Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Enhancement of the Precision Time Protocol in Automation Networks with a Line Topology

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

* Siemens AG, Corporate Technology, Information and Communications, Munich, Germany
** Siemens AG, Automation and Drives, Industrial Automation Systems, Amberg, Germany
*** Siemens AG, Automation and Drives, Advanced Technologies and Standards, Nürnberg, Germany
(e-mails: {na.chongning.ext, dragan.obradovic, ruxandra.scheiterer, guenter.steindl, franz-josef.goetz }@siemens.com)

Abstract: The precision time protocol (PTP) delivered by the IEEE 1588 standard has been proven to be an appropriate network synchronization protocol, which is widely applied in the areas of industrial automation, measurement and control, telecommunications and more. Many factors restrict the performance of the PTP protocol. In this paper, we highlight the influence of frequency drift on the synchronization performance. Based on analytic study, we develop an algorithm to improve the protocol, which is verified by simulation.

1. INTRODUCTION

Ethernet-based industrial networks are increasingly replacing field bus systems in the automation and production control area due to cheap cabling and infrastructure costs, high bandwidth, efficient switching technology, better interoperability, etc. Automation applications usually require the network to provide an accurate timing, which is not an easy task over Ethernet. Standard Network Time Protocol (NTP) [Mills 1989], [Mills 1994] provides a synchronization accuracy in the millisecond level, which is appropriate for processes that are not time critical. But for many automation applications, for example in motion control where a latency of 1ms and a jitter of 1μs is allowed, a more accurate synchronization solution is needed. Precision Time Protocol (PTP), delivered by the IEEE 1588 standard [IEEE 2002] published in 2002 becomes a promising Ethernet synchronization protocol where messages carrying precise timing information are propagated in the network to synchronize the slave clocks to a master clock.

Factors that might affect the synchronization quality achievable by PTP include the stability of oscillators, the resolution of message time stamping, the frequency of synchronization message transmission, and the propagation delay variation caused by the jitter in the intermediate elements. Some work has been done to enhance the performance of IEEE 1588 taking the mentioned factors into consideration. Authors of [Jasperneite *et al.* 2004a] introduced the transparent clock concept to replace the so-called boundary clock. With boundary clocks, the local clocks adjust their own clock to the master clock and then the corrected clocks serve as the master clocks for the next network segment. Cascaded control loops are generated, which might lead to instabilities and deviation to distributed clocks. Using transparent clocks, 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. Hence the performance is improved. 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).

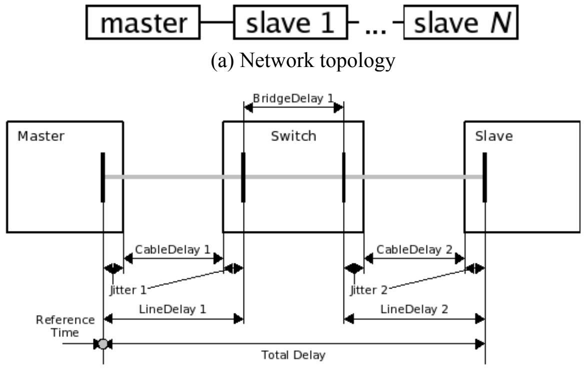
With the peer-to-peer transparent clock implementation of PTP, the current state of the art is to guarantee a synchronization precision of 1μs for topologies with no more than 30 consecutive slaves. To expand this limit it is important to study the factors that influence the quality of the synchronization process and find out methods to minimize effect of these factors. One difficulty arises from the frequency drifts caused by aging and by unpredictable and independent temperature changes at each node. In [Na *et al.* 2007], we analytically derive the expression for the error introduced by the frequency drift. The formulas provide a theoretical ground for the understanding of simulation results as well as guidelines for choosing both system and control parameters when applying PTP. Since master time estimation is necessary in each slave node, elaborate computationally expensive estimators for the master time are out of the question due on the one hand to cost reasons and on the other to the limited information provided by the 1588 messages. In this paper, based on the analytic study, we propose a new algorithm which reduces the effect of frequency drift, thus enhancing the synchronization performance.

The whole paper is organized as follows. Section 2 introduces the system model and briefly describes the PTP protocol. Analysis of the influence of the frequency drifts is presented in Section 3. Section 4 introduces the enhancement of the PTP protocol that reduces the effect of the frequency drift. Simulation results are shown in Section 5.

2. SYSTEM MODEL

2.1 Model of the Network

Figure 1 illustrates the time synchronization in a system with cascaded bridges.



(a) Network topology

(b) System parameters

Fig. 1: System Model

$N+1$ elements are connected in a line topology. The first element is the time server, also called (grand)master, which provides the reference time to the other N elements, called slave elements. The master element periodically sends Sync messages which carry the counter state of the master clock stamped at the time of transmission. The interval between two consecutive Sync messages is T . The i^{th} Sync message, generated by the master element at time t_i , consecutively passes through all slave elements. Quantities, certain or uncertain, linked with the Sync message transmitted by the master at time t_i are labeled by the superscript i . The propagation time between the n^{th} slave and its preceding element is called line delay and denoted by LD_n^i . The message will be forwarded to slave element $n+1$ after the bridge delay BD_n^i . For simplicity, we define LB_n^i to be the sum of line delay plus bridge delay of Sync message i at slave n . As the line delays and bridge delays are not necessarily constant in time, we define $\delta_{LB}^{i,n} = LB_n^i - LB_n^{i-1}$ to be the difference between the true LB value at slave n that affected Sync messages i and $i-1$.

2.2 Brief Description of PTP with transparent clocks

The transparent clock synchronization protocol ([Jasperneite *et al.* 2004a]) is depicted in Figure 2.

The PTP has a master/slave structure. Timing information is packaged in special telegrams and propagated along the network. The synchronization relies on two processes, the delay estimation process and the timing propagation process.

The delay estimation process relies on 4 time-stamps, S_{n,req_out}^j , S_{n-1,req_in}^j , $S_{n-1,resp\_out}^j$ and $S_{n,resp\_in}^j$: slave n sends a delay request message to slave $n-1$ (master for slave 1) and records its time of departure (1st). Slave $n-1$ (or master) replies with a delay response message which reports the time-stamps of receiving the delay request message and sending the delay response message (2nd and 3rd). Slave n records the time it receives the response message (4th). From these, the line delay is computed according to (4), see below.

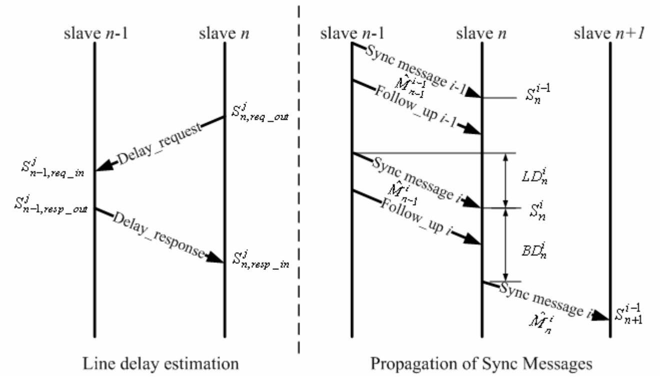


Fig. 2: Illustration of PTP with transparent clocks

In the timing propagation process each slave propagates the timing information of the master and estimates its frequency offset with respect to the master. The master sends out a Sync message which contains the timestamp M when this message was sent. A more precise timestamp of the transmission of the Sync message will be sent by a so-called "follow-up message". Slave 1 forwards the Sync message to Slave 2, augmenting its content by the frequency-offset compensated sum of its line and bridge delays, effectively transmitting its estimate of the master time for the time-instant of forwarding. This process is repeated in each slave until the message reaches the time client.

Slave n updates the estimated master-counter value packaged in the message by augmenting the estimated master-counter state $\hat{M}_{n-1}^i$ received from its uplink element with the time delay incurred since, calculated by translating the sum of the line delay LD_n^i (before slave n) and the bridge delay BD_n^i (within slave n) into master time. To express the line delay and bridge delay in master time, each slave element needs to know its frequency offset to the master element. A rate compensation factor RCF_n has been defined to be the ratio between the frequency of the master and slave n . RCF_n is calculated by using the estimated master-counter values in two Sync messages and the local counter values at the time when the messages arrive at slave n :

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

Slave n then translates the delay measured in local counter value to master-counter value by multiplying it by RCF_n . Hence, denoting by f_{S_n} his frequency, Slave n updates the estimated master-counter value according to:

$$\hat{M}_n^i = \hat{M}_{n-1}^i + f_{S_n} \cdot L\hat{B}_n^i \cdot RCF_n^i = \hat{M}_{n-1}^i + f_{S_n} \cdot \left(BD_n^i + L\hat{D}_n^i\right) \cdot RCF_n^i \tag{2}$$

While f_{S_n} is known only within a given manufacturing precision p around f_{nominal}, it is not needed in (2) since its product with bridge and line delays are obtained directly, as follows. Bridge delays are considered to be precisely determinable, with all uncertainties of time-stamping packed into the line delay, and are recorded at each local element:

$$BD_n^i \cdot f_{s_n} = S_n^{i,in} - S_n^{i,out} \quad (3)$$

while the line delays are estimated by using the timing information obtained in the delay estimation process:

$$L\hat{D} \cdot f_{S_n} = \frac{\left(S^j_{n,resp\_in} - S^j_{n,req\_out}\right) - \left(S^j_{n-1,resp\_out} - S^j_{n-1,req\_in}\right) \cdot RCF_{n-1} / RCF_n}{2} \quad (4)$$

2.3 Scenario Description

In this paper, we investigate the scenario where the master element is uniformly heated and the temperature, and hence the clock frequencies, at the slave elements stay constant. Transmission and reception jitter is neglected, and hence the line delays can be perfectly determined.

We have defined this simple scenario so that the analysis is tractable and on the other hand is representative of many practical real-life situations. The presented analysis can be followed by investigations into more complex scenarios in future work.

Figure 3 plots the frequency of each element as a function of the absolute time.

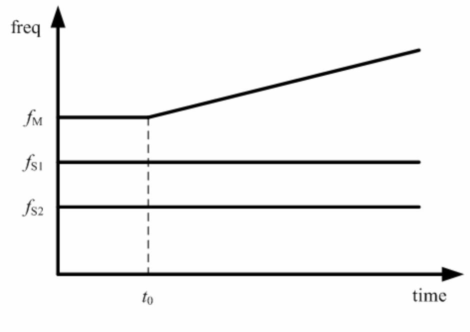


Fig. 3: The frequency change in master heating scenario

The temperature of all elements is constant until t_0, then the temperature of the master element increases linearly afterwards. And as a result, the frequency of master clock increases. We assume that the linear temperature change results in a linear change in clock frequencies, which is a reasonable assumption for a limited range of temperature change. In the case where the frequency change depends nonlinearly of the temperature, our analysis can be seen as a local first order approximation.

Let the slope of the frequency change of the master clock be Δ_M. So the master's frequency follows:

$$f_M(t_i) = f_M(t_{i-1}) + \Delta_M \cdot (t_i - t_{i-1}) \text{ with } t_i > t_{i-1} > t_0 \quad (5)$$

where t_i is the time when the i^{th} Sync message is transmitted by the master. The counter value increase of each element from time t_{i-1} to t_i is obtained by integrating the element's frequency over the interval (t_{i-1}, t_i). For the slave element, whose frequency is constant, the counter value evolves as:

$$S(t_i) = S(t_{i-1}) + f_S \cdot (t_i - t_{i-1}) \quad (6)$$

For the master element, the counter value increase is calculated as:

$$M(t_i) - M(t_{i-1}) = \int_{t_{i-1}}^{t_i} f_M(t) \cdot dt = \int_{t_{i-1}}^{t_i} \left[f_M(t_{i-1}) + \Delta_M \cdot (t - t_{i-1}) \right] dt =$$
$$= f_M(t_{i-1}) \cdot (t_i - t_{i-1}) + \frac{\Delta_M}{2} \cdot (t_i - t_{i-1})^2 \quad (7)$$

Due to the linearity of the frequency change, (7) can be alternatively expressed as the product of the frequency in the middle of the time interval times the interval length, which is sometimes a more useful form:

$$M(t_i) - M(t_{i-1}) = f_M\left(t_i - \frac{t_i - t_{i-1}}{2}\right) \cdot (t_i - t_{i-1}) \quad (8)$$

2.4 Notation

Here we summarize the notation that appears in this paper:

LD_n^i : the line delay before slave n for Sync message i

$L\hat{D}_n^i$: the line delay estimate of slave n for Sync message i, equal to LD_n^i in the absence of jitter

$\delta_{LD}^{i,n} = LD_n^i - LD_n^{i-1}$: the difference between the true line delay at slave n that affected Sync messages i and $i-1$, with $\delta_{LD}^{i,n} = 0$ in the absence of jitter

BD_n^i : the bridge delay of Sync message i at slave n, with $B\hat{D}_n^i = BD_n^i$ (the bridge delay can be determined precisely)

$\delta_{BD}^{i,n} = BD_n^i - BD_n^{i-1}$: the difference between the bridge delay values at slave n that affected Sync messages i and $i-1$ (of the order of magnitude of bridge delay)

$LB_n^i = LD_n^i + BD_n^i$: the sum of line delay (before slave n) plus bridge delay (within slave n) of Sync message i at slave n

$\delta_{LB}^{i,n} = LB_n^i - LB_n^{i-1}$: the difference between the true LB value at slave n that affected Sync messages i and $i-1$, with $\delta_{LB}^{i,n} = \delta_{LD}^{i,n} + \delta_{BD}^{i,n}$

RCF_n : rate compensation factor calculated by slave n, defined in (1).

3. ERROR ANALYSIS IN MASTER HEATING SCENARIO

In our previous work in [Na *et al.* 2007], we have analytically studied the error caused by the frequency drift at the grandmaster. We derived the general expression for the master-counter estimate of slave N at the time $t_i + \sum_{n=1}^{N} LB_n^i$ when slave N forwards the Sync message to slave $N+1$:

$$\hat{M}_{S_N out}\Big|_{t_i+\sum_{n=1}^{N} LB_n^i} \approx M(t_i) + f_M\left(t_i - \frac{T}{2}\right)\cdot \sum_{n=1}^{N} LB_n^i + \frac{\Delta_M}{2}\cdot\left[\left(\sum_{n=1}^{N} LB_n^i\right)^2 - \sum_{n=1}^{N} {LB_n^i}^2\right] - \Delta_M\cdot\sum_{n=1}^{N} LB_{n+1}^i \cdot \left(\sum_{k=1}^{n}\delta_{LB}^{i,k}\cdot\frac{1+\sum_{k=1}^{n} LB_k^i/T}{1+\sum_{k=1}^{n}\delta_{LB}^{i,k}/T}\right) \qquad (9)$$

The error takes the form:

$$M-\hat{M}_{S_N out}\Big|_{t_i+\sum_{n=1}^{N} LB_n^i} = \frac{\Delta_M}{2}\cdot\left[T\cdot\sum_{n=1}^{N+1} LB_n^i + \sum_{n=1}^{N+1} {LB_n^i}^2\right] + \Delta_M\cdot\sum_{n=1}^{N} LB_{n+1}^i \cdot \left(\sum_{k=1}^{n}\delta_{LB}^{i,k}\cdot\frac{1+\sum_{k=1}^{n} LB_k^i/T}{1+\sum_{k=1}^{n}\delta_{LB}^{i,k}/T}\right) \qquad (10)$$

where $M(t)$ is the true counter value at time t, and $\hat{M}$ is the estimated one.

The last term in (10) is a zero mean term which is present due to the changes of the line delays and the bridge delays. Suppose line delays and bridge delays are constant in time, then (10) becomes:

$$M-\hat{M}_{S_N out}\Big|_{t_i+\sum_{n=1}^{N} LB_n^i} \approx \frac{\Delta_M}{2}\cdot\left[T\cdot\sum_{n=1}^{N+1} LB_n^i + \sum_{n=1}^{N+1} (LB_n^i)^2\right] \qquad (11)$$

We use Figure 4 to illustrate the error propagation in (11). The error expression in (11) has two parts. One is proportional to the time elapsed between Sync messages, and to the total delay (rectangles in Figure 4). The other one is the sum of squares of local delays (triangles in Figure 4).

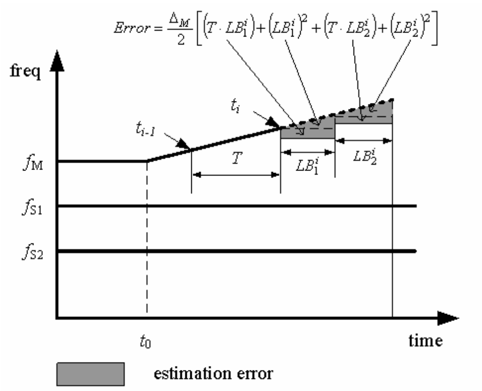


Fig. 4: Sync error at the second slave

The area under the curve of master frequency (f_M) corresponds to the true master counter. The white area is the estimated master-counter value at slave 2, which is the sum of the master-counter value in the original Sync message plus the product of local delay and *RCF* estimate and each slave. It is based only on the master freq. curve between t_{i-1} and t_i, shown as a solid line in the figures, and holds regardless of the further gradient, shown dotted. The gray area is the estimation error. From Figure 4, we can also see that through the propagation of Sync messages, the slave elements can partially follow the recent-past frequency change of the master. As the calculation of RCF uses two consecutive Sync messages, slave element learns the trend of the frequency change of the master from the counters delivered in these two Sync messages. If the frequency drift changes its direction after sending a Sync message, the propagation of this Sync message doesn't contain any information about the new frequency drift and the slaves will still follow the old frequency drift. The worst case scenario is depicted in Figure 5. The master frequency drift changes its direction immediately after the transmission of the i^{th} Sync message. In this case, $slave_N$'s master-counter estimate still takes the form in (9). The true counter value of the master at the corresponding time is:

$$M\Big|_{t_i+\sum_{n=1}^{N} LB_n^i} = M(t_i) + f_M^i\cdot\sum_{n=1}^{N} LB_n^i + \frac{\Delta'_M}{2}\cdot\left(\sum_{n=1}^{N} LB_n^i\right)^2 \qquad (12)$$

where Δ'_M is the slope of the new frequency change. The estimation error in this case will be:

$$M-\hat{M}_{S_N out}\Big|_{t_i+\sum_{n=1}^{N} LB_n^i} \approx \frac{\Delta_M}{2}\cdot T\cdot\sum_{n=1}^{N} LB_n^i + \frac{\Delta'_M}{2}\cdot\left(\sum_{n=1}^{N} LB_n^i\right)^2 + \frac{\Delta_M}{2}\cdot\left(\sum_{n=1}^{N} {LB_n^i}^2\right) - \frac{\Delta_M}{2}\cdot\left(\sum_{n=1}^{N} LB_n^i\right)^2 + \Delta_M\cdot\sum_{n=1}^{N} LB_{n+1}^i \cdot \left(\sum_{k=1}^{n}\delta_{LB}^{i,k}\cdot\frac{1+\sum_{k=1}^{n} LB_k^i/T}{1+\sum_{k=1}^{n}\delta_{LB}^{i,k}/T}\right) \qquad (13)$$

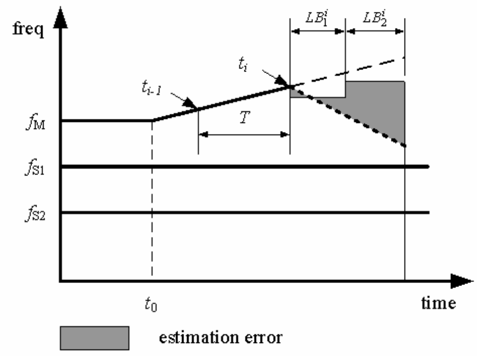


Fig. 5: Sync error at the second slave, propagation of Sync messages does not follow the actual Master frequency drift.

Equations (10) and (11) indicate how to achieve a reduction of the time synchronization error:

by minimizing the influence of the temperature change on the oscillator;

by choosing oscillators that are not sensitive to the temperature change;

by shortening the synchronization interval (however, this contradicts with the choice to minimize the influence of jitter, as shown in [Na *et al.* 2007]);

by regulating the forwarding of Sync messages so that bridge delays are approximately constant over time, this will remove the zero mean term.

Moreover, the error expression inspires a possible modification of the formula used at each slave to compute its estimate of the Master time, so that the effect of temperature change is compensated. This is treated in the next section.

4. ENHANCEMENT OF MASTER TIME ESTIMATION

In order to remove the bias from the estimation error in (10), i.e. the two terms of (11), we need to know the following variables at each slave:

LB , the line delay plus the bridge delay.

T , the interval between two consecutive Sync messages.

Δ_M , the slope of the frequency change at the master.

Equivalently, we need to know, at each slave:

$LB \cdot f_{S_n}$, which can be calculated using (3) and (4).

$T \cdot f_{S_n}$. As each slave stamps the time when Sync messages arrive, this value is the slave counter difference between the arriving time of two Sync messages, i.e.

$$f_{S_n} \cdot T = S_n^{i,in} - S_n^{i-1,in} \quad (14)$$

$\Delta_M / f_{S_n}{}^2$. This value can be estimated as follows:

As RCF is the ratio between the master frequency and the slave frequency, and we assume constant slave frequencies, the change of RCF reflects the frequency drift of the master clock. I.e., we can estimate Δ_M using RCF :

$$\Delta_M = \partial f_M / \partial t = \partial(RCF \cdot f_S)/\partial(S / f_S) = f_S{}^2 \cdot (\partial RCF / \partial S) \quad (15)$$

Due to the linearity property of the master frequency change, the RCF value calculated in (1), RCF_n^i , equals the true RCF value at the time $\left(S_n^i + S_n^{i-1}\right)/2$. Similarly, the previous calculation RCF_n^{i-1} equals the true RCF value at the time $\left(S_n^{i-1} + S_n^{i-2}\right)/2$. Having these correspondences, we can estimate $\Delta_M / f_S{}^2$ by:

$$\frac{\Delta_M}{f_S{}^2} = \partial RCF / \partial S \approx \frac{RCF_n^i - RCF_n^{i-1}}{\left(S_n^i + S_n^{i-1}\right)/2 - \left(S_n^{i-1} + S_n^{i-2}\right)/2} \quad (16)$$

so that the variables on the right hand side of the equation are all known to the element.

Now we can modify the master time estimation equation in (2) by adding an estimate of the bias term of the estimation error:

$$\begin{aligned} \hat{M}_n^i &= \hat{M}_{n-1}^i + f_{S_n} \cdot L\hat{B}_n^i \cdot RCF_n^i + \frac{\Delta_M}{2} \cdot \left(T \cdot L\hat{B}_n^i + L\hat{B}_n^{i\,2} \right) = \\ &= \hat{M}_{n-1}^i + f_{S_n} \cdot L\hat{B}_n^i \cdot RCF_n^i + \frac{\Delta_M}{2 \cdot f_{S_n}{}^2} \cdot \left(\left(T \cdot f_{S_n}\right) \cdot \left(L\hat{B}_n^i \cdot f_{S_n}\right) + \left(L\hat{B}_n^i \cdot f_{S_n}\right)^2 \right) \end{aligned} \quad (17)$$

In (17) all components are calculated as shown above. Using this calculation, the triangle part and the rectangle part of the error in Figure 4 will be removed.

In the case where the frequency drift suddenly changes its direction, the estimation of Δ_M will not be fast enough to immediately follow this change. The error in the worst case is shown in Figure 6.

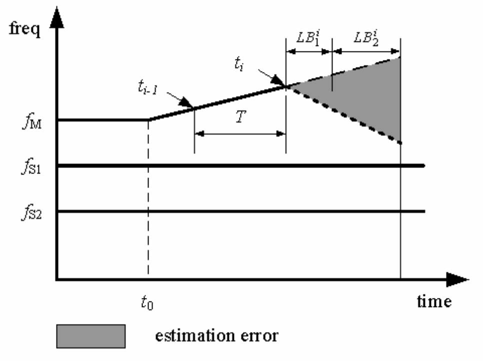


Fig. 6: Sync error at the second slave using new master counter estimation method, propagation of Sync messages does not follow the actual Master frequency drift.

Comparing Figure 6 with Figure 5, the error at the moment of change in frequency gradient is bigger, for the duration of a few sync intervals T , if we use the new master counter estimation method. However, when new Sync messages arrive - if no RCF averaging is done, 2 sync intervals would suffice - the slave can estimate the slope of the new frequency drift and then the error disappears.

5. 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 used this tool to simulate PTP in PROFINET [Jasperneite *et al.* 2005]. The model parameters, summarized in Table 1, are given by the Siemens Automation & Drive department.

Table 1. Simulation Settings

Parameter	Value
Number of elements	80
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 RCF calculation	200ms
Number of RCF averaging	7

In the simulation, the master temperature increases with a speed of 3K/s, resulting in a frequency drift with speed 3ppm/s. The temperature change starts at 20 seconds. It increases from 25°C to 85°C in the next 20 seconds. From 40 seconds on, the temperature will stay constant again. The frequency of slave elements never changes. The results are shown in Figure 7.

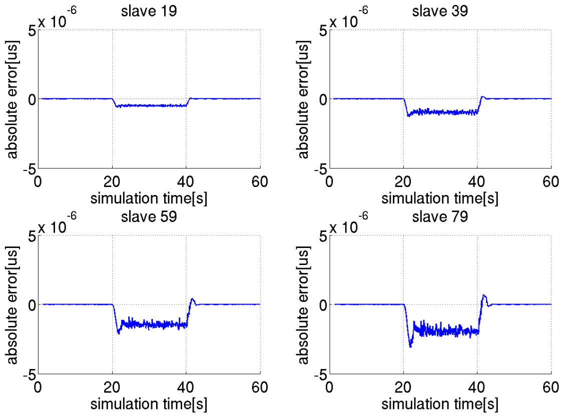


Fig. 7. Sync error at the different slaves when master frequency drifts from 20s to 40s.

From the result we can see, the temperature change introduces a bias to the estimated master time. The magnitude of this bias increases along the line. Our analytic result in (10) ~~also~~ shows that each slave add its own contribution to this bias. When the temperature change stops, this bias disappears. The noise on top of the bias is present due to the variation of the line delays and bridge delays. It corresponds to the last term in (10).

Now we simulate the case where we use the new master counter estimation method introduced in Section 4. The results are displayed in Figure 8.

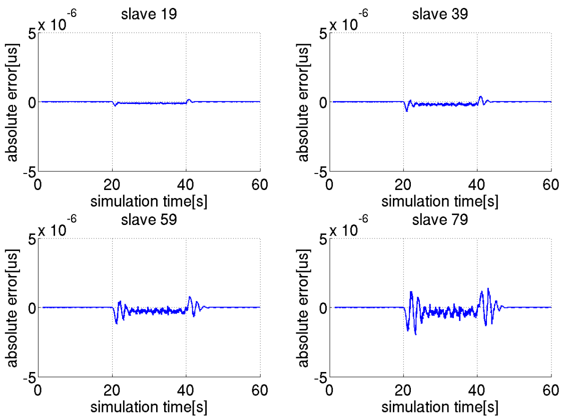


Fig. 8. Sync error at different slaves when the master frequency drifts starting at 20s up until 40s. The new master time estimation method is used.

From the results in Figure 8 we can see that the new method removes the bias of the estimation. A large error can be observed at the beginning and at the end of the frequency change while it takes time for the estimator to follow the actual frequency change, which has been explained in Section 4. In this case the embedded RCF averaging retards the computation of the new gradient.

6. CONCLUSIONS

In this paper~~,~~ we have studied the influence of the master frequency drift on the synchronization performance in the PTP protocol. The error expression has been derived which indicates how the choice of different parameters will affect the synchronization performance. Based on the error analysis, we have developed a practical modification of the master time estimator presented in [Na *et al.* 2007] which enhances the performance of PTP by removing the bias in the synchronization error.

To simplify our model, we have ignored in the analytic treatment other factors that might influence the synchronization, for example the jitter. However, all jitters were included in the simulations, which corroborate our analytic results. Moreover, we have only investigated the frequency drift at the master clock. This is due to our finding that the frequency drift of the master clock is more problematic than the frequency drift of transparent slave clocks because the master clock provides the reference time that every slave should follow, and will be analytically proved in the future work.

REFERENCES

Eidson, J.C. (2006). *Measurement, control and communication using IEEE 1588*, Springer, London.

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

Jasperneite J., K. Shehab, K. Weber (2004). 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*, pp. 239-44.

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

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.

Mills, D.L. (1989). Internet time synchronization: The network time protocol. Network Working Group Request for Comments.

Mills, D.L. (1994). Precision synchronization of computer network clocks. *ACM SIGCOMM Computer Communication Review*, Vol. 24, pp. 28-43.

Na, C., D. Obradovic, R.L. Scheiterer, G. Steindl and F. Goetz (2007). Synchronization Performance of the Precision Time Protocol. In: *2007 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Vienna.