision Time Protocol", in: 2007 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, Vienna, 2007.

[7] IEEE, IEEE P1588TM D2.2 Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE, New York, 2007.

[8] D. Obradovic, R.L. Scheiterer, C. Na, G. Steindl and F. J. Goetz, "Clock Synchronization in Industrial Automation Networks: Comparison of different Syntonization Methods", in 5th International Conference on Informatics in Control, Automation & Robotics, ICINCO 2008, Portugal.

[9] Jasperneite J., J. Feld (2005). PROFINET: an integration platform for heterogeneous industrial communication systems. In: *Proc. of ETFA 2005, 10th IEEE International Conference on*, pp. 9-12.

[10] R.L. Scheiterer, D. Obradovic, C. Na, G. Steindl and F. J. Goetz, "Synchronization Performance of the Precision Time Protocol: Effect of Clock Frequency Drift on the Line Delay Computation", in 7th IEEE Int. Wkshop on Factory Comm. Systems, WFCS 2008, Dresden.