Clock Skew Estimation Using Kalman Filter and IEEE 1588v2 PTP for Telecom Networks

Zdenek Chaloupka, *Member, IEEE*, Nayef Alsindi, *Senior Member, IEEE*, and James Aweya, *Senior Member, IEEE*

***Abstract*—Accurate and precise frequency synchronization is an essential requirement for different areas in telecommunication industry. Emerging practise for frequency synchronization is to utilize packet networks since it is highly cost effective. One method of distributing frequency over a variety of packet networks is based on well known IEEE 1588v2 PTP standard that uses a master-slave architecture. Accuracy and precision of the frequency synchronization over IEEE 1588v2 PTP is mainly deteriorated by the Packet Delay Variations (PDVs) experienced on the transmission path. This problem can be overcome by deploying so-called timing aware routing elements, however slaves' frequency accuracy and precision still heavily relies on the quality of the slave's clock synchronization algorithm. Hence, this work introduces an improved Kalman filter based algorithm for frequency synchronization over IEEE 1588v2 protocol that outperforms other prior art techniques. The algorithm's performance is evaluated using simulation, yet the network impairments (PDVs) are based on experimentally measured data.**

***Index Terms*—Clock skew, frequency synchronization, IEEE 1588, Kalman filter, linear programming, linear regression.**

I. Introduction

TIMING transfer using combination of the IEEE 1588v2 PTP [1] and a well designed slave clock recovery mechanism can provide time and frequency synchronization that meets ITU-T Primary Reference Clock's precision and accuracy requirements. In reality, however, such requirements are met only when Full On-path Timing Support (FOTS) is provided [2] due to impairments introduced in various Network Elements (NEs), e.g. routers and switches. FOTS uses Boundary (BC) or Transparent Clocks (TC) functionality [1] to mitigate queuing delays, however, its deployment does not guarantee zero PDVs due to finite precision of BC and TC implementations [3]. Thus, a high quality clock recovery mechanism is still required. Since FOTS architecture is costly, network operators use so called Partial On-path Timing Support (POTS) as an alternative [2]. POTS architecture deploys BC or TC enabled devices only for the most utilized NEs to mitigate the PDVs and consequently, a high quality clock recovery mechanism is required in the slave device to suppress the residual PDVs.

Another much demanded application of frequency synchronization is for the attractive small cell base station deployment that uses a Long-Term Evolution Frequency Division Duplex (LTE-FDD) technology. A key enabling factor for a massive deployment of small cells is to find a solution for frequency synchronization that is inexpensive (i.e., packet networks based), yet meets the accuracy/precision LTE-FDD requirements of frequency error less than 16 Parts Per Billion (PPB) at network interface [2].

The main focus of our work is to find a solution to the clock skew estimation (frequency synchronization) problem that is practical, low complexity and meets LTE-FDD requirements. What is regarded as a practical solution in telecom networks is an algorithm with a short convergence time and an effective usage of the transmission bandwidth. To conserve bandwidth algorithms are required to operate solely in One Way Mode (OWM), see Fig. 1. Note that OWM also avoids forward/backward path asymmetry problems that are otherwise quite common because of the telecom network's core design.

Literature review indicates that there has been a number of different clock skew estimators designed over the last decade, yet only the algorithms mentioned here match the OWM requirement. In [4] three approaches with different level of complexity were discussed: Kalman Filter (KF); optimization based Linear Programming (LP); and an averaging method (omitted here due to underperformance). In [5] a Linear Regression (LR) based algorithm was proposed, however its comparison to KF and LP methods is missing.

In this letter we first introduce a modified KF (Section III) that has the capabilities to meet the stringent requirements of LTE-FDD standard (in contrast to the former KF model [4]). Then we compare its performance with prior art techniques using realistic data from standardized ITU-T G.8261 test bed (Section IV). Since all algorithms are, in a way, affected by the finite window length (whose effect on the performance has not yet been shown in literature), we analyse the effect of windowing in Section IV. As we require a low complexity algorithm, we will show that KF's complexity is superior to other prior art techniques (Section IV). Based on the above the main contribution of this letter is summarized as:

1) Propose a modified KF model.
2) Compare the performance and complexity of our updated KF model with prior art techniques using experimental data from standardized ITU-T G.8261 test bed.
3) Analyse the effect of windowing on performance.

II. Background

IEEE 1588v2 PTP distributes time information from a master to its slaves by sending accurately timestamped packets at nanosecond level granularity. The PTP message exchange process (similar to NTP architecture) is described below and illustrated in Fig. 1. Note that symbols and notations used in this letter are explained close to the equation that uses them. We

Manuscript received January 20, 2015; accepted April 23, 2015. Date of publication April 28, 2015; date of current version July 8, 2015. The associate editor coordinating the review of this paper and approving it for publication was H. Mehrpouyan.

The authors are with Etisalat BT Innovation Center (EBTIC), Khalifa University of Science, Technology and Research, Abu Dhabi, UAE (e-mail: zdenek.chaloupka@kustar.ac.ae; nayef.alsindi@kustar.ac.ae; james.aweya@kustar.ac.ae).

Digital Object Identifier 10.1109/LCOMM.2015.2427158

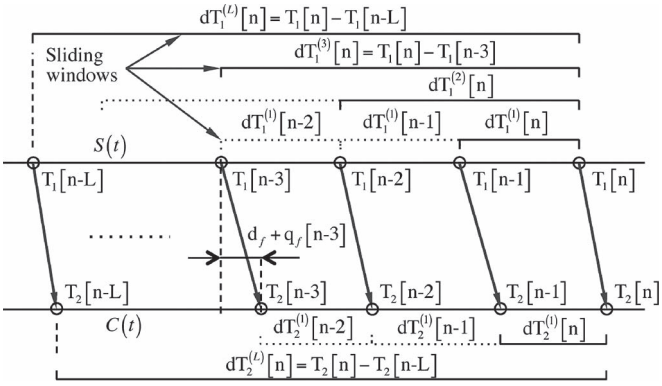

Fig. 1. IEEE 1588v2 PTP master/slave sync message exchange only.

define now a generalized clock offset and skew equation for the synchronization problem. We assume that at any particular time instant, the instantaneous view of the relationship between the master (server) clock with timeline $S(t)$ and the slave (client) clock with timeline $C(t)$, can be described by the well-know simple skewed clock defined as:

$$S(t) = (1+\alpha)C(t) + \theta, \tag{1}$$

where α represents skew and θ is time offset of the slave clock. Clock skew α is sometimes referred to as frequency offset defined as $dC(t)/dt - 1$. (1) can be extended to a discrete case where the master clock and slave clock exchange messages over a communication link with delay.

Let us assume that nth Sync message departs the master with timestamp $T_1[n] \in S(t)$ and arrives at the slave with timestamp $T_2[n] \in C(t)$ experiencing a fixed physical link delay of d_f plus variable queuing delay of q_f (see Fig. 1). With respect to the PTP protocol we can write:

$$T_1[n] + d_f + q_f[n] = (1+\alpha[n])\, T_2[n] + \theta. \tag{2}$$

A key assumption is that the message exchanges occur over a period of time so small that the skew α can be assumed constant over that period. This is a valid assumption since telecom grade oscillators (Stratum 3) change their parameters very slowly (i.e. short term stability less then one PPB). A skew's drift interval is several orders higher (hours) than the time during which PTP message exchanges occur (milliseconds) as PTP is usually set in range of 16$\sim$64 Messages Per Second (MPS). Note that we do not introduce the classical two way message exchange as Sync message's stream is sufficient for any clock skew estimation algorithm working in OWM. It is obvious that to syntonize a slave clock to master's clock one only has to find the slave clock's skew (α).

III. Kalman Filter Design

We update the definition from [4] with respect to IEEE 1588v2 protocol as follows. Let us define the first difference of the timestamp T_i in a sliding window as:

$$dT_i^{(L)}[n] = T_i[n] - T_i[n-L], \tag{3}$$

where L is the sliding window's length (see Fig. 1 for an illustration) and index i marks timestamp type (i.e., $i = 1, 2$). Observe that for $L = 1$, i.e. $dT_i^{(1)}[n] = T_i[n] - T_i[n-1]$, (3) agrees with KF from [4]. Note that (3) introduces a significant performance improvement to the former KF definition as proven analytically in Section III-A and experimentally in Section IV-C. Differentiating (2) with respect to (3) we write:

$$dT_1^{(L)}[n] + q_f[n] - q_f[n-L] = (1+\alpha[n])\, dT_2^{(L)}[n], \tag{4}$$

$$dT_1^{(L)}[n] - dT_2^{(L)}[n] = \alpha[n] dT_2^{(L)}[n] + v[n], \tag{5}$$

where (5) is KF's measurement equation, $v[n] = q_f[n] - q_f[n-L]$ is the measurement (zero mean not necessarily Gaussian) noise with variance σ_v^2, and we assume that $\alpha[n] \approx \alpha[n-L]$ thus $\alpha[n]T_2[n] - \alpha[n-L]T_2[n-L]$ in (5) becomes $\alpha[n]dT_2^{(L)}[n]$. This assumption holds as long as the differentiation in (3) happens over short periods of time as explained in the last paragraph of Section II. Further, we define KF's state equation as in [4]:

$$\alpha[n+1] = \alpha[n] + w[n], \tag{6}$$

where the process noise $w[n]$ is tied to the underlying oscillator's noise. Due to scalar form of the measurement and process noises, we have measurement Q and process noise R variances instead of noise covariances of the general model [6]. Note that Q and R represent estimates of the true underlying process and measurement noise variances respectively. The estimation of the measurement noise is shown in Section III-B. We evaluate on how Q affects the skew estimation precision in Section IV-C.

A. Analysis of the Sliding Window Effect

This section analyses the impact of the sliding window on the KF's functionality by means of Signal-to-Noise Ratio (SNR). Let us define SNR for the right hand side of (5) as follows:

$$SNR = \frac{\frac{1}{N}\sum_n \left(\alpha[n] d\widetilde{T}_2^{(L)}[n]\right)^2}{\sigma_v^2}, \quad n = 0 \ldots N, \tag{7}$$

where $d\widetilde{T}_2^{(L)}[n]$ is the clean part of the signal $dT_2^{(L)}[n]$, i.e. $dT_2^{(L)}[n]$ that does not experience any queuing delay $(q_f = 0)$. Now for the sake of argument let's assume that both $\alpha[n]$ and $dT_1^{(1)}[n]$ are constant for all n. Using the fact that $q_f = 0$ in (4) we have $d\widetilde{T}_2^{(L)} = dT_1^{(L)}(1+\alpha)^{-1}$ and thus (7) becomes:

$$SNR = \frac{\alpha^2}{(1+\alpha)^2} \frac{\left(dT_1^{(L)}\right)^2}{\sigma_v^2} = \frac{\alpha^2}{(1+\alpha)^2} \frac{L^2 dT_1^{(1)^2}}{\sigma_v^2}. \tag{8}$$

Note that (8) uses the fact that if $dT_1^{(1)}[n]$ is constant, hence $dT_1^{(L)} = dT_1 L$. It is obvious from (8) that requirements on high SNR and short convergence time are contradictory, since lower MPS yields higher SNR, but increases convergence time.

A direct implementation of KF as defined in [4] (i.e. $L = 1$) yields a frequency error as much as $7e^3$ PPB when used with PTP (see Fig. 4) and hence, increasing the SNR is an absolute necessity. Note that increasing dT_1 interval would result in an unacceptable convergence time as explained above.

A closer examination of (8) indicates that the sliding window length parameter L in (3) has a positive effect on SNR (i.e.,

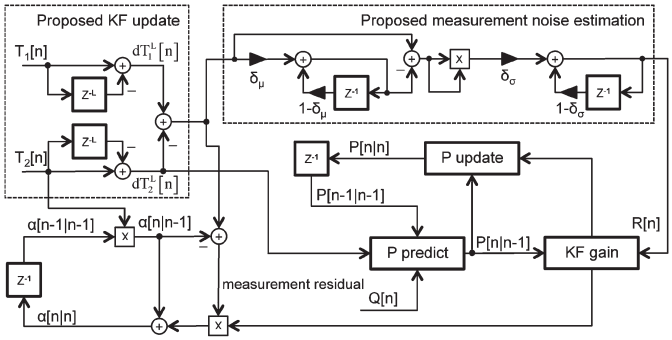

Fig. 2. A schematic flow of our KF algorithm.

SNR is increasing with a factor of L^2). Thus, our slight modification to prior art KF algorithm has introduced a mechanism that improves SNR significantly without loosing the advantage of the short convergence time (higher MPS). Increasing window size L does not affect the statistics of the noise v at all, since from a statistical point of view a random variable $X^{(1)} = X[n] - X[n-1]$ equals random variable $X^{(L)} = X[n] - X[n-L]$. Note that L is recommended to be as large as possible under the condition that $\alpha[n] \approx \alpha[n-L]$ so that difference of (2) still yields (5).

B. Noise Variance Estimator

The paper [4] did not specify how to estimate the measurement noise variance σ_v^2 in practise. A simple estimator using (5) and sample statistics over a window of L samples can be defined as follows:

$$\hat{R}[n] = \underset{k}{var}\left\{dT_1[k] - dT_2[k]\right\}, \quad k = n-L \ldots n, \tag{9}$$

where $\hat{R}[n]$ is the sought noise measurement variance estimate (i.e. estimate of the true underlying σ_v^2). To derive (9) from (5) we use the fact that $\alpha \ll 1$. Since the complexity of (9) is dictated by the window length L, we define here a simpler noise variance estimator based on the fact that expectation operator $E\{\cdot\}$ can be approximated by an Exponentially-Weighted Moving Average (EWMA) filter and thus, variance operator $E\{(x[n] - E\{x[n]\})^2\}$ can be implemented as:

$$\hat{R}[n] = (1-\delta_\sigma)\hat{R}[n-1] + \delta_\sigma \left(x[n] - \hat{\mu}[n]\right)^2, \tag{10a}$$
$$\hat{\mu}[n] = (1-\delta_\mu)\hat{\mu}[n-1] + \delta_\mu x[n], \tag{10b}$$

where $x[n] = dT_1^{(L)}[n] - dT_2^{(L)}[n]$, $\delta_{\mu,\sigma}$ is a smoothing factor $0 < \delta_{\mu,\sigma} < 1$, $\hat{\mu}$ is the EWMA based mean estimator, and (10a) is our low complexity implementation of (9). Notice that (10a) does not depend on window length parameter L. An overview of our KF based clock skew estimation algorithm with measurement noise estimator is depicted in Fig. 2.

IV. Results

A. Simulation Model

In [7] a high precision simulation model for packet networks synchronization was shown that is capable of nanosecond precision with reasonable processing requirements. We have implemented that model and used it to generate timestamps T_1 and T_2 with rate of 128 MPS. The oscillator skew α was set to 10^3 PPB. Queuing delays q_f are generated from measured data set collected using ten Ethernet switches interconnected as per ITU-T G.8261 (page 83) with cross background traffic set to 80/20% load. The statistical details of measured PDVs are summarized in Table I.

TABLE I
Packet Delay Variation Statistics

[μs] $\longrightarrow$	min	max	mean	median	mode	std. dev.
80% load	120	224.8	168.1	167.9	167.7	14.4
20% load	120	184.3	128.7	127.6	120	7.9

B. Prior Art OWM Based Techniques

This section first introduces prior art techniques mentioned in Section I to provide foundations for the experimental evaluation in Section IV-C. Further, a computational complexity of each technique is shown. Let us start with the LP method [4] for which we can write using (2):

$$\alpha[n]T_2[k] + \beta[n] \geq T_1[k] - T_2[k], \forall k \in [n-L \ldots n], \tag{11}$$

subject to minimization of the objective function:

$$f\left(\alpha[n], \beta[n]\right) = \sum_{k=n-L}^{n} \left(\alpha[n]T_2[k] + \beta[n]\right), \tag{12}$$

where $\alpha[n]$ is the sought clock skew, $\beta[n]$ is the estimated one way delay, and L is the window length. Note that according to literature LP's complexity is on average polynomial, i.e., $\mathrm{O}(L^b)$, but also depends on the problem definition and data type [8]. In [5] LR method was defined as follows:

$$\alpha[n] = \frac{L\sum T_1[k]T_2[k] - \left(\sum T_1[k] \sum T_2[k]\right)}{L\sum T_1^2[k] - \left(\sum T_2[k]\right)^2}, \tag{13}$$

where all sums run from $k = n-L \ldots n$. LR's asymptotic complexity derived from (13) is $\mathrm{O}(L)$, i.e., linear complexity. Our KF model's complexity is given by the predict and update phases plus the complexity of the noise variance estimator. Since none of the operations depends on the windowing parameter L (except for memory delays in (3) which do not increase complexity), the overall asymptotic complexity is $c \cdot \mathrm{O}(1)$, where $c \ll L$ for large L. This particular attribute makes KF very attractive for a practical implementation.

C. Analysis of Parameters and Results

Let us first comment on how EWMA filter parameter $\delta_{\mu,\sigma}$ affects KF's clock skew estimate precision. We have tested KF precision over all permutations of $\delta_{\mu,\sigma} \in [1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}]$ set. Although the accuracy and precision slightly deteriorate for $\delta_{\mu,\sigma} \geq 1e^{-2}$, the change is insignificant (less than 0.5 PPB). Hence, as a compromise between EWMA filter convergence time (responsiveness) and accuracy/precision, we further used $\delta_{\mu,\sigma} = 1e^{-3}$ for all tests. Note that in the following text we evaluate clock skew estimation accuracy/precision using clock skew error (ϵ) defined as a difference between the clock skew estimate ($\hat{\alpha}$) and the true clock skew (α), i.e., $\epsilon = \hat{\alpha} - \alpha$.

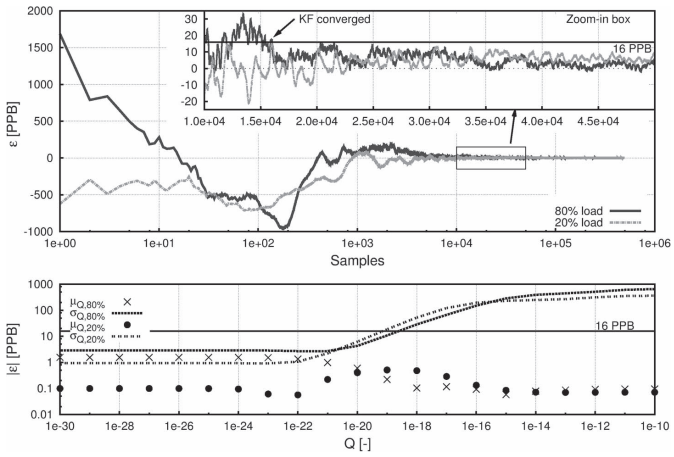

Fig. 3. KF's convergence time (top, $L = 2e^3, Q = 0$) and analysis of the effect of process noise Q setting (bottom, $L = 2e^3$).

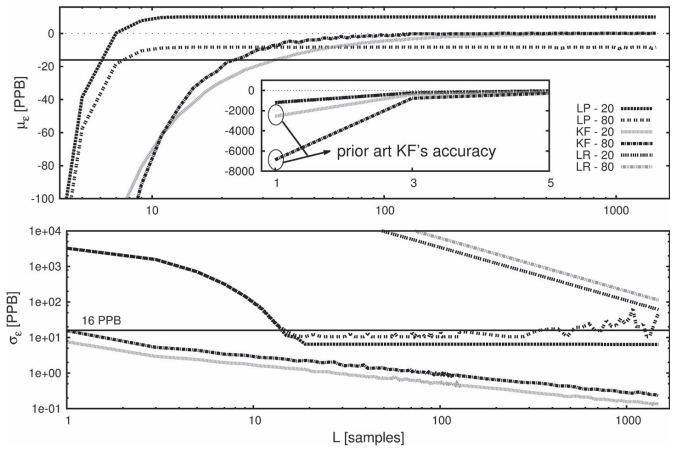

Fig. 4. A performance comparison along with clock skew estimate $\hat{\alpha}$ sensitivity to windowing parameter L for different algorithms ($Q = 0$).

Let us first analyse the KF's convergence depicted in the upper part of Fig. 3 with convergence details in the zoom-in box. KF requires about $1.6e^4$ samples (approx. 4 minutes for 64 MPS) to achieve better than 16 PPB accuracy/precision irrespective of load conditions. Note that the convergence time depends solely on the number of MPS negotiated between the master and slave (KF converges after a certain amount of samples) and thus does not change with L.

The bottom part of Fig. 3 shows that for $Q > 1e^{-20}$ the resulting precision no longer conforms to the LTE-FDD requirements. Setting Q further below $1e^{-22}$ does not have any significant effect on precision or accuracy. Note that having non zero Q value might be necessary, since it allows the KF model to track better the slow changes of the underlying oscillator (due to ageing etc.).

The main contribution of this letter is depicted in Fig. 4 that shows performance comparison of prior art techniques and our KF model. An inherent part of the performance comparison is the analysis of the windowing parameter effect on the accuracy/precision. Note that the data statistics, i.e., mean (μ_ϵ) and s. dev. (σ_ϵ) of error (ϵ), were always computed after the convergence of the KF to avoid any bias. The simulation was repeated ten times for each parameter L. Since LR performs inferiorly to other algorithms, its mean error results were omitted in Fig. 4 for the sake of better clarity.

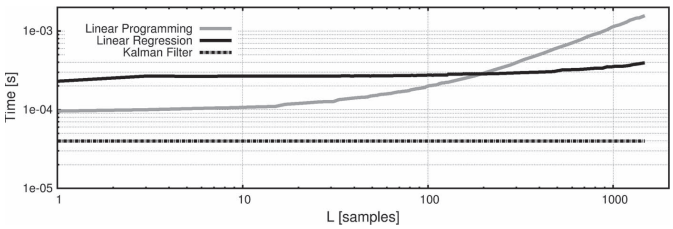

Fig. 5. An average processing time per incoming sample (packet) of different algorithms with respect to windowing parameter L in log scale. Observe that KF's computational time is invariant to L changes as explained in the last paragraph of Section IV-B.

To emphasize the importance of our update, observe the KF's precision for $L = 1$ (this is the original KF model [4]) for which the average estimation error can be as high as $7e^3$ PPB. The updated KF model (for large L, i.e. $L > 2e^2$) yields much lower mean and s. dev. of error when compared to other prior art techniques. Notice that LP method shows significantly higher σ_ϵ for small windows, which is caused by frequent erroneous clock skew estimate due to scarce occurrence of the minimally delayed packet.

Finally, Fig. 5 shows an average computational time per sample (incoming packet) that was measured on the 2.3 GHz Intel Core i7 processor during the windowing parameter testing. Observe that the average computational time grows linearly for LP and LR (note that Fig. 5 is in log-scale), whereas KF shows the best performance consistently, outperforming both LP and LR algorithms.

V. Summary

This letter has presented an updated Kalman Filter model for clock skew estimation using IEEE 1588v2 PTP over one way communication path. The updated model is able to maintain high accuracy (better than 16 PPB) even in the harsh environment of telecom backhaul networks. The simulation results verified that it is impossible to achieve the same precision with other prior art techniques, moreover, it was shown that our Kalman Filter implementation is the most computationally efficient while maintaining the highest accuracy and precision.

References

[1] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE 1588-2008, 2008.
[2] "Timing and synchronization for LTE-TDD and LTE-advanced mobile networks," Symmetricom, Aliso Viejo, CA, USA, 2013.
[3] L. Cosart, "Characterizing grandmaster, transparent, and boundary clocks with a precision packet probe and packet metrics," in *Proc. IEEE ISPCS*, 2011, pp. 56–61.
[4] A. Bletsas, "Evaluation of Kalman filtering for network time keeping," *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 52, no. 9, pp. 1452–1460, Sep. 2005.
[5] L. Ma *et al.*, "Impact of linear regression on time synchronization accuracy and energy consumption for wireless sensor networks," in *Proc. MILCOM*, 2008, pp. 1–7.
[6] S. J. Orfanidis, *Optimum Signal Processing, An Introduction*, 2nd ed. New York, NY, USA: McGraw-Hill, 1988.
[7] Z. Chaloupka *et al.*, "Efficient and precise simulation model of synchronization clocks in packet networks," in *Proc. IEEE CAMAD*, 2013, pp. 79–83.
[8] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *J. ACM*, vol. 51, no. 3, pp. 385–463, May 2004.