IEEE 1588 implementation with FLL vs. PLL

Zhu Wensi
Department of Electronics and Communications Engineering
Tampere University of Technology
P.O. Box 553, FIN-33101, Finland,
wensi.zhu@tut.fi

***Abstract*—This paper discusses the IEEE 1588 implementation with Frequency Locked Loop (FLL) and Phase Locked Loop (PLL). The work was supported by a Telecom technology provider; the objective is to study the feasibility of IEEE 1588 in the Base Stations (BSs) synchronization, on the basis of minimum structure changes. In order to meet the requirements of the mobile communication systems, IEEE 1588 was implemented using two different methods -- the FLL and PLL. Based on the synchronization result, it is observed that the FLL method shows over adjustments when the network Packet Delay Variation (PDV) is large while the PLL method is likely to be affected by the amount of traffic from the master to the slave clock. Additionally, it is also observed that the overdamped PLL gives high frequency accuracy, but also amplifies the effect of PDV, resulting in a larger frequency instability.**

Keywords—IEEE 1588, FLL, PLL, NTP, Time accuracy and frequency accuracy

I. INTRODUCTION

In the mobile communication systems, Base Stations (BSs) are usually required to be synchronized with each other with a frequency accuracy of 50 ppb and a time accuracy about 3 μs (depending on the employed technology).

The synchronization technique relies on its internal operation and the backhaul network structure. In legacy mobile communication systems, the backhaul network is constructed with techniques like Time Division Multiple (TDM) and Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH). BSs are synchronized to a physical signal, which is used as a carrier signal in the transmission media. Later, with the development of Ethernet, the costly TDM backhaul network is replaced by the Ethernet backhaul network. In another aspect, BSs can also be synchronized to the Global Positioning System (GPS) time services. However, the strict geographical requirements limit the usage of GPS in small scale BSs. One possible solution is to employ the IEEE 1588 precise time protocol in the Ethernet backhaul network. The BS acts as a slave clock without a GPS connection, while the master clocks are synchronized to a traceable time source, such as the GPS time. The backhaul network topology of using IEEE 1588 is illustrated in the Fig. 1 in Ref. [1].

However, additional considerations arise in the deployment. As shown in the Tab. 4.1 in Ref. [2], in the backhaul network, both the traffic and Packet Delay Variation (PDV) from the Base Station Controller (BSC) to the BSs (downlink) is much heavier than that in the opposite direction (uplink). It is claimed that a network with a high delay is usually accompanied with a high delay variation [3]. Moreover, it is observed that the delay and delay variation have different effects on the synchronization. The delay variation is more likely to lead to frequency errors whereas absolute delay affects the time accuracy.

There have been many studies of PDV and the unbalanced traffic effect on the IEEE 1588 performance and possible solutions. Many of these methods require the network devices to be updated. For example, using boundary clocks or transparent clocks needs the IEEE 1588-supported routers and switches. Aside from the arguable performance improvement [4], the downside is that these devices may be costly and not all customers are willing to update devices currently in use.

From the point of the Telecom solution providers, one of the most optimal schemes for employing the IEEE 1588 slave clock in the BS would be in the form of software services. This would require the least system structure modifications if the results meet the specifications.

The goal of this study is to see how accurate the synchronization could be using common network devices. Because of this, the boundary clocks, transparent clocks and even the priority are not used. Meanwhile, in the BS, the existing system hardware structures are intact. Two methods were implemented separately, i.e., FLL and PLL in the slave servo.

The paper is organized as follows. Section II, introduces the IEEE 1588 synchronization mechanisms and the concepts that are used in the implementation. In Section III, the slave clock using the FLL method is examined. It begins with the introduction of the schematics, followed by results of tests performed in a lab network setup and an analysis of the results. In Section IV, the PLL method is examined in the same way as for the FLL method, except that the same schematics are also investigated using Matlab simulations. Finally, conclusions are drawn in Section V.

II. IEEE 1588 SYNCHRONIZATION MECHANISMS

A. Schematics

Corresponding to Section 11 in the IEEE 1588-2008 standard [5], the timestamp exchange operation for delay request-response mechanisms and the packet types are shown in Figure 1. $t_1, \dots, t_4$ are the timestamps that mark the packet ingress and egress time.

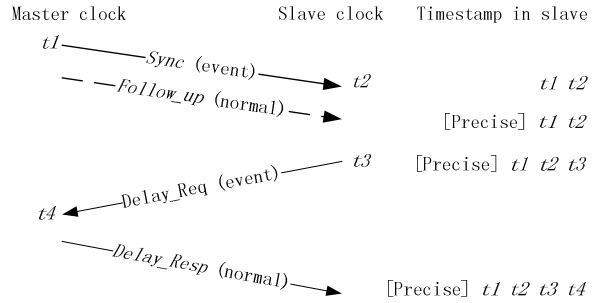

Fig. 1. Timestamp exchange operation.

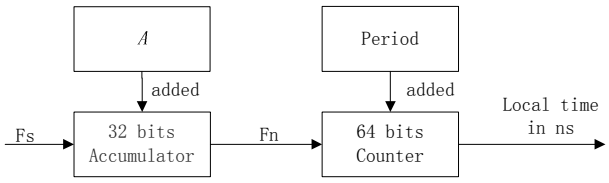

Fig. 2. Slave clock local time derivation.

When the transmission mode is unicast UDP over IPv4, the slave clock time differs from that in the master by

$$\delta = \mathrm{t}_2 - \mathrm{t}_1 - d \tag{1}$$

where δ is the time offset, d is the mean path delay, it satisfies

$$d = \frac{\mathrm{t}_2 - \mathrm{t}_1 + \mathrm{t}_4 - \mathrm{t}_3}{2} \tag{2}$$

B. The processor of slave clock

The designed slave was built on a processor that has IEEE 1588 supported at the hardware level. Figure 2 shows the main structure of the IEEE 1588 related function. In this figure, the 64 bit counter stores the current local time. The system frequency and period in our implementation are 200 MHz and 10 ns respectively. F_s and F_n are the system frequency and nominal frequency. They are related by

$$F_n = F_s \cdot \frac{A}{2^{32}} \tag{3}$$

The synchronization target is to keep F_n fixed at 100 MHz (when the *period* is 10), by adjusting A.

Additionally, the slave clock has its timestamps tagged at the data link layer (of the Open Systems Interconnection model).

III. SLAVE SERVO WITH FLL

A. Mechanisms

Intuitively, FLL is the most straight-forward method to characterize the clock frequency -- the frequency drift accumulated over a period of time would result in the time offset. The frequency error can be compensated for by adjusting the value of A relative to the frequency drift. The slave servo with FLL is shown in Figure 3. FLL_LPF is a set of filters that limit the FLL bandwidth and drop popcorn

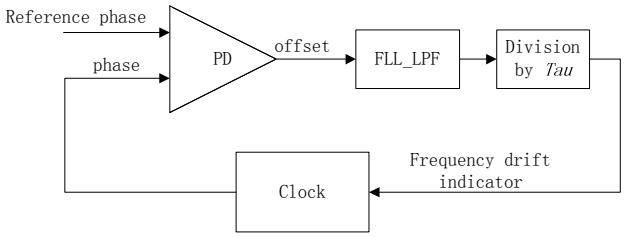

Fig. 3. Slave servo with FLL model.

spikes (occasional large offset spikes) [6]. Every time the slave gains a time offset according to (1), the slave records the moment. Thereafter, the FLL method adjusts the frequency of the clock by counting how much the slave time has wandered during the interval τ (*Tau* in figure).

For example, at the moment of t_A and t_B, the time offset is δ_A and δ_B. The frequency offset f_w can be calculated from the deviation of the time offset over the duration τ as

$$\begin{cases} f_w = \dfrac{\delta_B - \delta_A}{\tau} \\ \tau = t_B - t_A \end{cases} \tag{4}$$

Because the FLL method tunes the frequency drift, it indirectly adjusts the time offset. However, the time offset is an accumulation of the frequency offset, which might result in time drifting in one direction. Therefore, a threshold on the time offset is used to step the clock time directly.

In order to achieve the frequency stability, the following methods have been tried to gain a more stable frequency drift indicator:

- The exponential smoothing of the derived frequency offsets. The FLL bandwidth is 10000.
- The average of 10 successively collected frequency offsets. Namely, the clock is adjusted every 10 samples. The FLL bandwidth is 10000.
- Select the frequency offset that is accompanied with the smallest delay for the last 10 successively collected frequency offsets.

The last method is also the FLL schematics employed in the Network Time Protocol (NTP).

B. Experimental evaluation

The test environment was set up as follows. The grandmaster is a commercial product whose time source is traceable to the GPS time. It is connected with the slave clock through an enterprise Ethernet switch. The network is part of a larger network which contains around 100 hosts. The system frequency modification is reflected by examining the value A, according to (3).

The above three methods are listed in a sequence of the frequency stability -- the smallest delay selection method is more stable than the other two. Even so, the frequencies show some instability.

Based on the 3rd FLL solution, Figure 4 shows the observed frequency indicator, the adjusted A, the offset before the

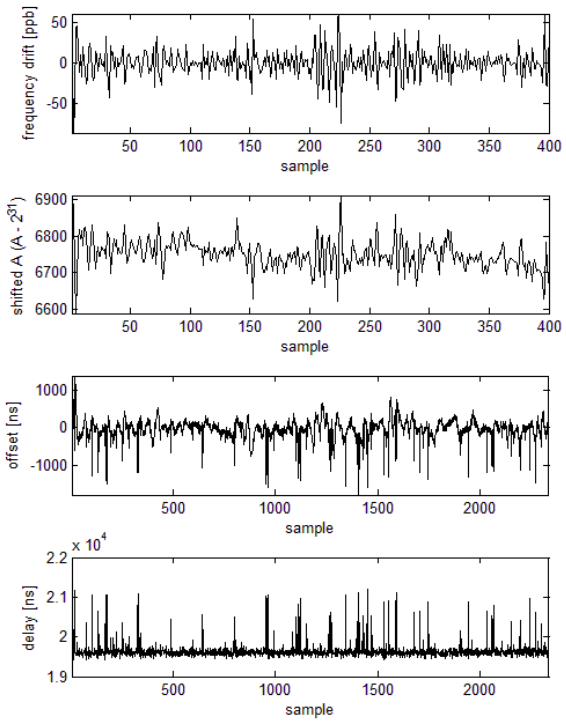

Fig. 4. The observed frequency indicator (top), the adjusted A (top 2nd) and the offset and delay before selection takes place (the last two).

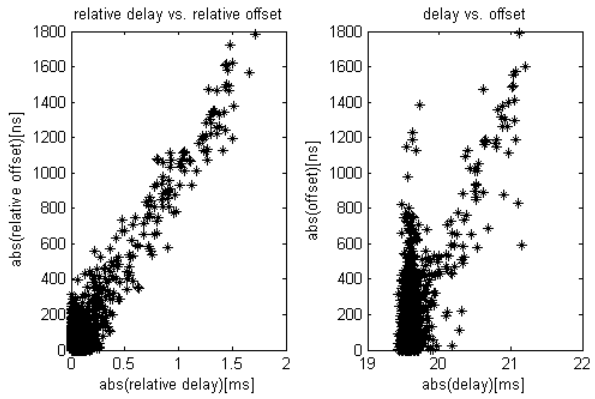

Fig. 5. Comparison of the relations between relative/absolute delay and offset.

selection takes place and the delay, which are collected from the slave clock. On the one hand, the obvious symmetry indicates that adjusting the oscillator frequency would soon cause a predictable adjustment in the opposite direction. On the other hand, the down slope indicates that the system frequency is increasing during the observing period. This is probably caused by environmental temperature changes. The valid sample space of frequency drift is 409, and the space of the offsets (collected before the selection phase) is 2340, so the selection ratio is 17.48%. At last, according to (3), because the average A is equal to $2^{31} + 6800$, the frequency error of the system frequency is about 3.166 ppm.

Another interesting phenomenon is the relation between the delay and the offset. In NTP, the Huff-n'-puff filter explained in [6] is claimed to be useful or even the only useful method to suppress the asymmetric path delay. It is based on an observation that the delay and offset are coherent. But in this implementation, the Huff-n'-puff filter was not observed to improve the performance. The correlation of the relative delay and offset is -0.867 while that of the absolute delay and offset is -0.596. The comparison is shown in Figure 5. The reason for the failure of the filter could be that the experimental network in this paper does not meet all the prerequisites claimed in [6]:

- Severe asymmetry,
- Multiple links available but only one of them is relatively slow,
- The slave clock offset is very small.

IV. SLAVE SERVO WITH PLL

A. Mechanisms

Compared with FLL, the PLL method does not take the duration between two time offset samples into consideration. As one of the most popular implementations, the PLL together with a PI controller method is commonly adopted in both NTP and IEEE 1588 [7]. The PLL implementation in this paper uses the open source project PTPd as a reference [8]. In PTPd, the mean path delay calculated by (2) is fed to an Infinite Impulse Response (IIR) filter before it is subtracted from the slave-to-master offset. The derived time offset is passed through a Finite Impulse Response (FIR) filter. The filtered offset is used as the input of a PI controller before the clock is finally adjusted. The clock is adjusted every time a valid offset is derived. This procedure is illustrated in Figure 6.

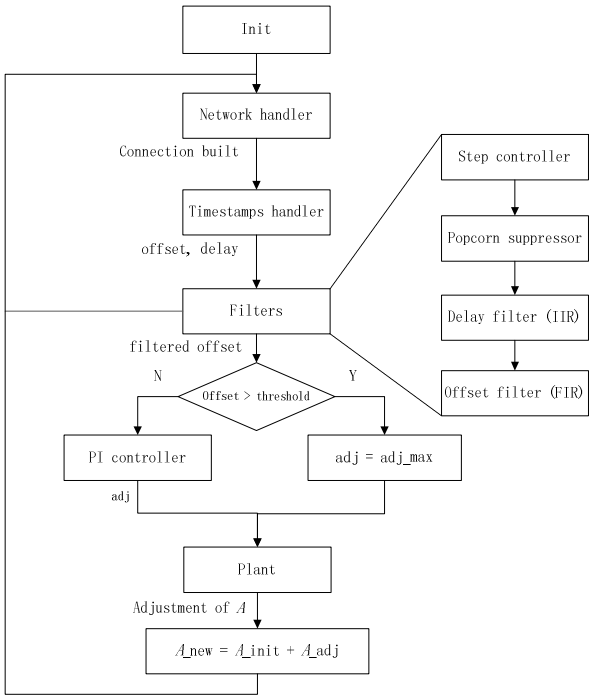

Fig. 6. Flow chart of the slave servo with the PLL method.

B. *Experimental evaluation*

The PLL method was tested with the same test environment as for the FLL method. The performance is evaluated by investigating the calculated time offset from the point of the slave clock. Figure 7 shows the observed offset. The samples exhibit an oscillation that resembles a sinusoidal waveform. Because of the usage of IIR filters, the offset and delay correlation degrades to 0.025 while the relative offset and delay correlation is 0.404. It is obvious that when compared with the FLL method, the PLL is less affected by both the delay and delay variation.

It is mentioned in many materials that the *Sync* packet rate affects the synchronization performance in a way that better characterizes the network path condition better (e.g., the congestion and delay). However, during the experimental test, the higher *Sync* packet rate did not help to improve the time accuracy or the frequency accuracy. With the help of the statistical function of an oscilloscope, the Pulse per Second (PPS) signals from the grandmaster and the slave are compared. Collecting the data from the oscilloscope, Figure 8 shows the time difference and the standard deviations of two PPS signals for different *Sync* rates. The tested *Sync* rates are {0.5 1 2 4 16 64} pkt/s. Due to the difficulties of connecting the slave PPS signal pin with the probe of the oscilloscope, the test for each *Sync* rate lasted about 5 to 10 min.

One possible reason for the poor effect of the higher *Sync* rate could be that the PLL method adjusts the clock every time it receives a valid offset, meaning that a 64 pkt/s *Sync* rate results in a 64 times per second clock adjustment.

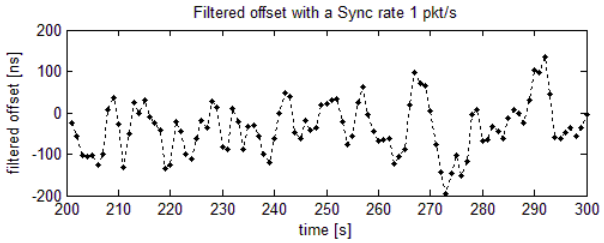

Fig. 7. Observed filtered timing offset.

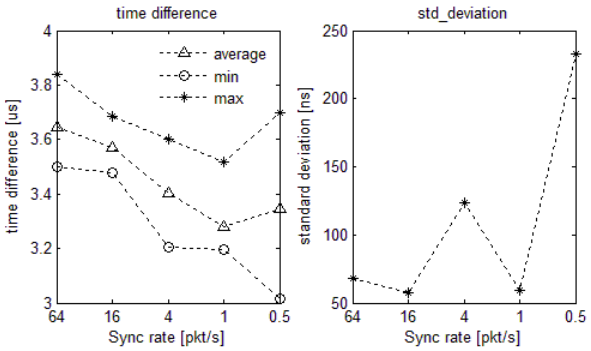

Fig. 8. The time difference (left) and the standard deviation (right) of two PPS signals.

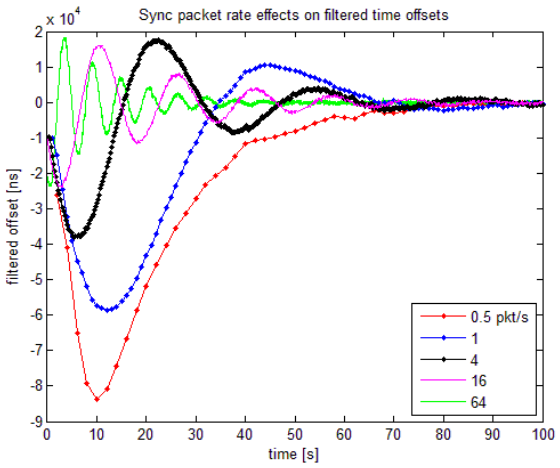

Fig. 9. The *Sync* rate effect on the offset start-up phase.

Figure 9 shows the *Sync* rate effect on the offset start-up phase. It is obvious that the higher the *Sync* rate, the larger the fluctuation. Moreover, regardless of the *Sync* rate, the slave clock exhibits underdamped behavior, and the settling time is almost the same.

C. *Numerical simulations*

In order to find the reason for the PLL having regular envelope of the adjusted A, and the under-damping behavior of the clock, the PLL method was examined in Matlab simulations. The simulations were conducted for two cases,

- The polluted case where the delay effect on time offsets is simulated,
- The ideal case in which the delay does not exist and the system frequency does not vary.

The polluted case simulation is to inspect how the PDV affects the synchronization accuracy. The ideal case simulation is to study how the system characteristics affect the results.

1) Polluted case.

The simulation is achieved as follows: first, the delay and the unbiased time and frequency offsets are simulated separately. Secondly, the program passes the delay to the IIR filter. The effects (i.e. changes) of the filtered delay are added to the unbiased offset to form the biased offset at last.

The delays in two directions are generated as independent random variables. The large PDV, which occurs in the experimental tests, is simulated with a ratio of 99%. And the PDV data is generated by amplifying the corresponding noise by 3 times. To test the effects of different PDV positions, the PDV data is added in two directions separately. Moreover, in order to make the simulation more persuasive, the input delays are the same except the superimposed large PDVs.

The effect of the PDV on the frequency accuracy is shown in Figure 10, where the absolute value of the frequency offset is drawn for two scenarios separately. In this figure, d_{sm} and d_{ms} correspond to the slave-to-master delay and master-to-slave

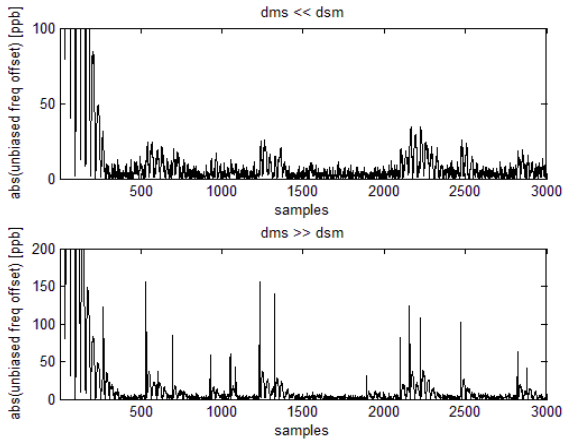

Fig. 10. The absolute value of frequency offset for two scenarios.

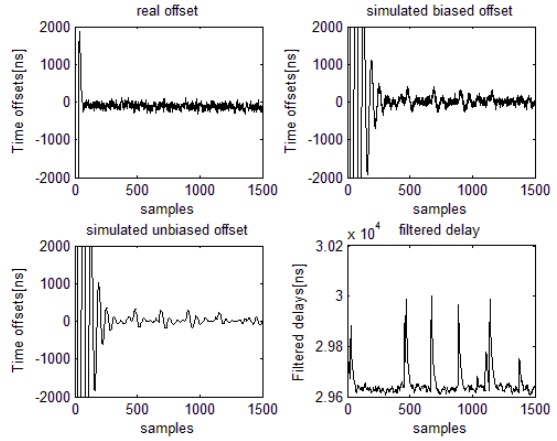

Fig. 11. Time offset result with live test collected offsets (upper left), simulated biased offsets (upper right), unbiased offsets (lower left) and filtered delay (lower right).

delay. It is obvious that a larger PDV in the d_{ms} causes more error in the frequency (bottom in Figure 9). The reason for this could be because of the IEEE 1588 mechanisms. The time offset calculated by (1) is more likely to be affected by the PDV in the master to slave direction. Even so, compared with the FLL method, PLL is more robust toward a large PDV.

On the other side, Figure 10 also shows that the sinusoidal amplitude of A in Figure 7 is actually the frequency fluctuation. The frequency fluctuation exhibits a certain periodic feature. This behavior is in accordance with the live test observation: the PPS signal of the slave clock wanders around a center position with a harmonic motion. This could be caused by the chosen parameters in the PI controller. Figure 11 shows the simulated biased offset and unbiased offset for the case that the large PDV resides in the slave to master direction. The proportional and integral parameters are 0.04 and 0.01.

2) Ideal case

To test how the PI controller parameters affect the system, the system is simulated without any delays. The system frequency is exactly 200 MHz. Given these conditions, from the point of the time offset, the mathematical expression of the system is

$$\begin{cases} d[n+1] = d[n] + \dfrac{A}{p} - \dfrac{1}{p} \cdot a[n] \\ a[n] = k_p \cdot d[n] + k_i \sum_{i=0}^{n} d[i] \end{cases} \tag{5}$$

where $d[n]$ is the current time offset in ns, p is the *Sync* rate in pkt/s, $a[n]$ is the output of the PI controller, and k_p, k_i are the proportional and integral parameters. Rewrite (5) into one equation, it gives

$$d[n+1] = d[n] + \frac{A}{p} - \frac{k_p}{p} \cdot d[n] - \frac{k_i}{p} \sum_{i=0}^{n} d[i] \tag{6}$$

The constant in (6) determines the oscillation phase and amplitude. It also causes the system oscillate when the initial condition $d[0]$ is 0. The coefficients of $d[n]$ affect both the envelope and the period. The initial condition $d[0]$ only affects the initial oscillation phase. The observed *Sync* rate effect shown in Figure 9 could also be explained by (6): it affects the oscillation by changing the coefficients of $d[n]$. Based on these, if the constant is not considered, and using

$$a_p = \frac{k_p}{p}, \qquad a_i = \frac{k_i}{p}$$

equation (6) becomes

$$d[n+1] = d[n] - a_p \cdot d[n] - a_i \sum_{i=0}^{n} d[i] \tag{7}$$

As can be seen from (7), from the point of the time offset, the PLL system could be simplified into a system that only contains proportional and integral blocks.

The above equations are drawn in form of 3D images with different a_p and a_i. The parameters are configured as such: $d[0]$ is 0, the constant is 10 and the loop size (n) is 500. The ranges of a_p and a_i are identical, i.e., 0.01 to 1 in steps of 0.01. Figure 12 shows the number of oscillation peaks.

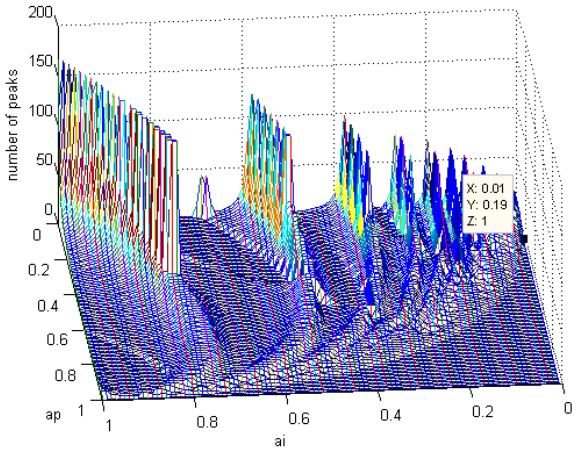

Fig. 12. The number of oscillation peaks.

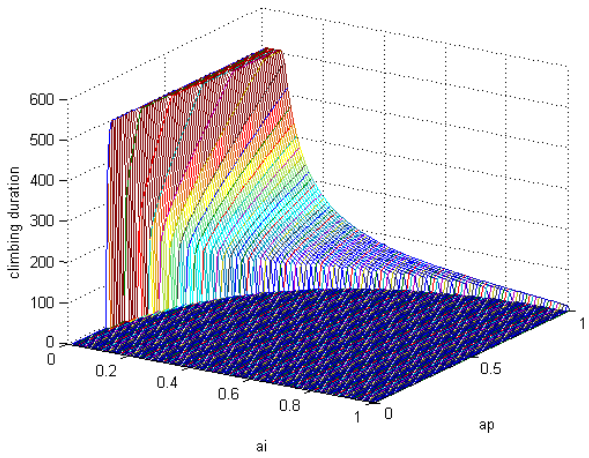

Fig. 13. The number of samples before the system reaches its final value for overdamped and critically damped scenarios.

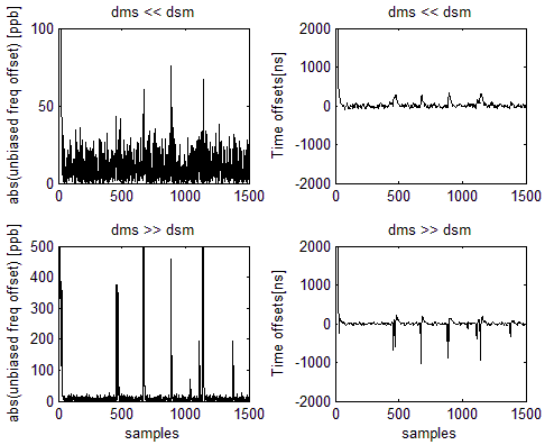

Fig. 14. The absolute value of frequency offset for two scenarios: large PDV resides in the slave to master direction (top) and in the master to slave direction (bottom).

When the number of peaks is 1, it is either overdamped or critically damped. Figure 13 shows the climbing duration with the same simulation parameters. The more samples it takes to reach the stable condition, the larger the damping ratio. In the simulation, the underdamped condition is described as having 0 climbing duration. Comparing Figure 12 and the left graph in Figure 13, the parameters that make system underdamped are identical, i.e., they are located within a quadrant (approximately).

We randomly choose the a_p and a_i pairs that are outside the quadrant (either overdamped or critical damped), and ran the polluted simulation program. The result is that, as expected, the clock does not behave underdamped, and when the slave to master PDV is heavier, the synchronization performance is improved. However, if the large PDV resides in the master to slave direction, the PI controller sharpens the time offset peaks. This is shown in Figure 14 (the parameters a_p and a_i are 0.19 and 0.01). Besides, by examining simulations with a larger climbing period a_p and a_i pairs, we observed that a higher damping ratio is accompanied with a larger frequency variation. The larger *Sync* rate p improves the frequency and time stability, with a prerequisite that the PI controller parameters are fixed. However, this is more likely because the larger *Sync* rate decreases the coefficients of the time offset as in (6) and thereafter suppresses the time offset variations. Moreover, in some cases (depending on the PI controller parameters), a certain *Sync* rate changes the harmonic conditions, e.g., from overdamped to underdamped.

V. CONCLUSIONS

This paper examines the performance of the IEEE 1588 slave clock implemented by FLL and PLL.

The experimental evaluation of the FLL method presented in this paper shows that over adjustments on the clock frequency occur from time to time. Even with the smallest delayed packet selection implemented, the clock frequency is observed to have one adjustment followed by an almost equal amplitude adjustment towards the opposite direction. The calculated frequency drift is sensitive to PDV.

The experiments of the PLL method show that it is more robust to PDV by having fewer over adjustments compared with the FLL method. Even so, the method of the time offset calculation suggests that a heavier PDV in the master to slave link introduces a larger frequency error. By analyzing the PLL simulations, the PI controller parameters and the *Sync* packet rate affect the slave clock by affecting the damping ratio. The parameters that make the slave clock overdamped would amplify the time offset variations (caused by PDV) resulting in a larger frequency instability.

REFERENCES

[1] J. Han and D. K. Jeong. "Practical Considerations in the Design and Implementation of Time Synchronization Systems Using IEEE 1588," Topics in Design and Implementation. IEEE Communications Magazin, Nov. 2009.

[2] A. Kalliola, "Applicability of IEEE 1588 to TETRA Base Station Clock Synchronization," Master's thesis, Aalto University, 2010.

[3] L. Cosart, "IEEE 1588 frequency and time transfer measurements and analysis: Clock PDV, and load," *in Proc. of the 42nd Annual Precise Time and Time Interval (PTTI) Meeting.*

[4] R. Zarick, M. Hagen, and R. Batoš, "Transparent clocks vs. enterprise Ethernet switches," in *Proc. 2011 IEEE Int. Symp. on Precis. Clock Synchr. for Meas., Ctrl and Commun. (ISPCS)*, Munich, Germany, 12-16 Sep. 2011, pp. 62-68.

[5] *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control systems,* IEEE Std. 1588-2008, Rev. IEEE 1588-2002, 2008.

[6] D. L. Mills, *Computer Network Synchronization: The Network Time Protocol*, 1st ed. Boca Raton, FL: CRC Press, 2006.

[7] J. C. Eidson, *Measurement, Control, and Communication Using IEEE 1588*, Springer-Verlag, 2006.

[8] T. Kovacshazy and B. Ferencz, "Performance evaluation of PTPd, a IEEE 1588 Implementation, on the x86 Linux platform for typical application scenarios," in *Instrumentation and Measurement Technology Conference (I2MTC)*, 2012 IEEE International, May 13-16, 2012, pp. 2548-2552.