

Accurate Clock Discipline For Long-Term Synchronization Intervals

Borja Martinez, *Member, IEEE*, Xavier Vilajosana, *Senior Member, IEEE*, and Diego Dujovne, *Member, IEEE*

Abstract—Efficient spectrum usage and optimized energy consumption directly depend on the radio duty cycle of the communicating devices. Tight synchronization of communicating nodes enables optimal orchestration of the access to the medium as nodes turn their radios on precisely when needed. Achieving synchronization, however, requires a reference time source to synchronize with and a periodic discipline mechanism to cope with the inherent clock drift. Packet-based synchronization or external time sources, such as GPS, are usually required to achieve that goal. When low-power operation is combined with bandwidth limitations and scale, as in the case of low power wide area networks, the overhead of such approaches is not affordable and advanced clock discipline mechanisms are required. In this paper, we propose a novel adaptive mechanism to discipline clocks in order to guarantee a 1ppm drift with minimal communication overhead.

Index Terms—Synchronization, clock discipline, drift correction, low power operation, LPWAN.

I. INTRODUCTION

THE emergence of Low Power Wide Area Network (LPWAN) technologies is accelerating the massive deployment of Machine-Type operated networks providing seamless communication services. LPWANs are a key answer for large sensor deployments, as smart metering and other smart city applications, on which low-power operation is critical, while communication requirements are loose and time-insensitive. The latest LPWAN technologies such as LoRaWAN [1], Ingenu [2] or the technology developed by Sigfox [3] make use of very robust modulations, exploiting narrow-band communications or spread spectrum techniques such as Chirp Spread Spectrum (CSS). This ensures long range links at low data rates, supported by powerful decoders at reception and avoiding multi-hop topologies. Currently, most of these technologies use simple Aloha medium access, subject to contention and limited performance under congestion to around 18% [4].

Manuscript received December 12, 2016; accepted January 25, 2017. Date of publication February 1, 2017; date of current version March 8, 2017. This work was supported by the Spanish Ministry of Economy and the FEDER Regional Development Fund under the SINERGIA Project under Grant TEC2015-71303-R. The associate editor coordinating the review of this paper and approving it for publication was Prof. I Koo.

B. Martinez is with Universitat Oberta de Catalunya, 08860 Barcelona, Spain (e-mail: bmartinezh@uoc.edu).

X. Vilajosana is with Universitat Oberta de Catalunya, 08018 Barcelona, Spain, and also with Worldsensing, 08013 Barcelona, Spain.

D. Dujovne is with the Escuela de Informática y Telecomunicaciones, Universidad Diego Portales, Santiago 8370191, Chile (e-mail: diego.dujovne@mail.udp.cl).

Digital Object Identifier 10.1109/JSEN.2017.2662323

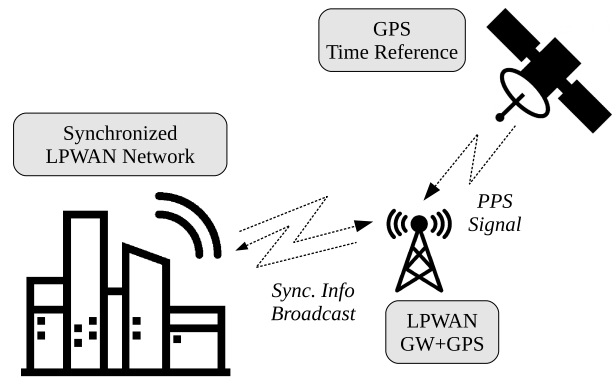


Fig. 1. Exemplifying a synchronized LPWAN network where a single time source reference (GPS) is used for a wide range of devices to align their clocks.

LPWANs are subject to strict duty cycle regulations [5], which impose limits in the number of packets to be transmitted per hour. This limitation, combined with the Aloha access, clearly affects the network performance when scalability is a requirement [6].

In order to improve its performance and, in addition, to optimize its energy consumption, synchronization appears as the most efficient choice. Synchronization enables nodes to schedule their transmissions and hence operate without collisions, mitigating the under-performance of contention-based MAC approaches. Yet synchronization enables optimal energy consumption as transmitting devices do not need large guard times or preambles to communicate. Since nodes are synchronized, they already know when to send or receive.

Synchronization in a network requires aligning the clock of the nodes to a time reference. This time source can be a remote node or an external time reference such a GPS (as depicted on Fig. 1). In such approach, the nodes' individual clocks are synchronized and re-aligned periodically to compensate for crystal clock natural drift, temperature dependence and inherent variability. These combined effects accumulate a significant clock misalignment in long sleep periods, which need to be compensated to reduce communication guard times or to increase the drift compensation period, as will be discussed in Section II. Moreover, within the context of low-power devices, an obvious goal is to minimize the synchronization overhead in terms of transmitted packets and energy consumption, while maximizing the synchronization accuracy.

Most common low power TDMA (Time Division Multiple Access) approaches use a periodic signal to align each node's

clock. In these approaches, nodes sleep during the inter-packet period and hence need to be waken up sufficiently in advance to receive the packet even with some drift. The latter forces the guard time to be longer than the clock drift for the inter-synchronization period.

Synchronization through pairwise communication is achieved by determining the reception offset of a packet with respect to the expected receive time. This offset must be smaller than a MAC layer guard time in order to receive the packet and hence, the drift for the synchronization period must be smaller than that guard time [7]. The most usual techniques are based on packet synchronization or acknowledgement-based synchronization. In the former, the time source node sends a packet and the receiver node determines the reception offset with respect to the ideal time; this difference is the amount of ticks to be corrected. The latter uses the time source node acknowledgement to notify the drift to the node being synchronized.

In the case of LPWAN technologies, the duty cycle regulation limits the amount of packets to be transmitted to a few packets per day [8]. Due to the former limitation, the drift accumulated for the inter-synchronization period requires large MAC layer guard times which impacts the low energy consumption boundary of the technology. These two limitations constitute a trade-off between the synchronization period and the guard time size, i.e. the larger the synchronization period, the larger the guard time and hence the larger energy consumption. The key element is to minimize the clock drift.

In this article, we propose a method to discipline the crystal clock frequency to reduce the long-term variability, with our target set in the 1 ppm (parts-per-million) horizon for the whole operational temperature range.

As in TDMA, this method relies on eventual clock realignment to a time source reference (through pairwise packet exchanges), enhanced in our case with an adaptive drift rate prediction and compensation.

Specifically, we propose the introduction of statistical self-learning tools to infer a characteristic thermal-drift compensation curve for each device. These tools allow us to combine the information learnt during calibration with information self-learned in running time, providing an automatic and smooth transition between each other.

Our ultimate goal is to set the basis and enable LPWAN synchronization without jeopardizing the network capabilities in terms of traffic and low energy consumption boundaries.

The article is organized as follows: Section II defines the main challenge; Section III summarizes the former work in the area; Section IV explains in detail the proposed method; Section V describes the simulation results and finally Section VI concludes this work.

II. QUANTITATIVE PROBLEM STATEMENT

The frequency of a crystal oscillator is affected by changes in the ambient temperature. These changes in the temperature can affect the value of any of the components which comprise the oscillator circuit. If these components variations do not cancel each other, a change in the nominal operating

frequency will result. The frequency determining component most severely affected by any temperature change is the quartz crystal [9].

Tuning forks have become the dominant type of low-frequency crystal due to their small size, low cost, low power dissipation (including the oscillator) and high stability (temperature, aging, shock and attitude) [10]. Other alternatives, such as temperature compensated crystals (XOTC), ensure reduced drift in the order of 1-5 ppm, but increase significantly hardware cost and power consumption.

Typically, the internal clock in a node is driven by a low-frequency tuning fork quartz crystal, being 32.768 kHz the most common frequency. The accuracy associated with this type of clock source is characterized by two main parameters. First, the *frequency tolerance*, a measure of crystal variability between different samples. This tolerance is mainly due to fabrication and crystal inherent imperfections and is typically in the order of tens of parts per million (ppm) for low-cost crystals. Second, *frequency stability*, a measure of temperature-dependent drift variability, as temperature modifies the resonant behavior of the oscillator circuit. This drift is typically rated from ten ppm accuracy in indoor conditions, while increasing with temperature range, raising up to hundreds of ppm for nodes placed outdoors exposed to severe environmental conditions.¹

A. Clock's Behaviour With Temperature

As a starting point in our evaluation, we first analyzed the temperature evolution in a mild climate area for a period of 3 months (See Fig. 2). The objective was to estimate the actual drift in real operating conditions due to temperature variations.

In detail, Fig. 2 presents the clock drift accumulated during 24h for operating devices exposed to different daily temperature profiles (Fig. 2a). Despite the apparently low drift values rated, this drift rate has a strong effect when accumulated over long time spans. For example, a 10ppm device in mild weather, (0°C- 30°C), can accumulate over 1s drift within 24h (Fig. 2b).

This clock drift has an important impact on the synchronization of the two devices, as they may have drifted, in the worst case, more than 2s apart after 24h of operation. When these two nodes communicate, they need to turn on their radio considering that drift, i.e. respecting guard times above 2s, which may be unacceptable. In a TDMA approach, the size of the guard time directly impacts the slot length and therefore the available bandwidth, energy consumption and network capacity. Clearly, one desirable goal is to reduce the drift so communication requirements can be minimized and, at the same time, keep guard times small to reduce energy consumption.

B. Disciplined Clocks

In general, for a disciplined-clock device, synchronization accuracy has a two-region behavior [12] (see Fig. 3). In the

¹See, for instance, the ABS07-120-32.768kHz-T tuning fork crystal data-sheet, used as the reference in this work.

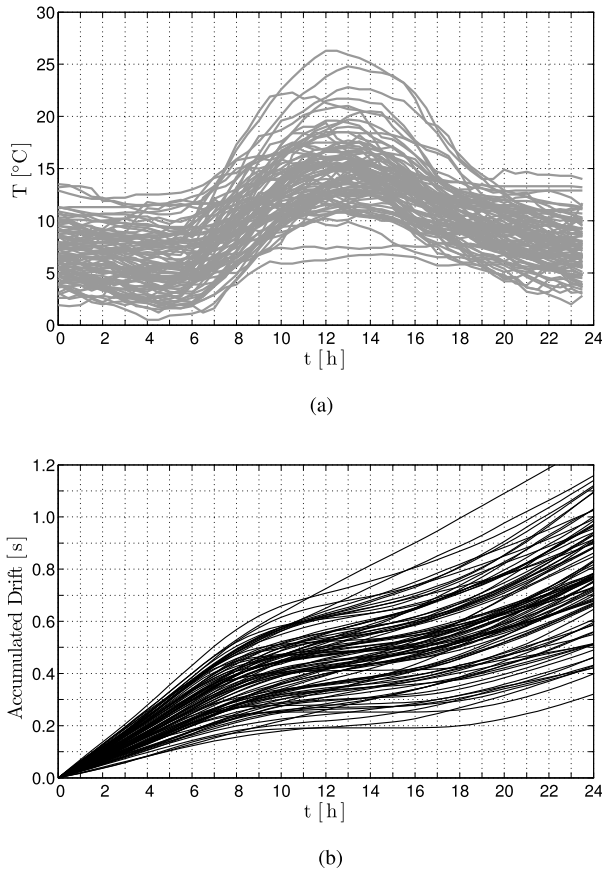


Fig. 2. Outdoor temperature daily profile in a mild climate city (temperature record available at [11]) and the impact on the crystal drift. (a) Daily temperature variability in a mild climate area over 3 months. (b) Accumulated Drift due to temperature variation (24 hours span).

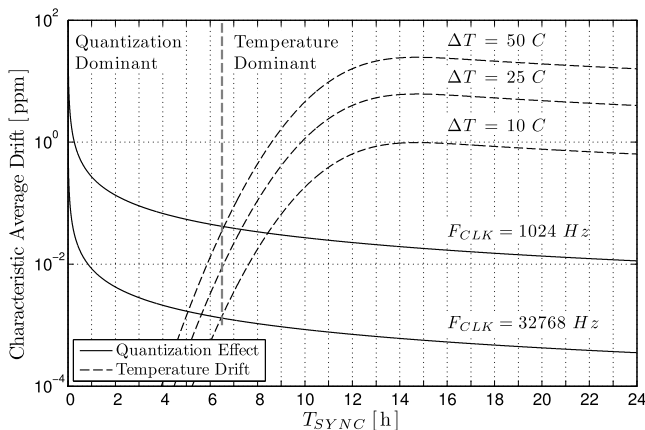


Fig. 3. Characteristic frequency drift, comparing the main sources of clock inaccuracy: temperature and time quantization due to integer timers. The quantization effect is dominant for short synchronization intervals. However, for longer periods, the drift due to temperature becomes dominant.

first region, the accuracy is limited by the quantization error, which is directly related to the integer nature of timers [13] and the accuracy of internal sampling. When a device uses radio messages to align its clock drift to a time source device, the integer steps of timers generate a quantization error for the timestamp. From [12], the quantization error can be

characterized by Eq. (1):

$$\delta_Q(T) = \frac{1}{f_0 \cdot \tau_{SYNC}} \quad (1)$$

Faster timers (higher f_0) minimize this effect (See Fig. 3, on which a 1 kHz system timer is compared with a 32.768 kHz clock), as a tick represents a smaller portion of time and therefore the absolute error is smaller [12]; but they are not appropriate for the node's sleep mode, given their high energy consumption. Alternatively, this error can be minimized when the timestamp difference is obtained from longer intervals (longer τ_{SYNC}), causing a reduction of the relative error, as the quantization step is constant. However, this leads to the second limitation.

In the second region, the impact of temperature changes on achievable accuracy becomes dominant. In this region, longer periods between radio synchronization messages (higher τ_{SYNC}) imply higher drifts, since the cumulative effect of temperature variations increases with time. Fig. 3 shows the simulated accumulated drift for three different temperature profiles (with a temperature span ΔT of 10°C, 30°C and 50°C, respectively). It is clear from this figure that there is a need for an efficient temperature correction mechanism when synchronization intervals become longer.

The two aforementioned technological constraints drive how the synchronization problem has been addressed in the literature, as detailed in the following section.

III. RELATED WORK

Since the quartz crystal is an anisotropic medium, the resonance frequency is highly dependent on the crystallographic orientation, the so-called angle of cut. According to the systematic investigation of [14], the frequency versus temperature behavior can always be sufficiently approximated by a third-order polynomial $\frac{\Delta f}{f_0} = a_0 + a_1 \Delta T + a_2 \Delta T^2 + a_3 \Delta T^3$, where $\Delta T = T - T_0$ and $\Delta f = f - f_0$, being f_0 the frequency at a reference temperature T_0 .

For the specific case of tuning fork crystals, towards which we focus this work, it turns out that the frequency-temperature-characteristics behaves as a parabola. This is an intrinsic property of this type of cut [15]. Typically, the angle of cut of the resonator used in this kind of crystals is such that the zero temperature coefficient (symmetric point of the parabola) is at $T_0 = 25^\circ\text{C}$.

Taking all these facts into consideration, the relative frequency drift versus temperature of the 32.768 kHz crystal oscillators of our interest can be modeled by a simple second order polynomial given by Eq. (2), where δf is defined as $\delta f \doteq \Delta f / f_0$.

$$\delta f = K(T - T_0)^2 + M_0 \quad (2)$$

In Eq. (2), the temperature coefficient K governs the impact of temperature in the drift, i.e. the frequency stability. Typical values are around $-0.035\text{ ppm}/^\circ\text{C}^2$ [15]. In turn, M_0 accounts for production variability, i.e. the frequency tolerance (see [16] for a brief discussion). Generally, the main goal is to first compensate the latter effect, eliminating M_0 through an individual calibration process, and also periodically measuring

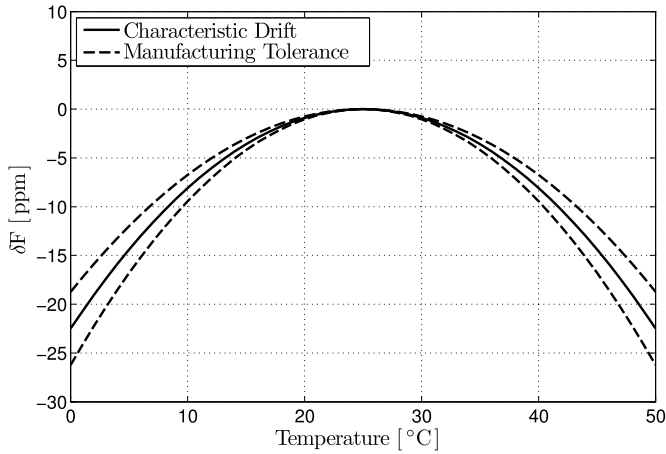


Fig. 4. Manufacturing tolerance. This plot shows the temperature-drift characteristic of Eq.(2) once the offset M_0 has been removed. The dashed lines represent the drift confidence, due to the tolerance of the temperature coefficient $K \pm \Delta K$ specified by the manufacturer (Min/Typ/Max values) of our reference crystal ABS07-120-32.768kHz-T.

the temperature to apply small corrections based on the drift-temperature curve.

To that end, Brunelli *et al.* [17] propose a pre-deployment calibration mechanism common for all devices. This correction is inferred in the design stage from a sample of sensors by fitting a 2nd order polynomial to the empirical devices' behavior, obtaining an averaged characteristic curve. This approach suffers from several drawbacks. First, it assumes a temperature dependence equal for all crystals of the same class. Manufacturing variations from crystal to crystal (parametrized by M_0) and component ageing effects are not addressed. Second, even after correcting the manufacturing offset M_0 , deviations remain in the temperature coefficient K . These variations can cause drifts above 10 ppm at the extremes of the range even for the same temperature, as shown in Fig.4 (See also [17]).

To cope with these limitations, Schmid *et al.* [12] proposes a dynamic look-up table (LUT) approach to compensate the clock drift. The table maps pairs of temperature-drift values measured locally at each node. The authors suggest a two-stage operation: a calibration stage,² on which nodes learn the drifting pattern; and the operation stage, on which nodes apply the previously learned pattern according to the temperature readings. The main drawback of this approach relates to the inherent inaccuracy in the determination of temperature-drift pairs. First, this empirical error is mainly attributed to the duty-cycled operation of the temperature sampling process (typically, an average temperature should be used as the value representative of the entire time interval between two synchronization events). Second, the limited accuracy of low-cost sensors can have a significant impact on the performance of the algorithm, as discussed in Section V-C.

In practice, this means that the entries of the LUT table may be corrupted by the error of the temperature measurement,

²In Schmid *et al.*, the term "calibration" differs from our nomenclature. In their case, it applies to a learning phase after deployment of the devices that is carried out in the field, but before normal operation. In contrast, in our case, the calibration is considered at design/production time.

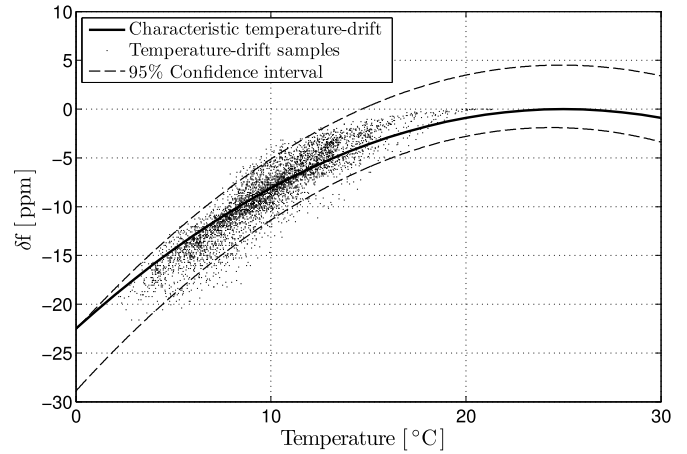


Fig. 5. Error due to temperature sampling. In addition to the sensor accuracy, this error is mainly associated with temperature variations within the inter-sample intervals of time, as the temperature is sampled at discrete instants. The confidence interval is computed according to Eq.(7).

and this type of noise imposes a limit to the accuracy of the corrections. This impact has been simulated in Fig.5 based on a real temperature profile. Interestingly enough, the calculated error has the same order of magnitude as the crystal's tolerance (approx. 10 ppm). Finally, ageing is not addressed, as it is not clear how the LUT is updated after the calibration stage. Therefore, this approach can not cope with long lifespan applications.

An alternative, and radically different approach, requires a time source node to synchronize to (commonly referred to as *discipline*). The synchronization is achieved by determining the offset with respect to a relative timestamp. The time source node measures the drift and notifies the variation with respect to that timestamp to the counterpart which corrects it. Assuming eventual pairwise synchronization between slave nodes and time source neighbor or base station, the slave nodes keep track of their time corrections and the interval between them. Taking into account these two parameters, a drift rate is calculated and applied. This approach, with minor variations, has been widely used in the literature (see [13], [18], [19], for instance). This method requires short time updates to avoid large temperature fluctuations between corrections. For this reason, the main drawback for these solutions is related to the integer nature of timers, both physical (HW) and virtual (RTOS sys-tick), as mentioned previously. In addition, the clock frequency is not smoothly adjusted and the clock interval is cropped. This may be unacceptable in some applications, as those requiring a constant sampling rate for further signal processing.

Of particular interest is the work from Stanislawski *et al.* [20], who developed a drift rate predictive algorithm on which discipline is optimized on the basis of local temperature, thus combining the two approaches. As long as the temperature is stable, nodes keep correcting their drift based on the last learned value, thus reducing the number of pairwise synchronizations. This resource has been exploited by several authors in the literature (see [16] for

Algorithm 1: Drift-Temperature Fitting**Initialization:**

Reset the current timestamp and set a default time offset $\tau_{SYNC}^{(0)}$ for packet reception.

Loop:1) **Discipline:**

- Wait for the i -th synchronization packet. Read the actual time elapsed $\tau_{SYNC}^{(i)}$.
- Get the local timestamp upon reception of the packet. Measure the local time elapsed $\tau_L^{(i)}$.
- Align clock to the time reference.

2) **Learning:**

- Determine δ_i as the difference between the local timestamp $\tau_L^{(i)}$ and the real time elapsed $\tau_{SYNC}^{(i)}$.
- Determine T_i as the average of the N temperature samples $T_{j=1\dots N}^{(i)}$ acquired since the last synchronization packet ($i-1$). Empty temperature queue.
- Update learning table with the pair $\{T_i, \delta_i\}$. Recalculate the fitting curve and update the values of the polynomial coefficients $\hat{\beta}^{(l)}$.

End**Algorithm 2: Drift Prediction****Loop:**1) **Temperature sample:**

- Get current temperature T_0 .
- Enqueue T_0 as $T_j^{(i)}$, the j -th temperature sample in the i -th (current) synchronization cycle.

2) **Drift prediction:**

- Evaluate calibration confidence $[\delta f_L^{(c)}, \delta f_H^{(c)}]$ at T_0 .
- Evaluate learning confidence $[\delta f_L^{(l)}, \delta f_H^{(l)}]$ at T_0 .
- Estimate drift $\delta f^{(x)}(T_0)$, selecting the model from the best confidence interval at T_0 :

$$\delta f^{(x)}(T_0) \Big| x = \underset{j}{\operatorname{argmin}} \|\delta f_H^{(j)} - \delta f_L^{(j)}\|, x \in c, l$$

End

instance). However, when temperature varies abruptly, this method forces a pairwise synchronization to cope with the foreseeable drift variation due to the temperature change. Consequently, this approach requires a high number of synchronization packets when the temperature variations are abrupt and frequent, making the approach unfeasible in LPWAN technologies exposed to sudden and extreme temperature changes.

IV. ALGORITHM DESCRIPTION

In this article, we propose a discipline algorithm to address fine-grained drift compensation.

The proposal takes advantage of more recent advances in this topic by combining: a) *Discipline*: eventual packet-based clock alignment to a time source reference, b) *Drift prediction*: periodic drift estimation based on a drift-temperature model of the crystal and c) *Learning*: Adaptive run-time fitting for model progressive refinement.

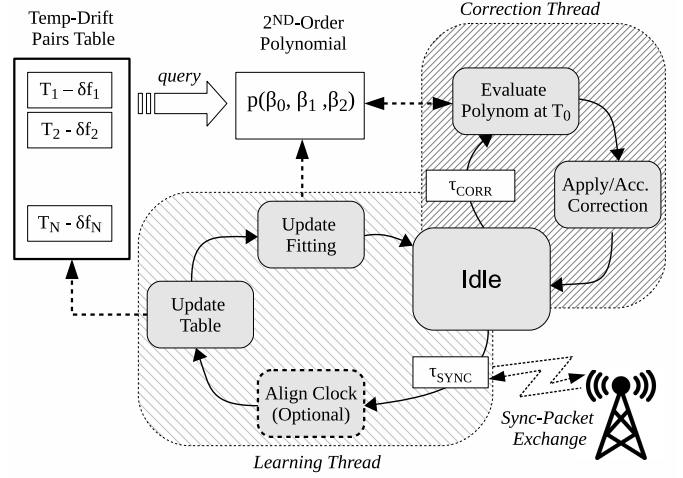


Fig. 6. Drift-temperature compensation algorithm flow chart.

In addition, the temperature-drift relation is pre-calibrated to speed-up the convergence of the fitting algorithm. To combine the two approaches (pre-calibration and dynamical fitting), we propose a novel mechanism that allows an automatic transition from the calibrated to the self-learning drift model. The rationale of the method is to take advantage of sparse packet-based synchronization events to improve the calibration information, and to progressively refine a drift-temperature fitted model according to the real measured behavior.

A. Discipline, Learning and Drift Prediction

Fig. 6 presents a simple flow chart sequencing the algorithm operation. When a synchronization packet from the time source reference is received, the number of drifted ticks since the last synchronization event is determined (as shown in Alg. 1). This can be achieved through different means, for example, by comparing the local time elapsed τ_L with the real time τ_{SYNC} , indicated in the packet payload. This value may be used to discipline the clock, i.e., to realign the local clock to the time reference source, by adding this value to the local timer-counter.

This information is also used to update a *learning table*, which serves as base for the subsequent model fitting. The learning table consists of a set of pairs $\{T_i, \delta_i\}$ with T corresponding to the average temperature for the inter-packet period and δ the measured drift. The average temperature is obtained as the mean value of all acquired temperature samples (with period τ_{CORR}) within the time interval between two synchronization packets (τ_{SYNC}). The temperature-drift characteristic function $\delta f(T)$ is obtained by quadratic least-squares fitting of the values stored in the table. To that end, Eq.(5) (Appendix VI) is solved using the pairs $\{T_i, \delta_i\}$ as support points. This results in some coefficients $\hat{\beta}$, identified with the model parameters M and K in Eq.(2), which define the general temperature-drift model:

$$\delta f(T) = \hat{\beta}_0 + \hat{\beta}_1 T + \hat{\beta}_2 T^2 \quad (3)$$

The drift correction is managed in a concurrent thread or process as detailed by Alg. 2. During the operation of a node, while waiting for synchronization packets, the node periodically checks the temperature to estimate the current drift rate according to the polynomial fitting (with the aforementioned interval period τ_{CORR}). To do this, the measured value T_0 is evaluated on the fitting polynomial function $\delta f(T_0)$ to obtain the characteristic drift δ_0 at that temperature (Eq. (6), Appendix VI).

At this point, the node can correct the clock phase by adding some tics corresponding δ_0 . However, to further increase the accuracy, the drift compensation value can be accumulated during the inter-packet synchronization period, thus mitigating the undesirable effects of quantized timers when applying short-term corrections (See Eq.(1) and Fig.3). In fact, for long-term corrections, the quantization effect is almost negligible as explained in Section V-C.

Hence, taking advantage of the derived fitting curve, a device is able to forecast its drift locally, and make use of the predicted error to schedule future actions (e.g. wake-up to receive a radio message) following, for example, the approach described in [20].

B. Transition From Calibration to Self-Learning

In a real implementation of any learning-based predictive mechanism, the bootstrapping phase is prone to critical errors due to the lack of a meaningful statistical base. This is especially significant in systems where synchronization packets are sparse and the learning process is slow.

To increase the reliability of the mechanism, the devices in our model start-up with a pre-calibrated drift-temperature model. This model is obtained from a subset of devices, whose thermal behaviour is characterized during the manufacturing process (following a similar procedure to that from [17]).³

As stated before, we propose a method that provides a smooth transition from the pre-calibrated to the self-learning operating mode. This method makes use of the well-known statistical methods based on confidence intervals.

A set of pairs $\{T_i, \delta_i\}$ obtained by calibrating N different devices is merged on a *calibration table*, which contains all the pairs $\{T_i, \delta_i\}_{j=1\dots N}$ from all the calibrated nodes. By solving Eq.(5) for this set of values, a joint calibration model is obtained. This results in some coefficients $\hat{\beta}^{(c)}$, determining the *calibration curve*:

$$\delta f^{(c)}(T) = \hat{\beta}_0^{(c)} + \hat{\beta}_1^{(c)}T + \hat{\beta}_2^{(c)}T^2$$

These values are fixed and therefore hard-coded on each device. Moreover, Eq. (8) gives the confidence intervals for this model, which are also stored at the device's memory (in what follows *calibration confidence*).

As described in the previous section, a local table, referred as *learning table*, is filled and updated dynamically by pairs of $\{T_i, \delta_i\}$. This table allows for a dynamic fitting that is re-evaluated on-the-fly each time a new synchronization event

arrives. This fitting, characterized by a set of parameters $\hat{\beta}^{(l)}$, gives an alternative model or *learned curve*:

$$\delta f^{(l)}(T) = \hat{\beta}_0^{(l)} + \hat{\beta}_1^{(l)}T + \hat{\beta}_2^{(l)}T^2$$

with its associated *learning confidence*.

In summary, for each measured value of temperature T_0 , Eq.(6) gives an expected value of the drift rate $\delta f^{(c)}(T_0)$ based on the calibration curve, and Eq.(8) a maximum and minimum value of the confidence interval around this value: $[\delta f_L^{(c)}, \delta f_H^{(c)}]$. In turn, Eq.(6) gives an expected value of the drift rate $\delta f^{(l)}(T_0)$ based on the learned curve, and Eq.(8) a maximum and minimum value of the confidence interval around this value: $[\delta f_L^{(l)}, \delta f_H^{(l)}]$.

Then, for each temperature sample, the two available intervals are compared at the current temperature. The narrower interval determines the model to use (*c* or *l*), and the expected value is obtained from the selected model (i.e. from $\delta f^{(l)}(T)$ or from $\delta f^{(c)}(T)$). In this way, the transition from calibration-based corrections to self-learned based corrections is performed in a smooth, completely transparent manner.

To illustrate how this transition works, Fig.7 presents different stages during the learning process of a device. In this figure, the calibration confidence (light grey), the learning confidence (mid grey) and the overlap region (dark grey) are represented, for any temperature in the selected range. The span of the intersection is an indication of the accuracy of the mechanism, as it represents the best confidence interval for a given temperature.

For instance, Fig. 7a presents the overlap of the confidence intervals after 12h from initialization. Let's consider that we want to determine the drift for a temperature $T_a = 16^\circ\text{C}$. For T_a the confidence interval derived from the learning model is wider than that of the calibration model. Therefore, the prediction derived from the calibration phase should be used.

Fig. 7b and Fig. 7c show the evolution after 24h and 48h respectively. As the *learning table* is filled with real drift samples, the overlap area is narrowed, thus improving the accuracy of the mechanism. In Fig. 7c, for example, we can see that the drift prediction for T_a would be more accurate when the *learned curve* is used.

V. RESULTS

First of all, it is important to recall that, when no synchronization technique is used, the accumulated drift for a 24h period can easily exceed 1 second per day (See Fig. 2b). This value sets an upper-bound on our discussion.

As a starting point in our evaluation, we analyzed the accuracy of the drift-temperature pairs on the basis of a temperature profile obtained experimentally during a period of 3 months (Fig. 2a). Additionally, we obtained the characteristic drift for the device during this period. Fig.5 shows the temperature-drift pairs and best fit for this operation period. This figure illustrates how this fitting curve acts as a filter compensating for the empirical errors. These deviations, mainly associated to the temperature variability during the inter-synchronization period and to the inherent temperature sensing error, are characterized by their random nature. This is an inherent

³Engineering testing is a fairly common industrial process which is performed over a sample of devices. These tests are carried out during production or at early deployment stages.

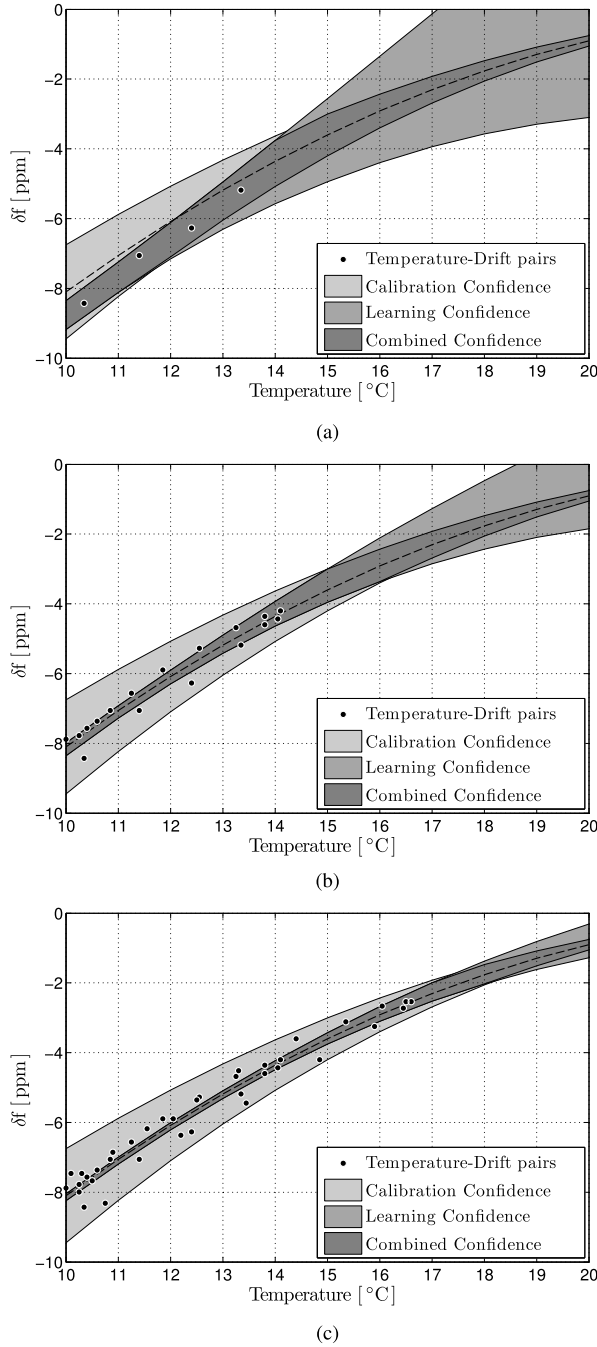


Fig. 7. Evolution of self-learned confidence in comparison with calibration confidence. In the figures, the medium-gray region represent the prediction confidence interval, which is dynamically updated as more real temperature-drift pairs are gathered (black dots). Likewise, the light-gray region is the calibration confidence interval, which is pre-calibrated on the device and thus static. The dark-gray region is the combined confidence, i.e., the best confidence among both intervals for each temperature, evaluated at each instant of time. The figure shows that, as more synchronization packets arrive, the confidence interval gets narrower out-dating the calibration information. (In this figure, we are only showing the points of the areas of interest.) (a) Confidence intervals after 12 h. (b) Confidence intervals after 24 h. (c) Confidence intervals after 48 h.

limitation of any temperature sensitive system and, ultimately, it is the main motivation of the presented method.

Then, in order to evaluate quantitatively the benefits of the proposed method, a set of simulations have been conducted. We assumed a set of devices keeping synchronization to a time

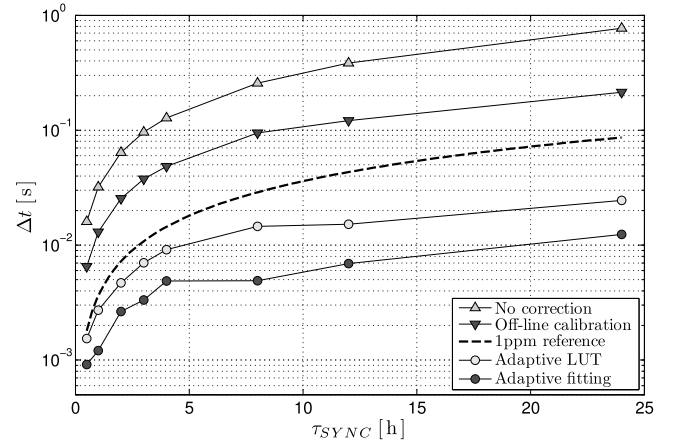


Fig. 8. Accumulated drift along inter-synchronization periods. Time misalignment Δt between the reference clock (network manager) and local clock (device), accumulated between two consecutive synchronization packets (τ_{SYNC}), for different approaches.

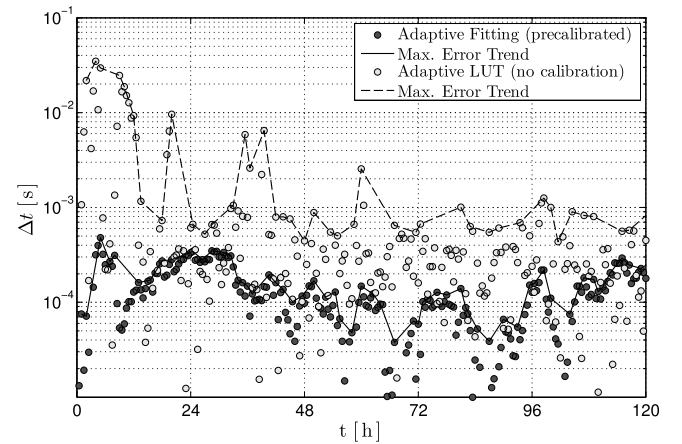


Fig. 9. Tracking of the accumulated drift during inter-correction periods. Evolution of time misalignment Δt , accumulated between temperature sample intervals ($\tau_{CORR} = 0.5$ h). For illustrative purposes only, the trend of the maximum error (i.e., the envelope of the local maxima) has been depicted in the figure.

source (Fig. 1). We defined different synchronization periods between devices, compatible with LPWAN requirements, typically periods of one packet every 1h to one packet every 24h, and computed the worst drift case for these periods. In our setup, temperature is sampled at a constant rate with 30 minutes period. For the LUT-based techniques evaluated, we round the temperature values to the nearest integer, in order to limit the number of inputs and thus the size of the table (the temperature is used as the index for the drift values stored). Thus, we consider temperature sensors with an accuracy $\leq 1^\circ\text{C}$.

A. Drift Prediction

Fig. 8 presents the simulation results for the evaluated synchronization techniques. The first simulation shows the drift accumulated between synchronization events when no drift correction is applied. The drift reaches almost 1 s for synchronization intervals of 24 h (confirming our first estimations done in Fig. 2b).

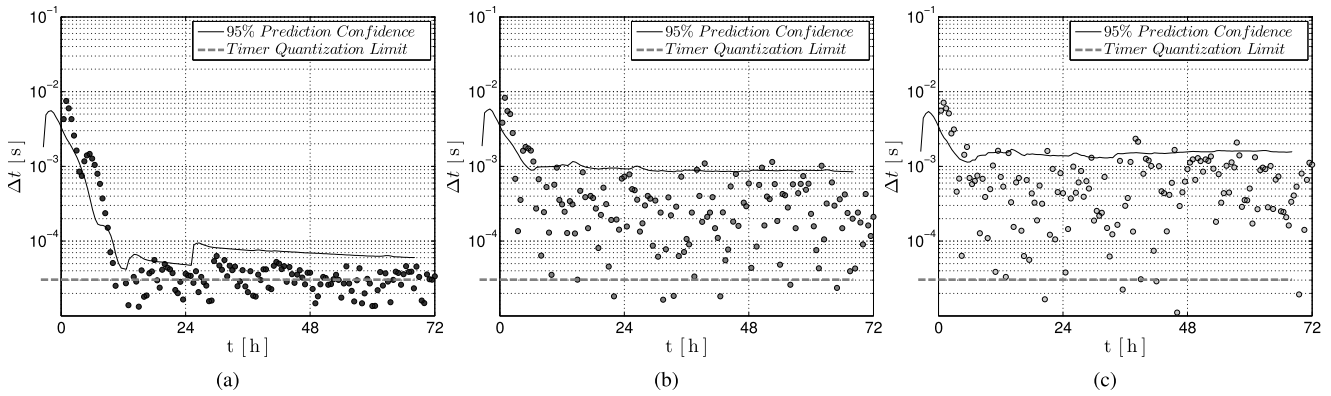


Fig. 10. Impact on drift prediction (deviation between prediction and actual drift) for different temperature sampling accuracies. For the idealized case of perfect accuracy (left), the algorithm converges to the quantification error of the timer. (a) Temperature Accuracy: $\pm 0.0^\circ$ C. (b) Temperature Accuracy: $\pm 0.5^\circ$ C. (c) Temperature Accuracy: $\pm 1.0^\circ$ C.

Second, when introducing a generic calibration table with characteristic drift/temperature mapping (following [17]), the synchronization accuracy increases significantly by close to one order of magnitude. Despite of that, the drift is still too large to maintain LPWAN synchronization. The latter is attributed mainly to the inaccuracies of using a non-device specific calibration (note that the manufacturing tolerances for commercial crystals are typically in the order of 10 to 20 ppm).

As stated before, adaptive synchronization approaches use each pairwise synchronization event to refine the temperature-drift characterization. The approach, taken by [13], [16], [20], leads to accuracy improvements of one order of magnitude for long synchronization intervals, as shown in Fig. 8. Still, there are important inaccuracies due to the temperature variations between measurements (see Fig. 5), not taken into account by previous approaches.

The *Adaptive Polynomial fitting* approach presented in this article aims to address that inaccuracies by inferring a continuous model. This approach filters the inherent error on the values of the temperature-drift table. As a consequence, our approach yields very significant results with a maximum of 10ms drift in 24h for the evaluated conditions. This is close to 0.1 ppm clock drift in the simulated scenario.

B. Bootstrapping

We also performed another simulation to verify the bootstrapping phase of the algorithm. The goal is to evaluate the benefit of using a calibration information base versus not using it. Fig. 9 presents the drift evolution for two simulation approaches, one using a calibration information base and the other without that knowledge.

As it can be observed, during the first 72h, the absence of initial information in the learning table causes several abrupt corrections, which are clearly minimized when starting with pre-calibrated, yet non-customized information. However, the use of a calibration base enables the stabilization of the network at bootstrap. Indeed, an unstable behaviour at the bootstrap phase may cause a node to not be able to synchronize for a long period. In the long term, both approaches converge to similar stability as the learning information becomes dominant.

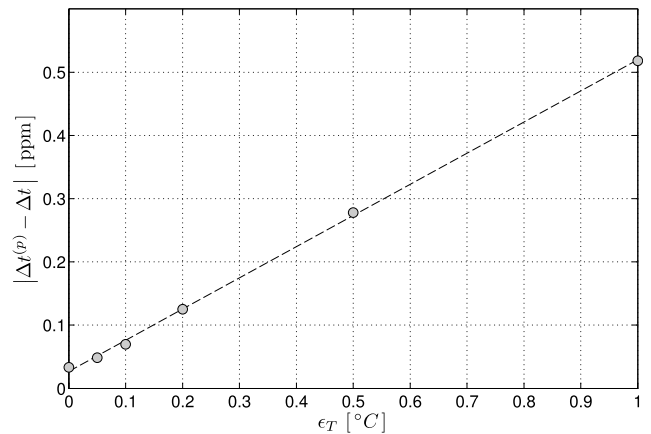


Fig. 11. Additional error on drift prediction induced by the temperature sampling error ϵ_T . The extra error is evaluated as the difference between predicted drift $\Delta t^{(p)}$ and actual drift Δt .

C. Predictability and Temperature Sampling Accuracy

The accuracy of the temperature sensor and, in general, any error related to the temperature sampling process, can impact significantly on the performance of any predictive algorithm that takes temperature as the basis for correction. In our particular case, we have conducted several simulations to analyze this effect on the presented results.

Fig. 10a shows the performance of the algorithm supposing perfect accuracy of the temperature sensor. In this idealized scenario, the algorithm converges to the quantization limit of timers (depicted with a dashed line). Indeed, this convergence illustrates the filter capabilities of our algorithm against other sources of errors, as described previously.

However, when the temperature sampling error is higher, the impact on the prediction performance can be severe (Fig. 10b and Fig. 10c), and may become the limiting factor of the proposed method. In other words, the sensor has to be carefully chosen according to the application requirements.

More quantitatively, Fig. 11 maps the required temperature accuracy to the output performance, and can be used to select a suitable sensor for each specific application. For example, this figure shows that an additional deviation of 0.5 ppm is expected when the temperature sampling accuracy is 1° C.

VI. CONCLUSION

A novel mechanism to impose tight discipline to networked devices' internal clocks is presented. The mechanism targets a 1 ppm clock drift horizon and enables networked devices to keep synchronization between them with limited re-synchronization opportunities of one communication per day. This mechanism targets the synchronization of the emergent LPWAN technologies where the spectrum usage is jeopardized by node density, regulations, data rates and contention and hence the communication opportunities between nodes and the base station are extremely scarce. The mechanism is based on leveraging calibration information to bootstrap a learning base. The discrete temperature-drift pairs are kept and a quadratic fitting curve is derived for both calibration and learning bases. Confidence intervals are used to determine the best information base to use, i.e., either the calibrated or the learned one. Results show that as learning evolves, the learned model tends to be tailored to the particular conditions and nature of the device and hence achieving a higher precision in drift prediction.

APPENDIX IMPLEMENTATION DETAILS

In the present work, we use some well-known statistical methods, such as least-squares regression and its associated confidence intervals. For the sake of completeness, we reproduce in this appendix the most important results. See, for example, [21] for extended details.

A. Least-Squares Parameter Estimation

Let z_1, z_2, \dots, z_r be a set of r predictors related to a response variable Y . The linear multivariate regression model [21] for the j -th sample unit has the form:

$$Y_j = \beta_0 + \beta_1 z_{j1} + \beta_2 z_{j2} + \dots + \beta_r z_{jr} + \epsilon_j \quad (4)$$

where $\beta_i, i = 0, 1, \dots, r$ are fixed (unknown) regression coefficients and ϵ is a random error. We assume that $E(\epsilon_j) = 0, \text{Var}(\epsilon_j) = \sigma^2, \text{Cov}(j, k) = 0 \forall \epsilon_j \neq \epsilon_k$.

In matrix form, Eq.(4) takes the form:

$$Y = Z \cdot \beta + \epsilon$$

where Y is the $n \times 1$ vector of observations, Z is an $n \times (r+1)$ matrix, β is the $(r+1) \times 1$ vector of coefficients and ϵ is $n \times 1$.

Then, $\hat{\beta}$, the least squares estimate of β is:

$$\hat{\beta} = (Z^T Z)^{-1} Z^T Y \quad (5)$$

then, predicted values are $\hat{Y} = Z \hat{\beta} = H Y$ where $H = Z(Z^T Z)^{-1} Z^T$. With this definition, the residuals are $\hat{\epsilon} = Y - \hat{Y} = (I - H)Y$, and the residual sum of squares is obtained as $\hat{\epsilon}^T \hat{\epsilon} = Y^T (I - H)(I - H)Y = Y^T (I - H)Y$

Finally, s^2 an unbiased estimate of σ^2 is:

$$s^2 = \frac{\hat{\epsilon}^T \hat{\epsilon}}{n - r + 1} = \frac{Y^T (I - H)Y}{n - r + 1}$$

B. Inferences From the Regression Function

Given an unknown value Y_0 , the response for $z_0 = [1, z_{0,1}, \dots, z_{0,r}]$ can be easily estimated as

$$Y_0 = z_0 \hat{\beta} \quad (6)$$

i.e, in statistical terms, the response for z_0 has conditional mean $E(Y_0 | z_0) = z_0 \beta$

1) *Confidence Intervals*: A $100(1-\alpha)\%$ confidence interval for $E(Y_0 | z_0) = z_0 \beta$ (the expected response at z_0), is given by

$$z_0 \hat{\beta} \pm t_{(n-r-1)}(\alpha/2) \sqrt{z_0^t (Z^T Z)^{-1} z_0 s^2} \quad (7)$$

2) *Prediction Intervals*: A $100(1-\alpha)\%$ prediction interval for $E(Y_0 | z_0) = z_0 \beta$ (the expected response at z_0) is given by:

$$z_0 \hat{\beta} \pm t_{(n-r-1)}(\alpha/2) \sqrt{(1 + z_0^t (Z^T Z)^{-1} z_0) s^2} \quad (8)$$

C. Computational Cost

Equations (5), (7) and (8) require the evaluation of $(Z^T Z)^{-1}$ Matrix inversion is, in general, computationally expensive (for instance, the Gauss-Jordan elimination method is $O(n^3)$). However, in our particular case, crystal models are second order polynomials. This means that $Z^T Z$ is a 3×3 matrix and the inversion is computed with a few tens of arithmetic operations. Even for the more general case of cubic oscillators, the inversion of the 4×4 matrix is in the order of a hundred operations, which is perfectly suitable for low-power embedded systems.

REFERENCES

- [1] LoRa Alliance. (2015). *LoRa Specification 1.0*. [Online]. Available: <https://www.lora-alliance.org/>
- [2] Ingenu. (2015). *About Ingenu*. [Online]. Available: <http://www.ingenu.com/company/about/>
- [3] SigFox. (2014). *About SigFox*. [Online]. Available: <http://www.sigfox.com/en/about>
- [4] L. G. Roberts, "ALOHA packet system with and without slots and capture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, Apr. 1975. [Online]. Available: <http://doi.acm.org/10.1145/1024916.1024920>
- [5] ETSI. (Feb. 2015). *ETSI EN 300 328 V1.9.1*. [Online]. Available: <http://www.etsi.org/>
- [6] F. Adellantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, and J. Melia. (2016). "Understanding the limits of LoRaWAN." [Online]. Available: <https://arxiv.org/abs/1607.08011>
- [7] X. Vilajosana, P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, and L. Alonso, "Standardized low-power wireless communication technologies for distributed sensing applications," *Sensors*, vol. 14, no. 2, pp. 2663–2682, 2014. [Online]. Available: <http://www.mdpi.com/1424-8220/14/2/2663>
- [8] B. Ponsard and A. Gaudon, *Low Throughput Networks (LTN); Protocols and Interfaces*, document ETSI GS LTN 003 V1.1.1 (2014-09), 2014.
- [9] M. Frerking, *Crystal Oscillator Design and Temperature Compensation*. Dordrecht, The Netherlands: Springer, 1978.
- [10] E. Momosaki, "A brief review of progress in quartz tuning fork resonators," in *Proc. Int. Freq. Control Symp.*, May 1997, pp. 552–565.
- [11] Servei Meteorologic de Catalunya. (2015). *Automatic Weather Station Network (XEMA)*. [Online]. Available: <http://www.meteo.cat>
- [12] T. Schmid, Z. Charbiwala, R. Shea, and M. B. Srivastava, "Temperature compensated time synchronization," *IEEE Embedded Syst. Lett.*, vol. 1, no. 2, pp. 37–41, Aug. 2009.

- [13] T. Chang, T. Watteyne, K. Pister, and Q. Wang, "Adaptive synchronization in multi-hop TSCH networks," *Comput. Netw.*, vol. 76, pp. 165–176, Jan. 2015.
- [14] R. Bechmann, A. D. Ballato, and T. J. Lukaszek, "Higher-order temperature coefficients of the elastic stiffnesses and compliances of alpha-quartz," *Proc. IRE*, vol. 50, no. 8, pp. 1812–1822, Aug. 1962.
- [15] R. M. Cerda, *Understanding Quartz Crystals and Oscillators*. Norwood, MA, USA: Artech House, 2014.
- [16] Z. Yang, L. He, L. Cai, and J. Pan, "Temperature-assisted clock synchronization and self-calibration for sensor networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 3419–3429, Jun. 2014.
- [17] D. Brunelli, D. Balsamo, G. Paci, and L. Benini, "Temperature compensated time synchronisation in wireless sensor networks," *Electron. Lett.*, vol. 48, no. 16, pp. 1026–1028, Aug. 2012.
- [18] T. Schmid, R. Shea, Z. Charbiwala, J. Friedman, M. B. Srivastava, and Y. H. Cho, "On the interaction of clocks, power, and synchronization in duty-cycled embedded sensor nodes," *ACM Trans. Sensor Netw.*, vol. 7, no. 3, pp. 24:1–24:19, Oct. 2010.
- [19] W. Masood, J. F. Schmidt, G. Brandner, and C. Bettstetter, "DISTY: Dynamic stochastic time synchronization for wireless sensor networks," in *IEEE Trans. Ind. Informat.*, to be published doi: 10.1109/TII.2016.2618348 [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7592904&isnumber=4389054>
- [20] D. Stanislawski, X. Vilajosana, Q. Wang, T. Watteyne, and K. S. J. Pister, "Adaptive synchronization in IEEE802.15.4e networks," *IEEE Trans Ind. Informat.*, vol. 10, no. 1, pp. 795–802, Feb. 2014.
- [21] J. O. Rawlings, S. G. Pantula, and D. A. Dickey, *Applied Regression Analysis: A Research Tool*. New York, NY, USA: Springer, 1998.



Borja Martinez (M'15) received the B.Sc. degree in physics and electronics engineering, the M.Sc. degree in microelectronics, and the Ph.D. degree in computer science from the Universidad Autónoma de Barcelona (UAB), Spain. From 2005 to 2015, he was an Assistant Professor at the Department of Microelectronics and Electronic Systems, UAB. He is currently a Research Fellow with the Internet Interdisciplinary Institute. His research interests include low-power techniques for smart wireless devices, energy efficiency, and algorithms.



Xavier Vilajosana (SM'15) received the M.Sc. and M.Eng. degrees from UPC, Barcelona, Spain, in 2004, and the Ph.D. degree from the Universitat Oberta de Catalunya, in 2009. He is a Principal Investigator with the Wireless Networks Research Laboratory, Universitat Oberta de Catalunya. He is also Co-Founder of Worldsensing and OpenMote Technologies. He is an active member of the IETF 6TiSCH WG, where he has authored different standard proposals. He also holds 30 patents. He has been a Visiting Professor at the Prof. Pister UC Berkeley Laboratory. He co-leads Berkeley's OpenWSN Project. He was a Senior Researcher at the HP R&D labs from 2014 to 2016 and a Visiting Researcher at the France Telecom R&D Labs Paris in 2008.



Diego Dujovne (M'08) received the electronic engineering degree from Universidad Nacional de Córdoba (UNC), Argentina, in 1999, and the Ph.D. degree from UNSA, France, in 2009. Between 1999 and 2004, he developed several university-industry collaboration projects on instrumentation, control, and communications at LIADE, UNC. He is currently an Associate Professor with Universidad Diego Portales, Chile, and the President of the IEEE ComSoc (Section Chile) Chapter. His current research interests include IPv6 over LLN routing and scheduling, wireless experimental platform development, and measurement methodologies.