Enhanced Timestamping Method for Sub-Nanosecond Time Synchronization in IEEE 802.11 over WLAN Standard Conditions

Abstract—**The Time-Sensitive networks paradigm envisions the integration of Operation Technology and Information Technology in the same network. One of the requirements for building Time-Sensitive networks is sharing a global time along the network. This requirement is especially critical in wireless systems, where there are few robust methods to perform accurate time transfer. In this paper, the problem of time transfer over realistic wireless channels is studied and a time distribution scheme is proposed. The time distribution scheme has three components: Precision Time Protocol, a novel timestamping method (enhanced timestamps) and an algorithm to implement the enhanced timestamps. The performance of the proposed scheme has been evaluated in MATLAB using the IEEE 802.11n standard under several standard Wireless Local Area Network channel models. The results show that the system can reach sub-nanosecond time transfer accuracy under Non-Line-of-Sight and time-variant conditions, but its performance greatly depends on the Signal-to-Noise-Ratio and on the channel variation rate.**

I. INTRODUCTION

Operation Technology (OT) and newer Real-Time (RT) Information Technology (IT) applications are pushing the legacy networks to the Time-Sensitive network paradigm [1]. This novel paradigm envisions the support for heterogeneous applications in the same network. The next applications, among others, are expected to coexist in the same network [2]. From the OT paradigm: autonomous driving, factory automation, smart transportation, and Wireless Sensor and Actuator Networks (WSAN); whereas from the IT paradigm: RT gaming, High Frequency Trading, video calls, multimedia streaming services and web browsing. A lot of efforts are being made by researchers in different fields to obtain a solid solution that could support the vast variety of applications and their requirements in a single integrated hybrid and heterogeneous network.

One of the main requirements and challenges of Time-Sensitive networks is sharing a common base time along every element of the network. For example, novel high-performance wireless systems for Time-Sensitive
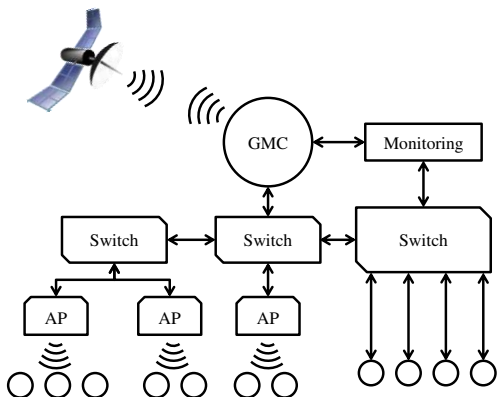

Fig. 1. Clock distribution along a hybrid network using GNSS as its reference time.

applications, such as wireless SHARP [3] or WirelessHP [4], require accurate time synchronization, as it is essential to reduce inter-frames spacing and improve the wireless system performance. Furthermore, hybrid Time-Sensitive networks also demand accurate time synchronization along the wired/wireless segments to deterministically schedule frames with minimum latency. However, the vast existence of heterogeneous technologies and implementations in hybrid networks could be a major issue to obtain high-performance time synchronization. On the other hand, most RT applications have strict time synchronization requirements. For example, time synchronization is especially critical in Cyber Physical Systems and Distributed Control Systems, where every element of the system must synchronously perform specific tasks, such as reading a sensor or changing the state of an actuator. One single failure in the time synchronization can lead to the failure of the entire system, which can cause from loss of profits to loss of human lives.

The synchronization problem is usually addressed from two main perspectives: Global Navigation Satellite System (GNSS)-based synchronization, and time distribution through the network. The first one requires Line-of-Sight (LoS) with GNSS constellation and its performance is severely deteriorated under bad environmental conditions. Furthermore, it is not cost-effective for many applications. The second perspective is not greatly affected by changes in the environment conditions and its performance can be similar to GNSS-based solutions. Finally, it is common to use a hybrid approach, where the master clock synchronizes its clock to a GNSS system. The master is usually situated outside of the facility with LoS conditions to obtain an adequate performance and it distributes its time to the rest of the nodes (Fig. 1). This approach results in a more accurate and cost-effective solution, as only one GNSS receiver is needed, and because every node will be synchronized to a common reference.

Among the existing protocols for clock synchronization, two protocols stand as the most widely used nowadays: Network Time Protocol (NTP) [5] and Precision Time Protocol (PTP) [6]. NTP protocol was designed to perform time transfer in IT services and its accuracy is in the millisecond range. Therefore, it cannot provide enough accuracy for OT applications. PTP accuracy is superior to the NTP protocol and hence the natural option to obtain very high clock accuracy in Time-Sensitive networks. Nonetheless, PTP timestamping precision is proportional to the timestamping clock period, and then it cannot provide sub-nanosecond time synchronization in most systems. A third protocol, White Rabbit [7], stands out as a solution to eliminate the PTP limitations. White Rabbit is based on three technologies: Synchronous Ethernet (SyncE) [8], PTP, and the Dual Mixer Time Difference [7]. White Rabbit protocol implemented on Ethernet links can effectively provide sub-nanosecond time synchronization [7].

Wireless systems exhibit several limitations that challenge successful high-performance time synchronization. We may highlight, among others, low bandwidth, multipath propagation, channel variation over time, and the lack of (Line-

of-Sight) LoS. The bandwidth imposes a timestamp granularity in the order of the inverse of the bandwidth, which clearly limits the synchronization accuracy. This is the main reason why Ultra-Wide Band (UWB) radios are usually found in Radio-Frequency (RF) indoor localization systems [9]. The multipath propagation produces signal time-dispersion that also affects time synchronization performance. In addition, the changes of the environment over time yield to a time-variant Channel Impulse Response (CIR) and, thus, a variant signal time dispersion. Time-variant channels are present in networks with mobility which might be produced by moving communication nodes and/or moving environment (people, machinery, etc.). Finally, the lack of LoS produces strong variations of the wireless channel delay, which in turn introduces a strong error in the time synchronization. The combined effect of these phenomena greatly increases the error on the timestamps and, thus, are challenging hurdles for high-performance time synchronization over real-world wireless channels.

In this work, PTP synchronization mechanism is described in detail, and it is shown that timestamps based on conventional frame start detectors (or conventional timestamps) have two main issues in wireless systems. In the first place, they cannot guarantee a time transfer accuracy better than the sampling period. In the second place, the multipath propagation and the wireless channel time-variant character reduces the precision of the frames Time-of-Arrival (ToA), which introduces an error component in the time synchronization. This is especially critical under Non-Line of Sight (NLoS) conditions, where a prevalent propagation path does not exist, thus the ToA may have strong variations.

Based on the above-mentioned analysis, we proposed an improved wireless timestamping method (enhanced timestamping) which precisely estimates the frames ToA in severe multipath conditions and overcomes the inherent limitations of conventional timestamps. The enhanced timestamping uses the whole CIR to precisely estimate the ToA and can be seamlessly used in both LoS and NLoS conditions, because it does not rely on the detection of the main channel component and because it is very robust to multipath propagation. The two main limitations of the enhanced timestamps precision are the channel variation rate and the Signal to Noise Ratio (SNR). The performance of the presented technique has been obtained numerically through Matlab simulations using the IEEE 802.11n physical layer and four wireless time-dispersive and time-variant channel models. These simulations show that the use of the enhanced timestamping compared to conventional timestamping yields a meaningful increase of synchronization accuracy that reaches sub-nanosecond performance. Finally, we include some guidelines to effectively implement the enhanced timestamping into already existing wireless systems or in proprietary solutions. The proposed time synchronization scheme has been designed to improve time synchronization in wireless Time-Sensitive networks and its possible applications (Fig. 2: industry 4.0, autonomous driving, smart transportation, etc.). In addition, there are some applications that need to deploy a communication system just for performing time distribution. These networks are usually found in scientific experiments [10], where a system must gather data with very high time accuracy over large areas. In this case, the use of wireless links for time transfer, instead of wired links, could lead to more cost-effective solutions, faster deployments, and lower failure possibility, due to the inexistence of cables. Finally, this solution may be useful in other fields where time
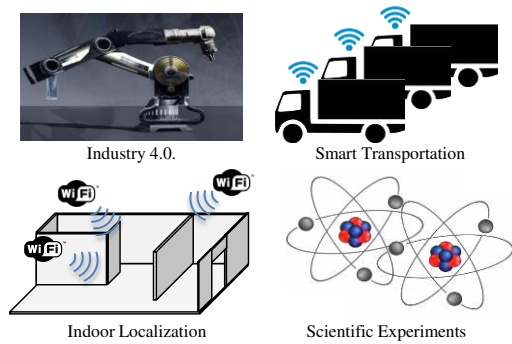

Fig. 2. Applications benefitting from accurate time synchronization.

synchronization has significant impact in the application performance, such as indoor localization [11] (Fig. 2).

In this work we have used the accuracy, trueness and precision terms as defined by the ISO 5725-1:1994 [12]. Trueness is defined as the difference between the mean value of a series of results and the true value, precision is defined as the std deviation of a series of results, and accuracy is defined as the combination of both the trueness and accuracy. Then, an accurate system is a system that has both good trueness, and good precision.

The rest of the paper is organized as follows. First, the state-of-the-art of time transfer is presented in section II. PTP protocol over wireless networks is described in section III. In section IV, the limitations of conventional timestamps are analyzed, and the enhanced timestamping method is proposed. The algorithm to implement the enhanced timestamps is described in Section V. Section VI details some guidelines to successfully integrate the synchronization scheme into a wireless system. Section VII presents the numerical results achieved with the proposed solution. Finally, section VIII summarizes some conclusions of the work.

II. BACKGROUND AND RELATED WORK

As previously stated, a common approach to time distribution is the use of a GNSS satellite constellation. GNSS-based solutions can obtain time synchronization accuracy in the ten nanosecond range [13]. In [14], some considerations to reduce the synchronization error to less than 1 ns are presented. These considerations assume that the nodes are situated in an open space with LOS to the GNSS system. In addition, bad environmental and climate conditions can significantly deteriorate the link quality and, hence, the synchronization performance. To the best of our knowledge, the performance of the considerations stated in [12] has not been tested yet in a real testbed.

One application that could use this approach is autonomous driving, as future vehicles will include a GNSS receiver. However, the existence of tunnels, underground roads and parking, and the fact that GNSS signals can be interfered or spoofed require the use of at least a non-GNSS dependent approach as a backup system or as the main synchronization system.

The second approach is time distribution within a communication system using PTP or a similar protocol. The performance of this approach heavily relies on the communication system capacities, its implementation and the deployment environment. Low accuracy, yet simpler, PTP software implementations can achieve a clock synchronization in the range of several hundreds of nanoseconds [15]. This solution is very far from the desired performance. PTP with Software-based timestamps is significantly overcome by PTP

with hardware-based timestamps, where the timestamps of PTP frames are taken at the physical layer [16]. However, to the best of our knowledge, there is not yet any wireless standard that natively supports PTP with hardware-based timestamps. This clearly contrasts with the Ethernet standard, which already covers the use of hardware timestamping and it is implemented in a high variety of chips.

There are several wireless custom systems in the literature for different wireless technologies that are based on PTP or similar protocols using hardware timestamps. In [16], several solutions for time synchronization over 802.11 are compared.

In [17], an 802.11g modem implementation over Field-programmable Gate Array (FPGA) with conventional hardware timestamps is described. The timestamps are taken using the frame start detection block (a preamble cross-correlator), and its timestamp granularity is 50 ns. The time synchronization performance was measured over the IEEE 802.11 Wireless Local Area Network (WLAN) standard channel models [18] using a channel emulator. The measurements show that the clock synchronization accuracy can be as low as 30 ns in low and medium-dispersive channels with time-variant conditions. However, this solution is far from the sub-nanosecond time synchronization performance, as its timestamps precision is fixed to the inverse of the bandwidth of the system.

Advanced timestamping techniques have been proposed in the literature in order to obtain sub-nanosecond time synchronization over IEEE 802.11 [19], [20] and [21]. In [19], it is described an 802.11b implementation in FPGA with hardware timestamps and subsample timestamping precision. This design is based on conventional hardware timestamps combined with a synchronizer used to calculate the phase difference between the transmitter and receiver clocks. This design resulted in a synchronization accuracy better than 600 picoseconds for static conditions. However, according to the experiments documented in [19], the performance of the system is highly deteriorated for time-variant channels, showing a synchronization error bigger than 20 ns. The receiver presented in [19] is improved in [20] and in [21] to combat the multipath propagation. The improvements proposed in [20] are mainly based on the use of frequency hopping in order to obtain timestamps considering several independent channels. This is used to average the timestamping error introduced by the multipath propagation. However, such a system needs a very specific implementation and the authors state in the paper that the ranging error is significantly larger when the target is moving. The solution proposed in [21] uses the interpolation of the cross-correlation peak to take timestamps with subsample precision and an equalizer to combat the multipath components and reduce the ranging error. However, the system is still designed for LoS conditions, and it is vulnerable to strong multipath components. In summary, these solutions [19], [20] [21] require a strong LoS because they rely on the detection on the first channel replica, which is not available under NLoS conditions.

Another wireless synchronization scheme based on the interpolation of the cross-correlation peak but over IEEE 802.15.4 Chirp Spread Spectrum (CSS) is shown in [22], [23] and [24]. In [21] it is developed a theoretical analysis of the feasibility of the CSS modulation to perform precise timestamping and an early experimental validation is presented. The system performance is reported through simulations under direct LoS and limited multipath propagation conditions. Furthermore, the authors note that the system is not designed for NLoS conditions and the lack of LoS would strongly affect the timestamping precision and the overall system performance. A real testbed is developed, and its performance is evaluated in [22]. It is shown that the system can obtain sub-nanosecond synchronization accuracy over LoS conditions, but its performance for NLoS conditions is not reported. Finally, the timestamping validation strategy presented in [24] does neither include NLoS conditions. In conclusion, this solution is very suitable for LoS and static conditions, but it is not suitable for NLoS and time-variant conditions.

There are also some attempts to port white rabbit to the wireless domain. The wireless white rabbit approaches are mainly based on transmitting the clock phase and frequency from the master to the slave, and synchronize the slave clock by using an Analog Phase Locked Loop (PLL). For example, the design presented in [25] estimates the clock phase and frequency to synchronize the slave of a 5G wireless backhaul link, obtaining (by simulations) an accuracy of about 40 ps in LoS conditions. However, the system performance is highly degraded when a simple multipath propagation channel is considered, and the synchronization accuracy drops immediately to more than 1 ns. On the other hand, the carrier phase can also be used to synchronize the slave clock phase and frequency. This approach is effectively implemented in [26], where it is detailed a partial implementation of a custom IEEE 802.11g modem with carrier phase estimation which obtains an exceptional accuracy of 50 ps. Again, this design requires LoS and no multipath.

Finally, a robust timing synchronization technique over a custom Orthogonal Frequency Division Multiplexing (OFDM) physical layer along with an enhancement to PTP are proposed in [27]. The robust timing synchronization technique is designed to reduce the jitter in the frame start detection, whereas the enhanced PTP is designed to reduce the error in the clock offset estimation. The performance of the timing synchronization technique was evaluated over several channel models with different time-dispersion characteristics. However, the presented results do not include the time transfer performance, but only the frame start detection precision. Although both parameters are related, the time transfer accuracy cannot be directly derived from the frame start detection precision, as the channel variation during the PTP frame exchange must be considered.

III. PRECISION TIME PROTOCOL

PTP is a well-known and broadly used protocol described in IEEE 1588 standard [6]. The protocol follows a master-slave structure, where the master shares its local time with the slaves connected to it. To perform the time distribution, PTP uses the timestamps taken in a four frame exchange (Fig. 3). Firstly, a Sync frame is transmitted by the master, which takes the t_1 timestamp. The frame arrives to the slave that estimates the ToA and takes the timestamp t_2'. Then, a Follow_up frame is sent from the master to the slave to transmit the timestamp t_1 to the slave. The follow up frame is optional, as the t_1 timestamp can be also delivered through the Sync frame if the PTP implementation supports that option. In this paper, it is considered that the follow up is not necessary. Afterwards, the Delay Request frame (Delay_Req) is transmitted to obtain two more timestamps (t_3', t_4). Finally, the Delay Response frame (Delay_Resp) delivers the t_4 timestamp to the receiver. Once the four timestamps are in the slave side, it performs the calculations stated in (1) and (2) to synchronize its time with the master time
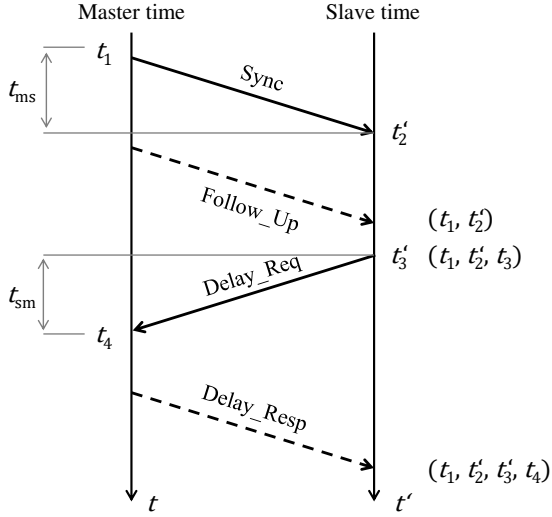
Fig. 3. PTP frame exchange, being t the time of the master clock and t' the time of the slave clock.

$$\tilde{t}_{ms} = \frac{t'_2 - t_1 + t_4 - t'_3}{2}, \quad (1)$$

$$\tilde{t}_o = t'_2 - t_1 - \tilde{t}_{ms}, \quad (2)$$

where $\tilde{t}_{ms}$ represents the estimated path delay and $\tilde{t}_o$ is the estimated clock offset between the master and slave clocks. The time correction is performed by subtracting $\tilde{t}_o$ from the slave time.

Whereas the IEEE 1588 standard clearly describes the mechanisms to obtain accurate time synchronization, differences among implementations can cause vast performance differences. The main error contributions in wireless PTP are: the time-dispersive and time-variant character of the wireless channels, the timestamping error, and the calibration of the nodes.

At a given instant, we can consider that the wireless CIR from master to slave equals the CIR from slave to master (symmetric channel) and, thus, $t_{ms} = t_{sm}$. However, a wireless CIR is varying along the time due to environment changes. Thus, to satisfy the assumption of symmetry it is necessary that the elapsed time between the PTP sync and PTP Delay Request is much smaller than the coherence time of the channel. Regarding timestamping errors, they may be induced by a plethora of sources: software jitter, precision in the estimation of the Time-of-Departure (ToD) and ToA, jitter caused by analog components, the communication system characteristics (narrowband, wideband, preamble length, etc.), and the quality of the communication, i.e., the SNR.

Finally, the timestamping calibration of the nodes is not related to the protocol, thus it is not analyzed in this paper.

IV. HIGH-PERFORMANCE TIME SYNCHRONIZATION THROUGH ENHANCED TIMESTAMPS

The key to obtain High-Performance time synchronization is tightly related to the timestamps quality. Timestamps quality is commonly defined as the difference between the exact ToA/ToD and the estimated ToA/ToD. The ToD (t_D) may be accurately estimated ($\tilde{t}_D$) whenever the communication chain jitter and the calibration error are negligible. In this work, it is considered that these conditions are satisfied. In addition, the estimated ToA can be defined as

$$\tilde{t}_A = \tau_h + t_D + t_e, \quad (3)$$

being τ_h the channel delay, t_e the timestamping error and $\tilde{t}_A$ the estimated ToA. A wireless channel is usually composed of several signal replicas received at different time instants (i.e. time-dispersive channel), thus a unique τ_h definition does not exist. Due to this, τ_h definition will depend on which algorithm is used to perform the ToA estimation. The most commonly definition is considering that τ_h is equal to the delay of first channel component. This definition is reasonable for LoS channels, but it will present strong variations in NLoS channels. In addition, the timestamping error t_e will greatly depend on the frame start detection algorithm and on the communication system properties.

In the following subsections, a comprehensive analysis of the limitations of conventional timestamps is shown, and the enhanced timestamps are developed.

A. Timestamp model

Let be $s[l]$ a pseudorandom white sequence with length and energy L and known by the receiver and transmitter. The sequence has the next property

$$R_{ss}[n] \approx L\,\delta[n], \quad (4)$$

where R_{ss} is the autocorrelation of $s[l]$ and $\delta[n]$ is the Kronecker delta. The sequence $s[l]$ is sent using a pulse-shaping filter with impulse response $g(t)$, thus, the transmitted signal may be written as follows

$$s_1(t) = \sum_{l=0}^{L-1} s[l]\,g(t - lT + \phi_{Tx}T), \quad (5)$$

where T is the symbol rate and $\phi_{Tx}T$ is the uncertainty in the sampling instant (i.e. the jitter of the transmitter) that is modeled as a zero-mean Gaussian distribution $\phi_{Tx} \sim \mathcal{N}(0, \sigma^2_{\phi_{Tx}})$. Without loss of generality, we will assume that $g(t)$ is a real, band-limited signal, with bandwidth $B = \frac{1}{2T}$ and unit energy. $s_1(t)$ is the complex baseband representation of the passband signal with carrier frequency f_c. Although $g(t)$ is not time-limited, it can be assumed that most of the energy $g(t)$ is confined in the interval $[0, T_g]$, thus it is approximated by a truncated version of $g(t)$. Therefore, $g(t)$ is considered a finite impulse response. The signal $s_1(t)$ is transmitted through a communication channel whose impulse response is noted as $h(t)$. The CIR is defined in the interval $[T_d, T_h + T_d]$, where T_d is the CIR start and T_h is the CIR duration. $h(t)$ is the complex baseband representation of the passband CIR with a carrier frequency of f_c. Hence, the signal at the input of the matched filter is

$$s_2(t) = \sum_{l=0}^{L-1} s[l]\,g_h(t - lT + \phi_{Tx}T) + n_2(t), \quad (6)$$

where $g_h(t)$ is the convolution of $g(t)$ and $h(t)$, i.e., $g_h(t) = (g * h)(t)$. The noise component $n_2(t)$ is modeled as Additive White Gaussian Noise (AWGN). The matched filter with a response equal to $g(-t + T_g)$ is applied to $s_2(t)$ and it results

$$r(t) = \sum_{l=0}^{L-1} s[l]\,p(t - lT + \phi_{Tx}T) + n(t), \quad (7)$$

being $p(t)$ the convolution of $g_h(t)$ and $g(-t + T_g)$, and $n(t)$ the filtered noise resulted from the convolution of $n_2(t)$ and $g(-t + T_g)$. Finally, $r(t)$ is sampled at the receiver with a sampling period T
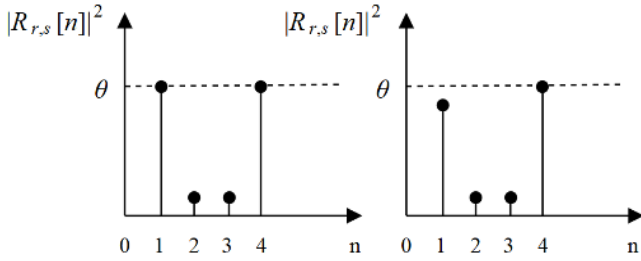
Fig. 4. Issue of threshold-based timestamps: small CIR variation can cause a big frame start detection error.

$$r[k] = r(t)|_{t=kT+\phi_{Rx}T}$$

$$= \sum_{l=0}^{L-1} s[l]p((k-l)T + \phi T) + n(kT + \phi_{Rx}T), \quad (8)$$

being ϕ_{Rx} the timing jitter resulted from the sampling of $r(t)$ and $\phi = \phi_{Rx} + \phi_{Tx}$, the whole jitter. ϕ_{Rx} is modeled as $\mathcal{N}(\mu_{\phi_{Rx}}, \sigma_{\phi_{Rx}}^2)$, being $\mu_{\phi_{Rx}}$ the unkown phase difference between the master and slave clocks. $\mu_{\phi_{Rx}}$ is considered time-invariant because the clock drift variation is negligible taking into account the time length of $r(t)$. <u>The quantization noise due to the signal sampling may be another source of error that could limit the synchronization accuracy. The quantization noise would basically create a noise floor, which limits the SNR and hence the synchronization accuracy. We have not included quantization noise in this work, but it should be considered in receivers with low resolution Analog to Digital Converters (ADC).</u>

B. Conventional timestamps

As stated before, PTP performance is mainly limited by the precision in the ToA estimation. A common approach to precisely estimate the ToA is to use the frame start detector included in the physical layer of the communication system. A widely used frame detector is based on detecting a known sequence in the received samples. For example, this detector is used in [17] to take PTP timestamps and perform time synchronization. In order to find the training sequence, $r[k]$ is cross-correlated with $s[k]$.

$$R_{r,s}[n] = (r \star s)[n]. \quad (9)$$

$R_{r,s}[n]$ will have a peak when the training sequence is cross-correlated with the received training sequence. If the peak exceeds a threshold, it will be considered that a frame has been detected. Hence, the estimated frame start (n_{ini}) is

$$n_{ini} = \min\{n \in \mathbb{N}/|R_{r,s}[n]|^2 > \theta\}. \quad (10)$$

The timestamping algorithms based on threshold detection rely on detecting the first component of $R_{r,s}[n]$, or the first signal replica. Therefore, the channel delay for this approach could be defined as $\tau_h = T_d$. The value of θ is usually set based on minimizing the number of undetected frames without exceeding a false alarm probability. This simple solution has two main drawbacks:

- The algorithm is not robust, as small differences in $R_{r,s}[n]$ can produce big differences in the frame start detection.
- The timestamps are quantized to the sampling period.

An example of the first drawback is depicted in Fig. 4, where a similar $R_{r,s}[n]$ results in a totally different ToA estimation. This situation can be very common in wireless channels without a strong path, such as in NLoS conditions. On

the other hand, the timestamps resolution cannot be higher than the sampling period. A detailed analysis of the quantization drawback of conventional timestamps is shown in the next subsection.

C. PTP with conventional timestamps

In order to illustrate the quantization drawback of conventional timestamps, an example of the PTP performance using conventional timestamps is shown in a simplified setup. It has been assumed that:

- $h(t) = \delta(t - \tau_h)$, being $\delta(t)$ a Dirac delta.
- The clock offset is $t_o = -(i_0 + \phi)T$.
- The master and slave clocks have the same clock drift, and the jitter is almost constant. Hence, $\phi = \mu_{Rx}$.
- The noise is negligible.

Being $i_o \in \mathbb{Z}$ and $\phi \in [0,1)$. i_o represents the number of clock cycles between the start of both clocks, and ϕ the clocks relative phase. The master clock signal $C_M(t)$ and the slave clock signal $C_S(t)$ are represented in Fig. 5.

The local time of the master and the slave can be expressed as

$$t_M(t) = t,$$
$$t_S(t) = t + t_o = t - (i_0 + \phi)T. \quad (11)$$

The master and slave represent their local time as a discrete distribution by sampling their local time at each clock rising edge.

$$t_M(t)|_{iT} = iT \rightarrow t_M[i] = iT.$$
$$t_S(t)|_{t=iT+\phi T} = (i - i_0)T \rightarrow t_S[i] = (i - i_0)T, \quad (12)$$

with i an integer value, i.e., $i \in \mathbb{Z}$. The PTP frame exchange starts at $t_M[i_1]$, where the master sends a PTP sync frame and takes the t_1 timestamp.

$$t_1 = t_M[i_1]. \quad (13)$$

The frame is received by the slave, which detects the frame at $t_S[i_2]$ and takes the timestamp t_2'. The relation between both timestamps can be expressed as

$$t_2' = t_S[i_2] = t_M[i_1] - i_oT + \left\lceil \frac{-\phi T + \tau_h}{T} \right\rceil T = t_M[i_1] - i_oT + \hat{t}_{ms}, \quad (14)$$

being $\hat{t}_{ms}$ the apparent channel delay from master to slave and $\lceil \cdot \rceil$ the ceil operator. Afterwards, the slave transmits a Delay Request frame at $t_S[i_3]$:

$$t_3' = t_S[i_3]. \quad (15)$$

The Delay Request arrives to the master, which detects the frame at $t_M[i_4]$

$$t_4 = t_M[i_4] = t_S[i_3] + i_oT + \left\lceil \frac{(\phi-1)T + \tau_h}{T} \right\rceil T = t_M[i_1] + i_oT + \hat{t}_{sm}, \quad (16)$$
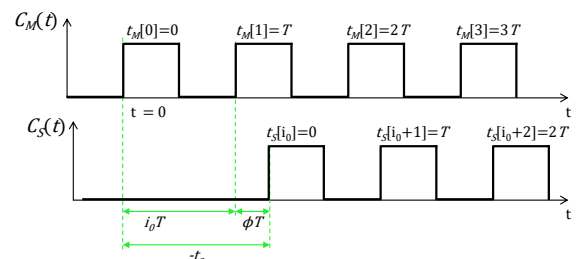

Fig. 5. Representation of master and slave clocks with a time offset between them of $t_o = -(i_0 + \phi)T$.

being $\hat{t}_{sm}$ the apparent channel delay from master to slave. Finally, the PTP Delay Response frame delivers the t_4 timestamp to the slave.

Once the slave has the four timestamps, it calculates the channel delay and clock offset between its local clock and the master clock.

$$\tilde{t}_{ms} = \frac{t'_2 - t_1 + t_4 - t'_3}{2} = \frac{\hat{t}_{ms} + \hat{t}_{sm}}{2}. \tag{17}$$

$$\tilde{t}_o = t'_2 - t_1 - \tilde{t}_{ms} = -i_0 T + \frac{\hat{t}_{ms} - \hat{t}_{sm}}{2} = t_0 + \phi T + \frac{\hat{t}_{ms} - \hat{t}_{sm}}{2}. \tag{18}$$

where it has been considered that $t_o = -(i_0 + \phi)T$. From equation (18), we might identify the synchronization error, ϵ_t, as the difference between the real clock offset and the estimated clock offset, i.e.,

$$\epsilon_t = \phi T + \frac{\hat{t}_{ms} - \hat{t}_{sm}}{2} = \phi T + \left(\left\lceil \frac{\tau_h - \phi T}{T} \right\rceil - \left\lceil \frac{\tau_h + (\phi - 1)T}{T} \right\rceil \right) \frac{T}{2}. \tag{19}$$

Using equation (19), it is clear that $\epsilon_t = \frac{T}{2}$, if $\phi = 0$ or 1; and that $\phi T \le \epsilon_t \le \left(\phi + \frac{1}{2} \right) T$, if $0 < \phi < 1$, for these intermediate values of ϕ, the actual value of ϵ_t depends also on τ_h. From this analysis it is clear that PTP with conventional timestamps cannot provide a perfect synchronization, $\epsilon_t = 0$, and that, for most of cases, the jitter will be distributed in the interval $\left[\phi T, \left(\phi + \frac{1}{2} \right) T \right]$.

D. Enhanced timestamps

In this subsection the enhanced timestamping method is stated. The enhanced timestamps have been designed with two main purposes. The first goal is to overcome the sampling clock period bound, T, that limits the precision of the conventional timestamping method. The second goal is the robustness against small CIR variations that produce strong ToA fluctuations in conventional timestamps. For the sake of clarity, (8) is repeated here

$$r[k] = r(t)|_{t=kT+\phi_{Rx}} = \sum_{l=0}^{L-1} s[l]\, p(kT - lT + \phi T) + n(kT + \phi_{Rx}T). \tag{20}$$

As can be seen, (20) leads to a similar expression of the classical Symbol Timing Recovery (STR) problem [28]. From here two cases can be distinguished

$$\begin{aligned} B_c > B, \\ B_c \le B, \end{aligned} \tag{21}$$

where B_c is the coherence bandwidth of the channel and B is the system bandwidth. When the channel coherence bandwidth is higher than the system bandwidth, the CIR can be approximated by a Dirac delta with weight h_0 [29]. The magnitude of h_0 is equal to the square root of the channel power gain and its phase is equal to the phase of the CIR. Thus, the received signal can be expressed as

$$r[k] = h_0 \sum_{l=0}^{L-1} s[l]\, g_1(kT - lT + \phi T) + n(kT + \phi_{Rx}T), \tag{22}$$

being g_1 the convolution of $g(t)$ and $g(-t + T_g)$. Therefore, the ToA can be precisely estimated by using any STR algorithm, which are already included in most communication systems. Very similar solutions are described in other works [18] [21], where the receiver estimates the phase of the master clock and takes timestamps with sub-sample precision.

Nonetheless, this solution can only be effectively used over channels with a strong direct component and over static channels, because STR algorithms performance is deteriorated under time-dispersive and time-variant channels. Furthermore, STR algorithms rely on the detection of only one component, which would lead to a very unstable ToA estimation in NLoS, as shown in Fig. 4. Hence, the aim is to design an algorithm to precisely estimate the ToA in time-dispersive channels (i.e. $B_c \le B$).

The ToA in a wireless system operating under a time-dispersive channel should not be defined as a unique instant, as the signal is replicated and received in multiple instants. Therefore, the channel delay τ_h has been defined as the mean delay spread of $p(t)$ [30]

$$\tau_h = \bar{\tau} = \frac{\int_{-\infty}^{+\infty} |p(t)|^2\, t\, dt}{\int_{-\infty}^{+\infty} |p(t)|^2\, dt}. \tag{23}$$

It must be noted that, under time-variant channel propagation, the CIR can change over time, so the channel delay will also change. Nonetheless, from definition (23), it is clear that two similar $p(t)$ will produce similar τ_h. Furthermore, the mean delay spread property

$$\bar{\tau} = \frac{\int_{-\infty}^{\infty} |p(t - t_0)|^2\, t\, dt}{\int_{-\infty}^{\infty} |p(t - t_0)|^2\, dt} = \frac{\int_{-\infty}^{\infty} |p(t)|^2\, t\, dt}{\int_{-\infty}^{\infty} |p(t)|^2\, dt} + t_0, \tag{24}$$

shows that it is linearly affected by the reference instant. This property offers a great advantage in the implementation of the enhanced timestamps, as they can be implemented as an improvement to the conventional timestamps. These details are developed in Section V, where it is proposed an algorithm to implement a receiver with enhanced timestamps.

Furthermore, it can be proven by using Lemma 1 that $\bar{\tau}$ can be calculated from the discrete mean delay operator applied to the discrete version of $p(t)$ sampled at the Nyquist Rate

$$\bar{\tau} = \frac{\int_{-\infty}^{+\infty} |p(t)|^2 t\, dt}{\int_{-\infty}^{+\infty} |p(t)|^2\, dt} = T \frac{\sum_{n=-\infty}^{+\infty} |p[n]|^2 n}{\sum_{n=-\infty}^{+\infty} |p[n]|^2}. \tag{25}$$

Lemma 1. Let $p(t)$ be a causal signal of duration T_g, with unit energy, and approximately band-limited to bandwidth B; and let $T = 1/2B$ be the sampling period. Then, it follows the next identity:

$$\int_{-\infty}^{\infty} |p(t)|^2 \cdot t\, dt = T^2 \cdot \sum_{n=-\infty}^{+\infty} |p[n]|^2 n, \tag{26}$$

being

$$p[n] = p(t)\ |_{t=nT} = p(nT). \tag{27}$$

Hence, the discrete mean delay spread also has the shift property (24), which means that it is unaffected by delay shifts in $p[n]$, such as the unknown sampling phase. Therefore, the operator does not have any quantization and can be used to obtain the ToA without error.

Nevertheless, in most communications systems the CIR is not known, thus it has to be estimated at the receiver. A suitable option is estimating the CIR based on the received sequence $r[k]$. The most widely used CIR estimators are the minimum mean-square error (MMSE) estimator and the Least Square (LS) estimator [31]. However, a more appropriate estimator in this case could be the cross-correlation of $r[k]$ and $s[k]$, because the estimation is directly obtained in the time domain.

$$\tilde{p}[n] \propto R_{r,s}[n] = (r \star s)[n]. \tag{28}$$

$R_{r,s}[n]$ is proportional to $\tilde{p}[n]$ in the absence of noise considering that the autocorrelation of $s[k]$ is approximately equal to a Kronecker delta with amplitude L. Furthermore, $\tilde{p}[n]$ and $R_{r,s}[n]$ will be equivalent, because the mean delay spread normalizes the signal energy. On the other hand, $R_{r,s}[n]$ is not limited in energy, thus the sum of the discrete mean delay spread operator must be limited. To establish the sum limits, it is assumed that the autocorrelation of $s[k]$ is approximately equal to a Kronecker delta. Then, the sum start is

$$n_s = \left\lceil \frac{T_d}{T} - \phi \right\rceil, \tag{29}$$

and its length is

$$N = \left\lceil \frac{T_h + 2T_g}{T} \right\rceil. \tag{30}$$

Thus, the enhanced timestamps operator reduces to

$$\bar{\tau} = T_s \frac{\sum_{n=n_s}^{n_s+N-1} |R_{r,s}[n]|^2 n}{\sum_{n=n_s}^{n_s+N-1} |R_{r,s}[n]|^2}. \tag{31}$$

Eq. (31) results in a very simple and robust expression to estimate the ToA of the received frames, because: it is not vulnerable to start errors, as it integrates the whole CIR and it does not have a resolution bound. Therefore, the problem inherent of conventional timestamps is overcome by the definition of the enhanced timestamps. Nonetheless, n_s and N are not known in advance, and they must be estimated to ensure that $\tilde{\tau}$ is obtained with minimum jitter. To do so, a simple and robust algorithm based on a dual $\bar{\tau}$ calculator is proposed. The algorithm is described in the next section.

V. ENHANCED TIMESTAMPS IMPLEMENTATION

The enhanced timestamping method relies on the knowledge of n_s and N to establish the correlation window limits, but this information is not known in advance. To obtain an adequate performance, the estimation of n_s and N must be very robust, because errors in the integration window position will deteriorate the timestamps precision.

First, the correlation window length, N, is pre-configured according to the communication systems properties. For example, in an OFDM-based system T_h can be set to the duration of the cyclic prefix, due to it is the maximum CIR duration to avoid inter symbol interferences. On the other hand, T_g should be chosen according to the implemented pulse-shaping filter. Therefore, the problem is reduced to find n_s.

To estimate n_s, an iterative algorithm based on two timestamping methods is proposed. The algorithm is stated in *Algorithm 1*. The received signal is first introduced to the cross-correlator (9) and the frame start index is detected using a threshold (10). n_{ini} is set as the mid position of the CIR, so $n_s = n_{ini} - N/2$. The captured samples are sent to the discrete delay spread operator. The result of the operation will be the enhanced timestamp ($\tilde{\tau}$). However, the threshold detector is prone to errors, thus n_{ini} estimation may not be exact. Therefore, n_{ini} is recalculated using $\tilde{\tau}$ to improve its precision. This process is iteratively done until a fixed number of iterations K_{it}.

This algorithm performs a rounding in step 3.3, which adds an error to the correlation window position. To eliminate this error, a fractional delay filter based on sinc interpolation [32] can be used instead of the rounding operation. This operation allows a perfect alignment of the correlation window and eliminates the error bound of *Algorithm 1*. However, the computational complexity of this operation is considerably

---

Algorithm 1. Enhanced timestamps implementation

Input: $r[k], s[k], K_{it}$
Output: $\bar{\tau}$
1. Compute $R_{r,s}[n] = (r \star s)[n]$;
2. Find $n_{ini} = \min\{n \in \mathbb{N} / |R_{r,s}[n]|^2 > \theta\}$;
3. **repeat**
  3.1. $n_s = n_{ini} - N/2$;
  3.2. $\bar{\tau} = T \frac{\sum_{n=n_s}^{n_s+N-1} |R_{r,s}[n]|^2 \cdot n}{\sum_{n=ns}^{n_s+N-1} |R_{r,s}[n]|^2}$;
  3.3. $n_{ini} = round\left(\frac{\tilde{\tau}}{T}\right)$;
  3.4. $k_{it} = k_{it} + 1$;
  3.5. **until** $k_{it} == K_{it}$;

---

higher than the complexity of the rounding operation and its implementation should only be considered for specific cases, when the error caused by the rounding operation limits the synchronization performance.

Regarding the algorithm implementation in real devices, some specific parts of the algorithm must be implemented in hardware and other parts can be either implemented in hardware or in software. The steps 1 and 2 of *Algorithm 1* must be implemented in hardware because of the high computation complexity of the cross-correlation, and because n_{ini} must be obtained with high precision using a hardware clock. Once a frame has been detected and n_{ini} has been obtained, the result of the cross-correlation (i.e. the CIR) along with n_{ini} can be transmitted to a software layer to compute $\bar{\tau}$. Therefore, the effort to add the enhanced timestamps to a wireless system is low, as the operations of step 1 and 2 are usually included in wireless receivers, such as in the receiver implemented in [17], and the operations of step 3 can be done at software.

VI. INTEGRATION OF THE SYNCHRONIZATION SCHEME INTO A WIRELESS SYSTEM

The presented synchronization scheme combines PTP with our novel timestamping method and it is able to provide high synchronization accuracy over a large variety of wireless conditions. To achieve this purpose, it has been assumed that the uplink CIR equals the downlink CIR during the PTP sync and PTP delay request frame exchanges and that the CIR stays constant during that period of time (T_{S-D_r}). The uplink-downlink channel symmetry requires the use of Time Division Duplex (TDD). The invariance of the CIR during T_{S-D_r} time entails a fast frame exchange whose requirements we have found through simulations (see results section). For instance, for low mobile conditions (up to 3 km/h), $T_{S-Dr} = 1$ ms is enough to keep the time synchronization performance, meanwhile for higher speeds (more than 80 km/h) an elapsed time $T_{S-Dr} = 0.1$ ms might be necessary.

Low T_{S-Dr} is usually difficult to fulfill in wireless technologies depending on their throughput and their medium access procedure. For example, $T_{S-Dr} = 1$ ms is unfeasible in legacy IEEE 802.11 because IEEE 802.11 medium access is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [33], which is not deterministic. Hence, the PTP delay request may be delayed several milliseconds after the PTP sync frame. Furthermore, broadcast PTP sync frames could be received at the same time by several slaves. Then, the slaves would try to answer with a PTP delay request almost at the same time, which would probably cause frame collisions, and would greatly reduce wireless system overall throughput. Thus, the implementation of the

synchronization scheme over legacy 802.11 arises two challenges: avoiding frame collisions and achieving low latency.

To avoid frame collisions, a wireless system can use a unicast PTP implementation. Although common PTP implementations are multicast/broadcast, the use of unicast PTP over wireless is very convenient, as it solves some issues found in wireless, such as the collisions problem and, in fact, PTP standard supports unicast implementations [6]. Then, the master clock will keep each slave synchronized by pooling them with Sync frames, which will be answered with PTP delay request frames.

To achieve a low T_{S-Dr}, some already existing mechanisms of IEEE 802.11 may be used. For example, the master clock may reserve some airtime by using a predefined NAV counter value at the transmission of the sync frame. The NAV counter [33] is used to defer the access of other wireless nodes, and would allow the slave to freely answer the PTP delay request as fast as possible. The delay may be in the range of 0.5 ms to 1 ms using this procedure. Although these changes are not standard compliant and need specific SW implementation, the modifications can be easily carried out and the nodes would be still compatible with the 802.11 standard. Another option to obtain low T_{S-Dr} in IEEE 802.11 would be using the IEEE 802.11 ACK frame as the PTP delay request frame to take the timestamps t_3 and t_4. This would not strictly be PTP compliant, but it is a simple modification, and $T_{S-Dr} < 150\ \mu s$ can be obtained. In this case, the airtime duration of the PTP sync frame limits the value of T_{S-Dr}.

Other medium access schemes, such as Time Division Multiple Access (TDMA), which is used in 5G and LTE, would allow an easier implementation of the synchronization scheme. In TDMA, the AP can pre-allocate radio resources to allow the deterministic transmission of the PTP sync, the PTP delay request and (if needed) the PTP delay response. In this case, $T_{S-Dr} = 1\ ms$ can be fulfilled if a proper scheduler is designed.

Finally, a dedicated procedure may be followed in proprietary wireless systems specifically designed for industrial applications, such as SHARP [3], or wireless systems for indoor localization [9], where time synchronization is a vital part of the system performance. In this case, the delay request could be sent as an ACK frame to the sync frames. With this procedure, $T_{S-Dr} < 0.2\ ms$ could be obtained, depending on the bandwidth of the wireless system and its physical layer complexity. For example, in SHARP using a bandwidth of 20 MHz, T_{S-Dr} may be lower than $80\ \mu s$ if a high order modulation is used.

VII. SIMULATION SETUP AND NUMERICAL RESULTS

The wireless physical layer used to test the synchronization performance is the physical layer of the IEEE 802.11n WLAN standard. The carrier frequency has been set to 2.412 GHz. The bandwidth of the system has been set to 20 MHz. A Single Input, Single Output (SISO) antenna configuration has been used.

The 802.11 frames comprise two main training sequences, the Short Training Field (STF), and the Long Training Field (LTF). The STF and LTF sequences have a total length of 160 samples each. The LTF is situated after the STF, and it is used to detect the frame start after the frequency offset correction. Therefore, the training sequence $s[l]$ is the LTF. Specifically, $s[l]$ is the last 128 samples of the LTF sequence (the cyclic prefix has been eliminated to gain robustness against inter-symbol interference). The 802.11 frames have been generated using the MATLAB® WLAN Toolbox™.

The system performance has been evaluated over the A, B, C and E 802.11 standard channel models [18]. These channel models are widely used to evaluate wireless systems and they represent four different environments, from small office to open space. The channel models have an RMS delay spread of 50 ns, 100 ns, 150 ns and 250 ns respectively.

The system performance has been also evaluated over mobile conditions, because the CIR variation is the main limitation of the synchronization scheme performance along with the SNR. Therefore, and to test the synchronization scheme over a high variety of scenarios, the simulations have been carried out over mobile conditions using mobile nodes from nearly static nodes to nodes running at 300 km/h. It should be noted that IEEE 802.11 physical layer is not designed for running at high speeds because 802.11 nodes are meant to be nearly static. Nonetheless, the proposed synchronization scheme could be used in other wireless systems meant to support high mobile conditions, such as 5G and LTE.

The wireless channel variation due to the movement of the nodes has been modeled with a Doppler spectrum following a Jakes model [34]. The maximum Doppler shifts have been calculated from the speed of the nodes, v, and the carrier frequency, f_c.

Regarding the PTP configuration, the PTP frame exchange period has been set to 1 s, which means that the whole frame exchange is performed every 1 s, and it is considered that $T_{S-Dr} = 1\ ms$. Furthermore, a Proportional-Integral Loop filter has been used to reduce the noise of the $\tilde{t}_0$ calculation. The filter was configured with K_i and K_p constants equal to $2.6 \cdot 10^{-3}$ and $5.5 \cdot 10^{-2}$ respectively.

The clock sampling period has been set to 50 ns, equal to the bandwidth of the system. The maximum clock drift of both master and slave clocks has been set to 10 ppm and the standard deviation of the clock jitter has been set to 8 ps following a normal distribution, which is a common value in oscillators used in wireless systems. The clock drift and the time start of the master and the slave clocks have been set to a random value for each simulation. The clock jitter moves the temporal position of the clock rising edges and hence its error is directly summed to the timestamping error. Furthermore, the jitter also modifies the transmitted and received signal shape, as the signal is not sampled at the perfect moment. Nonetheless, the SNR and channel impairments are the dominant error source. In the most favorable conditions ($SNR = 30\ dB$, $v < 0.1\ km/h$ and on channel A), they induce more than 100 ps and, hence, the clock jitter effects are negligible for common clock jitter values.

It has been found in preliminary simulations that the algorithm to estimate the start window shows little improvements for more than 2 iterations. Therefore, the number of iterations of the algorithm for the simulation has been set to $K_{it} = 2$. Besides, the length of the channel has been set to $T_h = 16T$ (800 ns), (equal to the cyclic prefix length in a 802.11 frame), and $T_g = 7T$ (350 ns), thus $N = 30$.

The simulation tool used to evaluate the performance of the enhanced timestamps is MATLAB®. The simulation has been carried out as follows. Firstly, the channel models are generated using the Power Delay Profile (PDP) obtained from [18] and the nodes speed configuration. Afterwards, a fixed number of PTP frame exchanges are performed and the slave clock is corrected in each frame exchange. Finally, the time synchronization error is calculated in each frame exchange as
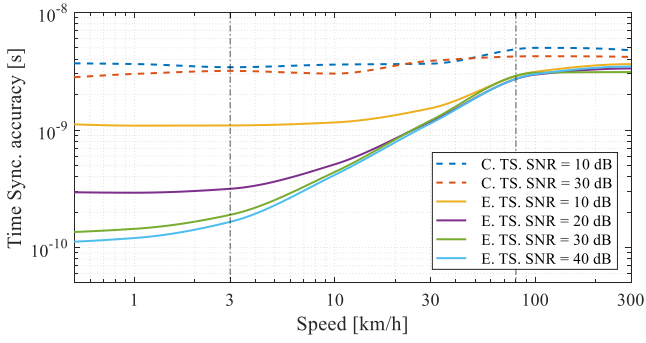
Fig. 6. Time synchronization accuracy over WLAN channel A (RMS delay spread = 50 ns) as function of the speed of the nodes for different SNR.
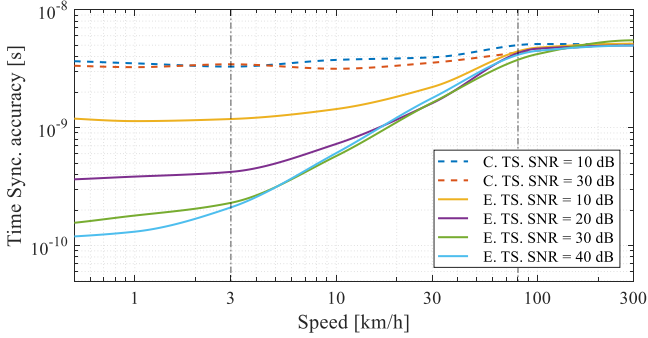

Fig. 8. Time synchronization accuracy over WLAN channel C (RMS delay spread = 150 ns) as function of the speed of the nodes for different SNR.

Fig. 7. Time synchronization accuracy over WLAN channel B (RMS delay spread = 100 ns) as function of the speed of the nodes for different SNR.

Fig. 9. Time synchronization accuracy over WLAN channel E (RMS delay spread = 250 ns) as function of the speed of the nodes for different SNR.

the difference between the master time and the slave corrected time. A total of 10^4 PTP frame exchanges have been carried out for each speed and SNR.

The results of the time synchronization accuracy are depicted in Fig. 6, Fig. 7, Fig. 8 and Fig. 9. The enhanced timestamps using *Algorithm 1* are labeled as (E. TS.) and the conventional timestamps are labeled as (C. TS.). The conventional timestamps are based on the detector stated in (10). The time synchronization accuracy is the combination of the precision and trueness of the measurement. Nonetheless, the trueness of the measurement is approximately equal to 0 ns in every simulation, and thus accuracy and precision match for the simulations shown in this section.

The results show that the enhanced timestamps have three different performance regions regarding the channel variation rate during the PTP sync and PTP delay request exchange: slowly time-variant ($v < 3$ km/h), mid time-variant (3 km/h $< v < 80$ km/h), and fast time-variant ($v > 80$ km/h). The three regions are noticeable in the results over the four channel models and are indicated by the vertical black dashed lines.

In the first region, the slowly time-variant region (< 3 km/h), the channel coherence time is 30 times higher than T_{S-Dr}, thus the wireless channel can be considered symmetric during the PTP frame exchange. This is clearly represented in the results as the proposed synchronization scheme obtains a time synchronization of less than 220 ps at an SNR of 30 dB over channel A and B, which is 25 times smaller than the achieved by the conventional timestamps under the same conditions. On the other hand, and compared to the results over channel A and B, the performance of the enhanced timestamps is slightly deteriorated in the simulation over channel C and severely deteriorated in the simulation over channel E. This is caused by the error of the correlation window. The number of CIR components increases and so it
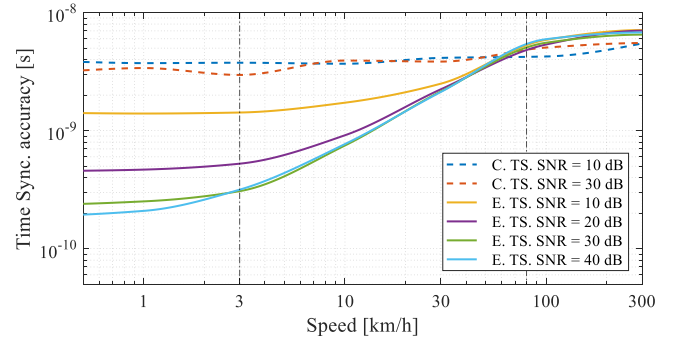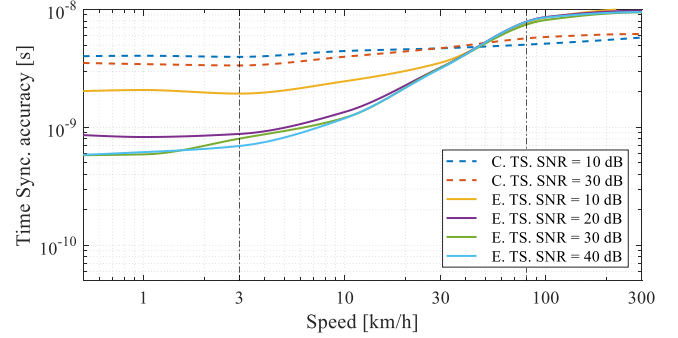
does the probability of misalignment in the integration window and the probability of missing some CIR components in the integration. This causes a small error in the ToA estimation, which is added to the time synchronization error. Furthermore, the performance bound over channel A and B, which is found at an SNR about 30 dB, is caused by the error of the rounding operation in step 3.3 of algorithm 1. Nonetheless, the bound is found for SNR that are unrealistic in wireless systems, and thus the use of more complex algorithms to implement the enhanced timestamps is not necessary.

In the second region (from 3 km/h to 80 km/h), the synchronization performance is linearly deteriorated as a function of the speed of the nodes. This is the expected behavior, as the channel coherence time is still very high, but it is not enough to consider a perfectly symmetric channel. Regarding the synchronization accuracy with conventional timestamps, it is not very affected by the changes in the environment, but their performance is still very far from the performance of the enhanced timestamps.

Finally, in the fast time-variant region (> 80 km/h) the synchronization performance of the conventional timestamps and the enhanced timestamps converges. This situation is reached at 80 km/h because the channel coherence time at such speed is approximately equal to $T_{S-Dr} = 1$ ms.

To gain more insight about the relation between the channel variation rate and the value of T_{S-Dr}, we have carried out one more simulation over the channel model B, but using $T_{S-Dr} = 0.1$ ms. $T_{S-Dr} = 0.1$ ms is a very challenging requirement in wireless communications, but it can be obtained following the guidelines stated in Section VI. The results of this simulation are depicted in Fig. 10. The results show that there is a strong relation between the time synchronization performance and T_{S-Dr}. In fact, the threshold between the first and the second region is shifted to the right from 3 km/h to 30 km/h. Therefore, the requirement of channel symmetry during the
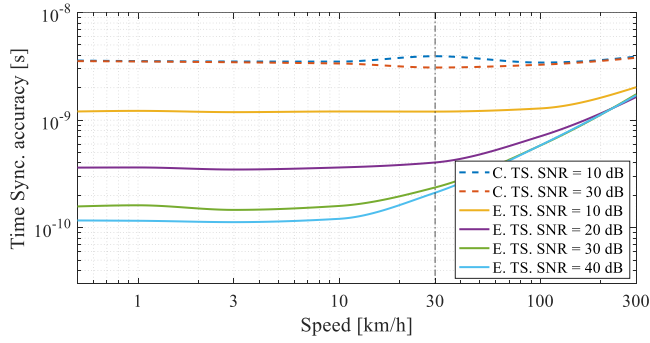
Fig. 10. Time synchronization over WLAN channel B (RMS delay spread of 100 ns) as function of the speed of the nodes for different SNR. $T_{S-Dr} = 0.1\ ms$.

PTP frame exchange has been verified numerically, and it has been shown that T_{S-Dr} has a great impact in the synchronization performance. Thus, to ensure almost perfect channel symmetry and obtain high time synchronization performance T_{S-Dr} should be at least 20-30 times lower than the channel coherence time.

VIII. CONCLUSIONS

In this paper, we deal with the inherent issues of delivering high-performance time synchronization through wireless systems. Time distribution protocols (i.e. PTP) rely on precisely estimating the frames ToA to synchronize the master clock to the slave clock. However, conventional ToA estimators have two main limitations over wireless: the timestamps quantization caused by the usually low bandwidth of wireless systems and the wireless channel behavior. The wireless channel behavior is the most challenging limitation, as wireless channels usually present multipath propagation, which causes time dispersion and deteriorates the ToA estimation. Furthermore, the multipath propagation is subject to variations caused by the movements of the nodes or environment changes, which dynamically varies the channel delay. Therefore, in this paper we have analyzed in detail these impairments and we have proposed a time synchronization scheme based on a novel timestamping method. We have analytically proven that the enhanced timestamping method precision is independent of the sampling period and that it offers a very robust ToA estimation over time-dispersive and time-variant channels.

The numeric simulations show that PTP combined with the enhanced timestamps can provide sub-nanosecond time transfer accuracy at slowly time-variant conditions, i.e. when the CIR can be considered constant during T_{S-Dr}. The results also show that the synchronization performance is deteriorated when the ratio between the channel coherence and T_{S-Dr} is lowered, because the channel cannot be considered symmetric as it is not invariant. The performance of the enhanced and conventional timestamps converge when the channel coherence is equal to T_{S-Dr}. Hence, it can be concluded that the enhanced timestamps greatly outperforms the conventional timestamps under the design assumption of channel symmetry between the master and slave.

The proposed synchronization scheme may be very useful in the implementation of wireless systems that needs accurate time synchronization, such as wireless industrial networks. Furthermore, its use may be very interesting in other applications that needs a network just to perform time synchronization, such as scientific experiments or wireless localization.

APPENDIX

Lemma 1. Proof.

Let be $f(t)$

$$f(t) = \sqrt{t}\, p(t). \tag{32}$$

The energy of $f(t)$ is

$$\|f(t)\|^2 = \int_{-\infty}^{+\infty} |f(t)|^2\, dt = \int_{-\infty}^{+\infty} |p(t)|^2 t\, dt, \tag{33}$$

taking into account that $p(t)$ is causal.

Now let consider

$$f[n] = f(t)|_{t=nT} = \sqrt{nT}\, p(nT), \tag{34}$$

$$p[n] = p(t)|_{t=nT} = p(nT). \tag{35}$$

$f(t)$ can be expressed as function of $f[n]$

$$f(t) = \sqrt{T} \sum_{n=-\infty}^{\infty} f[n]\, h(t-nT), \tag{36}$$

being $h(t)$ an ideal interpolator filter of unit energy

$$h(t) = \frac{1}{\sqrt{T}} \operatorname{sinc}\left(\frac{t}{T}\right), \tag{37}$$

and being

$$\operatorname{sinc}(t) = \frac{\sin(\pi t)}{\pi t}. \tag{38}$$

The energy of $f(t)$ can be expressed in terms of the energy of the sampled version

$$\|f(t)\|^2 = \int_{-\infty}^{\infty} |f(t)|^2 dt$$

$$= T \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[n]\, h(t-nT) \cdot \sum_{m=-\infty}^{\infty} f^*[m] h^*(t-mT)$$

$$= T \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[n]\, f^*[m] \int_{-\infty}^{\infty} h(t-nT) \cdot h^*(t-mT). \tag{39}$$

Due to

$$\int_{-\infty}^{\infty} h(t-nT) \cdot h^*(t-mT) = \operatorname{sinc}(m-n) = \delta_{m,n}, \tag{40}$$

being

$$\delta_{m,n} = \begin{cases} 1, & if\ m = n \\ 0, & if\ m \neq n. \end{cases} \tag{41}$$

Then

$$\|f(t)\|^2 = T \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[n]\, f^*[m]\, \delta_{m,n} = T \sum_{n=-\infty}^{\infty} |f[n]|^2$$

$$= T^2 \cdot \sum_{n=-\infty}^{+\infty} |p[n]|^2 n. \tag{42}$$

Finally,

$$\int_{-\infty}^{\infty} |p(t)|^2 \cdot t\, dt = T^2 \cdot \sum_{n=-\infty}^{+\infty} |p[n]|^2 n. \tag{43}$$

REFERENCES

[1] Piotr Lipnicki , Daniel Lewandowski, Diego Pareschi, Enrico Ragaini, and Pakos Waldemar, "Future of IoTSP – IT and OT integration," in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2018.

[2] P. Popovski, "Ultra-Reliable Communication in 5G Wireless Systems," *Int. Conf. 5G Ubiquitous Connect.*, 2014.

[3] Ó. Seijo, Z. Fernández, I. Val, and J. A. López-Fernández, "SHARP : A Novel Hybrid Architecture for Industrial Wireless Sensor and

Actuator Networks," in *IEEE International Workshop on Factory Communication Systems (WFCS)*, 2018.

[4] M. Luvisotto, Z. Pang, D. Dzung, M. Zhan, and X. Jiang, "Physical Layer Design of High Performance Wireless Transmission for Critical Control Applications," *IEEE Trans. Ind. Informatics*, vol. 3203, no. c, pp. 1551–3203, 2017.

[5] D. Mills, U. Delaware, J. M. Ed., ISC, J. Burbank, and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification," *IETF, RFC5905*, 2010. [Online]. Available: https://tools.ietf.org/html/rfc5905.

[6] *IEEE Standard for a precision clock synchronization protocol for networked measurement and control systems*. IEEE Standard 1588, 2009.

[7] F. Ramos, J. L. Gutiérrez-Rivas, J. López-Jiménez, B. Caracuel, and D. Javier, "Accurate Timing Networks for Dependable Smart Grid Applications," *IEEE Trans. Ind. Informatics*, vol. 14, no. 5, pp. 2076–2084, 2018.

[8] K. Hann, S. Jobert, and S. Rodrigues, "Synchronous ethernet to transport frequency and phase/time," *IEEE Commun. Mag.*, vol. 50, no. 8, pp. 152–160, 2012.

[9] A. Jiménez Ruiz and F. S. Granja, "Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis," *IEEE Trans. Instrum. Meas.*, 2017.

[10] M. Rizzi, M. Lipiński, P. Ferrari, S. Rinaldi, and A. Flammini, "White Rabbit clock synchronization : ultimate limits on close-in phase noise and short-term stability due to FPGA implementation," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 2018.

[11] A. Makki, A. Siddig, M. Saad, J. R. Cavallaro, C. J. Bleakley, and S. Member, "Indoor Localization Using 802 . 11 Time Differences of Arrival," *IEEE Trans. Instrum. Meas.*, 2015.

[12] "ISO Accuracy (trueness and precision) of measurement methods and results -- Part 1: General principles and definitions," *ISO Stand. 5725-11994*, 1994.

[13] H. Guo, S. Member, and P. Crossley, "Design of a Time Synchronization System based on GPS and IEEE 1588 for Transmission Substations," *IEEE Trans. Power Deliv.*, 2016.

[14] M. Weiss, "Getting accurate time from GNSS receivers: Considerations to approach nanosecond time," *IEEE Int. Symp. Precis. Clock Synchronization Meas. Control. Commun. ISPCS*, 2017.

[15] A. Mahmood, R. Exel, and T. Sauter, "Delay and jitter characterization for software-based clock synchronization over WLAN using PTP," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1198–1206, 2014.

[16] A. Mahmood, R. Exel, H. Trsek, and T. Sauter, "Clock synchronization over IEEE 802.11 - A survey of methodologies and protocols," *IEEE Trans. Ind. Informatics*, vol. 13, no. 2, pp. 907–922, 2017.

[17] Ó. Seijo, I. Val, J. A. López-fernández, and M. Vélez, "IEEE 1588 Clock Synchronization Performance over Time-Varying Wireless Channels," *IEEE Int. Symp. Precis. Clock Synchronization Meas. Control. Commun. ISPCS*, 2018.

[18] V. Erceg *et al.*, "TGn Channel Models," 2004.

[19] R. Exel, "Clock Synchronization in IEEE 802.11 Wireless LANs using Physical Layer Timestamps," in *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication, ISPCS*, 2012.

[20] R. Exel, "Receiver Design for Time-Based Ranging with IEEE 802.11b Signals," *Int. J. Navig. Obs.*, vol. 2012, 2012.

[21] R. Exel and T. Bigler, "ToA Ranging using Subsample Peak Estimation and Equalizer-based Multipath Reduction," *IEEE Wirel. Commun. Netw. Conf.*, 2014.

[22] P. Ferrari *et al.*, "Timestamping and Ranging Performance for IEEE 802.15.4 CSS Systems," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 5, pp. 1244–1252, 2014.

[23] P. Ferrari, G. Giorgi, C. Narduzzi, S. Rinaldi, and M. Rizzi, "Timestamp Validation Strategy for Wireless Sensor Networks Based on IEEE 802.15.4 CSS," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 11, pp. 2512–2521, 2014.

[24] C. M. De Dominicis *et al.*, "Timestamping of IEEE 802.15.4a CSS Signals for Wireless Ranging and Time Synchronization," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 8, pp. 2286–2296, 2013.

[25] Ó. Seijo, I. Val, and J. A. López, "On the use of White Rabbit for Precise Time Transfer in 5G URLLC Networks for Factory Automation Applications," *IEEE Int. Conf. Ind. Cyber-Physical Syst.*, pp. 1–6, 2019.

[26] M. Appel, F. Winkler, and B. Meffert, "Phase-based High-Precision Synchronization for Wireless Networks using FPGAs," *2018 IEEE Int. Symp. Precis. Clock Synchronization Meas. Control. Commun.*, pp. 1–6, 2018.

[27] D. Shrestha, Z. Pang, and D. Dzung, "Precise clock synchronization in high performance wireless communication for time sensitive networking," *IEEE Access*, vol. 6, pp. 8944–8953, 2018.

[28] U. Mengali, *Synchronization Techniques for Digital Receivers*. Springer, 1997.

[29] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall.

[30] F. Pérez Fontán and P. Mariño Espiñeira, *Modeling the Wireless Propagation Channel: A Simulation Approach with MATLAB®*. Wiley, 2008.

[31] J.-J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson, "On channel estimation in OFDM systems," *IEEE 45th Veh. Technol. Conf. Countdown to Wirel. Twenty-First Century*, 1995.

[32] H. Zhao and J. Yu, "A Simple and Efficient Design of Variable Fractional Delay FIR Filters," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 53, no. 2, pp. 157–160, 2006.

[33] *IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE, 2012.

[34] M. Mfeze and E. Tonye, "Comparative Approach of Doppler Spectra for Fading Channel Modelling by the Filtered White Gaussian Noise Method," *Int. J. Comput. Sci. Telecommun.*, vol. 6, no. 11, 2015.