Improving the precision of time synchronization protocols in ultra-wideband networks estimating the time of flight of the radio signal

Juan J. Pérez-Solano[a,*], Antonio Soriano-Asensi[a], Santiago Felici-Castell[a], Jaume Segura-Garcia[a]

[a] *Departament de Informática, ETSE, Universitat de València, Avd. de la Universidad S/N, 46100 Burjassot, (Valencia), Spain*

Abstract

Modern applications of wireless sensor networks such as the conduction of collaborative tasks, the distributed detection of events or the alignment of data samples gathered by several nodes, require a precise time coordination of the nodes composing the network. Ultra Wide Band (UWB) technology has enabled the development of precise time synchronization protocols. The intrinsic properties of this technology allow an accurate estimation of the time of flight (ToF) of the radio signal in scenarios covering short distances. The integration of UWB transceivers in Wireless Sensor Networks (WSN) nodes is nowadays possible due to the availability of commercial chips implementing this technology. This combination facilitates the development of accurate time synchronization protocols that establish a global common clock throughout the network. In this context, the main contribution of this work is a protocol that allows the synchronization of the local clocks of these nodes with a common global reference kept in the master node. The master periodically broadcasts synchronization packets to keep the rest of the nodes synchronized. However, since the precision achieved in the time synchronization of the local clocks with UWB technology is comparable to the ToF of the RF signal for distances in the order of several meters, some considerations to estimate this delay are required. In this paper, a new algorithm is proposed that performs the estimation of the ToF of the UWB signal as well as the implementation of linear models to synchronize the local clocks of the network node, eliminating the dependence on the ToF. The experiments conducted with real devices reveal that an accuracy of tens of nanoseconds can be achieved regardless of the distance separating the network nodes.

Email addresses: juan.j.perez@uv.es (Juan J. Pérez-Solano), antonio.soriano-asensi@uv.es (Antonio Soriano-Asensi), felici@uv.es (Santiago Felici-Castell), jsegura@uv.es (Jaume Segura-Garcia)

Keywords: Time Synchronization, Ultra Wide Band, IEEE 802.15.4, Linear Regression, Wireless Sensor Networks

1. Introduction

Wireless Sensor Networks (WSN) constitute a pervasive and ubiquitous technology that has been applied successfully in a vast number of relevant applications for our society, such as: environmental monitoring [1], smart cities [2], intelligent transport systems [3], health care [4], smart grids [5], and industrial automation [6]. An important aspect that must be considered in the development of these networks is the maintenance of a global time to keep the local clocks of the nodes synchronized. This global time enables the temporal alignment of the data samples coming from different nodes and the realization of in-network collaborative tasks. The timing service provided by Global Navigation Satellite Systems (GNSS) represents an accurate solution that can be used to establish a global network clock. The term GNSS encompasses different systems such as the USA's Global Positioning System (GPS), Russia's GLONASS, Europe's GALILEO and China's BEIDOU. The time synchronization accuracy achieved with the GNSS service that relies on pseudo-range information and carrier phase analysis is in the order of tens of nanoseconds [7]. However, this time synchronization mechanism is not suitable for degraded or denied GNSS scenarios, such as indoor or underground environments where the GNSS signals are not available. In these conditions, non-GNSS time synchronization methods must be applied. This work addresses these scenarios without GNSS coverage in which synchronization must be performed by means of some alternative wireless communication technology.

Low-cost hardware embedded in the network nodes limits the stability of the local clocks, producing a continuous change of the relative offsets among them [8, 9]. Time synchronization protocols in WSN estimate periodically the offset between the different nodes local clocks transmitting information through a wireless communication link. Usually, these protocols perform a sequence of packets exchanges to collect timestamps at the medium access layer. Timestamps at different nodes are captured using their own local clocks and the synchronization protocol combines them to estimate their offsets. However, the drift of the local clocks causes this estimate to become out of date after a short period of time. Besides, the accuracy of the estimate is affected by the precision of the node local clock as well as the capturing process after transmitting or receiving the packets. In this context, the integration of low-cost Ultra Wide Band (UWB) transceivers in the network nodes facilitates the implementation of accurate time synchronization protocols due to the high time resolution of the pulses transmitted with this technology.

The IEEE 802.15.4 - 2011 standard [10], which establishes the features of the physical layer of this radio technology, defines a signal with a bandwidth greater than 500 Mhz composed by pulses of very short duration (in the order of nanoseconds) referred as Impulse Radio UWB. This standard is focused on the implementation of localization algorithms and applications requiring low data rates. The UWB radio frequency pulses reduce the influence of multipath interference during the signal reception and allow a precise determination of the signal transmission and reception timestamps. The high time resolution makes this technology a perfect candidate for implementing localization systems, since accurate estimates of the Time of Flight (ToF) of the signal can be obtained. Current UWB transceivers include high frequency timers to provide very accurate timestamps, which enables the implementation of ranging algorithms with an error of less than 10 cm [11].

The standard IEEE 802.15.4 - 2015 [12] adds a new physical layer and distinguishes the two versions using different names: a) High-Rate Pulse (HRP), which refers to the physical layer defined in the initial standard, and b) Low-Rate Pulse (LRP), which denotes the new physical layer added. The main difference between them is the rate at which the pulses are transmitted. HPR mode transceivers transmit more but smaller pulses than in LPR mode, although remaining the maximum transmitted energy the same in both cases. Due to the higher peaks used in LPR, the instantaneous signal to noise is better and the complexity and the power consumption of the receiver is greatly reduced. This makes LPR transceivers more suitable for applications where the devices must be powered by batteries. Finally, the standard IEEE 802.15.4z - 2020 [13] proposes new enhancements for both physical modes (HRP and LWP) to im prove the signal detection with new coding, packet preambles and modulations.

In this regard, the work in [14] has shown how the interconnection of commercial UWB transceivers, such as the Qorvo's DW1000 [15] based on the IEEE 802.15.4 - 2011 standard, to a microcontroller timer allows the development of protocols capable of reducing the synchronization error between two nodes to values of tens of nanoseconds, which is close to the timers resolution. The capturing process of the timestamps at the microcontroller is supported by the activation of an output at the UWB transceiver after the detection of the Start of Frame Delimiter (SFD) field, which is included in the IEEE 802.15.4 UWB packet format. This output is connected to the capture input of the micrcontroller timer to acquire precise timestamps just when the SDF field is detected at the receptor transceiver. The acquisition of these timestamps relies on hardware elements and eliminates random delays that can appear at the software layers of the protocol. Once a set of timestamps is saved, the protocol performs a linear regression to predict future offset values between the clocks of the nodes. However, the acquisition of the timestamps carried out in this work are based on the one way packet transmission. This mechanism simplifies the packet exchange protocol between the network nodes. It only requires a packet transmission from the master node, which is the node with the global network time, while the other nodes (slaves) will capture and track it. Notice that the ToF of the UWB signal cannot be obtained with this packet transmission scheme and the error produced by the existence of different ToF values can not be eliminated. Therefore, the error committed by not considering the different ToF delays associated with each node, prevents the algorithm to achieve an accurate time synchronization when the difference in the radio range between nodes is greater than 10m [14].

In this paper, we propose a new packet exchange mechanism between the master and the network nodes that allows an efficient estimation of the ToF of the UWB signal. Once the ToF is estimated and cancelled, the resulting synchronization protocol can predict future offset values applying linear regression with a time precision that does not depend on the node position. The performance of the protocol has been validated in an experimental deployment with static nodes.

The paper is divided in the following sections. Section 2 is devoted to presenting the related work. In section 3, the mechanism to acquire the timestamps at the different nodes jointly with the ToF estimation is explained. Section 4 describes the synchronization algorithm applied to predict the global time with ToF cancellation. In section 5, the conducted experiments and the obtained results are presented. Finally, section 6, includes the conclusions and summarizes the contributions of this paper.

2. Related Work

Time synchronization is a major issue in the development of WSN. The local clock of the nodes is commonly implemented with a timer that receives periodic impulses from an oscillator that integrates a quartz crystal. The stability of the crystal is indicated by the manufacturer, which provides the maximum deviation from the nominal frequency in ppm (parts per million). The divergence from the nominal frequency is caused by changes in the environmental conditions such as temperature, pressure and acceleration, but also by the supply voltage, the board capacitors, aging factors and imperfections in its manufacture [8, 9]. All these effects produce an unpredictable and continuous drift of the oscillation frequency and change the relative skew that exists between clocks. In this sense, WSN synchronization protocols deal with these clocks drift and try to establish a common global clock throughout the whole network. In this section a revision of the more important protocols found in the literature is provided.

The first group of synchronization protocols considered is focused on protocols supported by the initial version of the IEEE 802.4.15 standard, often referred to as Zigbee, or wireless technologies with similar capabilities in the 2.4Ghz frequency band. The two main references in this group are: Timing-sync Protocol for Sensor Networks (TPSN) [16] and Reference Broadcast Synchronization (RBS) [17]. The former is a pairwise synchronization algorithm that collects four timestamps following a sender-receiver [16] mechanism after the transmission and reception of two packets. These timestamps allows the algorithm to calculate the instantaneous offset between the clocks of the two nodes with a synchronization precision of 16.9 μs using Mica nodes [16]. The later proposes a different approach based on a receiver-receiver procedure. In this case, a broadcast packet is sent from the reference node and is received by the rest of nodes located in its same radio range. After capturing the reception timestamps, all the nodes inside the broadcast domain exchange their timestamps and estimate their relative offsets. The precision achieved with this protocol employing Mica nodes is 11.2 μs.

The above protocols are mainly focused on short-term time synchronization as they estimate the offset just after exchanging the synchronization packets. However, the offset between the clocks of the nodes changes continuously due to the clocks drift and the estimate becomes out of date after a short time. Within this framework, there are some alternative protocols that model the different skew of the clocks by means of a linear regression to achieve better estimates of future offsets. These protocols provide long-term time synchronization and estimate periodically the relative skew between local clocks considering the previous timestamps. The Flooding Time-Synch Protocol (FTSP) [18] pertains to this group of protocols. FTSP can synchronize nodes in a multi-hop network transmitting broadcast packets with a period of 30 s and applying linear regression to estimate the relative skew of the clocks. The experimental results of this protocol show that a precision of 1.48 μs in a one hop network with Mica2 nodes can be achieved. Scaling up the network to 60 motes in a 6-hop topology the accuracy slightly worsens to 3 μ. Tiny-Sync and Mini-Sync [19] protocols follow the same approach and apply linear regression to obtain the relative skew and offset between two nodes but establishing upper and lower bounds to these values to improve the estimation accuracy. An important issue that must be considered with these algorithms is the window size, i.e. the number of previous timestamps considered in the linear regression. There are some protocols that attempt to obtain an optimal window size, because short ones limit the accuracy of the model and too long sizes are not suitable to follow the drifts of the local clocks. Rate Adaptive Time Synchronization (RATS) in [20] is an example of protocol that selects the optimum window to minimize the synchronization error. In that work is shown how using Mica2 devices the error can be minimized to a value of 1.6 μs employing a time window of 8 minutes. Another example is the Adaptive Window Size (AWS) protocol [21], where an algorithm to obtain the optimum window size is proposed. The results of this work show that a pairwise synchronization with TelosB nodes can achieve an average accuracy of 8.95 μs, whereas in a 4-hop topology the accuracy worsens to 12.42 μs. Finally, the work in [22] presents the Gradient Time Synchronization Protocol (GTSP), which introduces a completely distributed protocol where the nodes periodically broadcast synchronization beacons to their neighbours. After some time, the logical clock estimated by each node converges to a common logical clock establishing the global synchronization. Experiments on a testbed with a network of 20 Mica2 devices demonstrates that the average synchronization error achieved in this case is 4μs.

Protocols presented so far relied on wireless transceivers that are compliant with early versions of the IEEE 802.4.15 standard. A significant improvement in the accuracy of the protocols can be fulfilled considering UWB communications supported by the standards IEEE 802.15.4 - 2015 and IEEE 802.15.4z - 2020. Currently, there are commercially available low-cost UWB transceivers, such as the Qorvo's DW1000 UWB transceiver [15], which allow the acquisition of very accurate timestamps during the packet transmission or reception. Moreover, the timestamps are captured using the internal timers of the UWB transceivers that are connected to high frequency clocks (64 GHz in the case of the DW1000 transceiver). Main applications of the networks built with such devices are related to the implementation of indoor localization systems. In this scenario, the network is normally composed of a group of distributed anchors, placed at known locations, and there is one or more tags whose coordinates must be estimated. The position estimation can be computed basically by two different types of measures: Time of Flight (ToF) or Time Difference of Arrival (TDoA). The acquisition of ToF measures of the signal transmission between the tag and the anchors requires the use of a Two-Way Ranging (TWR) mechanism [23], similar to the sender-receiver procedure of the TPSN protocol. Some alternative TWR methods have been proposed in the literature to avoid the inaccuracies produced by clocks drift. The Symmetrical Double-Sided Two-Way Ranging (SDS-TWR) [10] method utilizes three packets to improve the accuracy of the ToF measurement. However, this method requires equal reply times during the reception and the transmission of the packets involved in the process in both nodes, which is difficult to accomplish. This requirement can be avoided with the alternative double-sided TWR (ADS-TWR) [23], which still needs the transmission of three packets but estimates the ToF in a non-linear manner. Thus, the ADS-TWR error depends only on the clock drift of one device and reduces significantly the complexity of the packet exchange process. On the other hand, localization methods based on TDoA follow the One-Way Ranging (OWR) mechanism, which requires the transmission of only one packet from the tag to the anchors or vice versa. However, these methods need a very strict time synchronization between the clocks integrated in the transceivers to determine the time difference of arrival accurately [24]. To enhance the synchronization accuracy, the work in reference [25] utilizes two reference nodes with DW1000 transceivers that periodically broadcast the synchronization packets. The primary reference node transmits synchronization packets that are received by all the anchors and one of them serves as a secondary reference to retransmit this information after a predefined delay. In this way, anchors that are too far can receive the synchronization packets from the secondary reference. The protocol compensates the effect of the clocks drift and achieve a precision of 1.5 ns for 90 % of the TDoA measurements. Another example is the work in [26], which proposes a master-slave synchronization protocol similar to the TPSN between the anchor with the reference clock and the rest. Besides, the protocol performs the calculation of the skews between different clocks and employs a Kalman filter to reduce the noise in the collected timestamps. Simulation results included in the paper prove the improvement achieved in the TDoA estimation. In the same line, the algorithm in [27] carries out a TDoA localization procedure transmitting packets under the OWR method. It avoids the influence of the clocks drift and estimates the position of the tags applying an extended Kalman filtering. In [28] the authors propose the Broadcast Relative Localization and Clock Synchronization (BLAS) algorithm that carries out a joint localization and clock synchronization procedure for multiagent applications. The algorithm collects real time spatio-temporal information that is distributed among the nodes to find pairwise clock parameters applying Kalman filters. The ranging accuracy accomplished in this way is equivalent to a clock synchronization of 0.172 ns. The work in [29] presents an optimized clock synchronization protocol that estimates the distance and the delay of the UWB signal transmission with DW1000 transceivers. The algorithm uses a proportional-integral estimator that makes the error converge to a sub-nanosecond time scale. Finally, the work in [30] introduces an asymptotic gradient clock synchronization, which addresses the chaotic nature of the gradient synchronization and improves the estimation of the relative clocks dynamics with a Kalman filter. The results obtained show a synchronization accuracy in the order of tens of picoseconds. It should be notice that the synchronization accuracy achieved in all these protocols relays on the timestamps captured with the local clock of the DW1000 UWB transceiver. This timer uses a clock signal with a frequency of 64 GHz and a time resolution of 15.6 ps. However, this timer cannot be used as the local clock of the node to get timestamps triggered by the internal microcontroller events.

It is worth mentioning some other synchronization algorithms that combine the DW1000 transceiver with some other hardware components to capture timestamps on the microcontroller timers or generate a Pulse Per Second (PPS) type output signal. The work in [31] proposes a time synchronization approach that is conceived as a side effect of the indoor localization procedure implemented with the DWM1001 Real-Time Location System (DRTLS) kit. The authors take advantage of the SFDLed output pin of the DW1000 UWB transceiver, which is connected to the capture input of one timer, to obtain a reference signal for the reception timestamps. The results of this work characterize the OWR time synchronization performance of two nodes, obtaining a maximum jitter of 3.3 μs and a standard deviation of 0.7 μs.

Another example is the work in [32], where the micrcontroller and the UWB transceiver share an external 38.4 MHz reference clock to enhance the precision of the captured timestamps. In addition, a Phase-Locked Loop (PLL) circuit based on a chip-scale atomic clock has been integrated to increase the clock stability. With this platform and a synchronization protocol based on the TWR mechanism to cancel the ToF, the nodes provide a PPS output signal with an average accuracy of 2.12 ns and a standard deviation of 0.84 ns. However, although this reference presents outstanding synchronization results, the cost of the hardware components employed hampers its practical use in applications requiring a large number of nodes like WSN. In [33] a synchronization algorithm denoted as Wicsync is presented. The network nodes have a PPS output and are based on the DWM1000 UWB transceiver and additional hardware implemented in a FPGA. The protocol relies on the TWR to collect the timestamps at the local clocks of the UWB transceivers. The experiments conducted show an instantaneous clock synchronization accuracy of 3.4 ns and an average time error of 0.141 ns covering distances up to 75m. Finally, reference [14] proposes a synchronization protocol utilizing low-cost hardware components and the DW1000 UWB transceiver. The protocol allows the synchronization of all the nodes in the same radio range of the master node. It is based on the OWR mechanism and the SFDLed output pin of the transceivers to sample the timestamps at the timers of the microcontrollers. The main contribution of this work is the proposal of a linear regression algorithm to achieve long-term synchronization of the local clocks of the microcontrollers with an accuracy in the order of tens of nanoseconds. However, the main limitation of this protocol is the impact that the distance separating the nodes has on the algorithm performance, since the ToF of the radio signal begins to be comparable to the error committed for distances greater than 5 meters.

Table 1 summarizes the synchronization protocols presented in this section. The table provides the mean error of the synchronization protocol, the wireless technology used, jointly with the platform and the transceiver integrated in the network nodes. The synchronization timer shows the hardware component that implements the register where the local clock is kept and the oscillator frequency.

As a conclusion, it should be pointed out that although there are many time synchronization algorithms for UWB devices in the literature most of them are designed to meet the requirements of the TDoA-based indoor localization. The main goal of these methods is to synchronize the internal

Table 1: Summary of time synchronization protocols with different wireless technologies and platforms. The Synchronization Timer column indicates the hardware that comprises the register where the local clock is implemented and the frequency of its oscillator.

Reference	**Mean error (one hop)**	**Technology/Platform/Transceiver**	**Synchronized timer with clock frequency or output signal**
TPSN [16]	16.9 μs	ASK 2.4 GHz/Mica/TR1000	Processor timer with 4MHz
RBS [17]	11.2 μs	ASK 2.4 GHz/Mica/TR1000	Processor timer with 500KHz
FTSP [18]	1.48 μs	FSK 2.4 GHz/Mica2/CC1000	Processor timer with 7.37MHz
RATS [20]	1.6 μs	FSK 2.4 GHz/Mica2/CC1000	Processor timer with 125KHz
GTSP [22]	4 μs	FSK 2.4 GHz/Mica2/CC1000	Processor timer with 921KHz
TinySync [19]	10.78 μs	Zigbee/Telos/CC2420	Processor timer with 32KHz
AWS [21]	8.95 μs	Zigbee/Telos/CC2420	Processor timer with 32KHz
[25]	1.5 ns	UWB/-/DW1000	TDoA using DW1000 timer with 64GHz
[26]	0.2 ns	simulation	TDoA estimation
[27]	0.8 ns	UWB/-/DW1000	TDoA using DW1000 timer with 64GHz
BLAS [28]	0.172 ns	UWB/STM32F427/DW1000	ToA using DW1000 timer with 64GHz
[29]	> 1 ns	UWB/STM32F103/DW1000	DWM1000 timer with 64GHz
[30]	40 ps	UWB/DWM1001/DW1000	DWM1000 timer with 64GHz
[31]	0.7 μs	UWB/DWM1001/DW1000	SFDLed output signal
Pulsar [32]	2.12 ns	UWB/Specific/DW1000	PPS output signal
Wicsync [33]	0.141 ns	UWB/Specific/DW1000	PPS output signal

local clocks of the UWB transceivers of the system anchors. Thus, the algorithm presented in [30], which presents a significant synchronization accuracy, has been conceived to deal with the intrinsic features of the DW1000 local timer and the error produced during the timestamp capturing process in this transceiver. This fact prevents a direct application of this algorithm to the timestamps obtained in the microcontroller timer. Other approaches, such as [32] and [33], are focused on the development of synchronization hardware elements that provide an accurate PPS synchronization output, but they rely on specific and expensive hardware elements to achieve this objective. In contrast, our goal is the synchronization of the internal clocks of the low-cost microcontrollers embedded in the network nodes. The work presented in this paper is a continuation of the work included in reference [14], but in this case we propose a new algorithm that estimates the ToF of the RF signal and eliminates its impact in the synchronization process to improve the protocol accuracy. Therefore, the main contributions of this paper are twofold. On the one hand, the new protocol optimizes the acquisition of the timestamps at the network nodes to estimate the ToF accurately. On the other hand, the implementation of a long-term synchronization algorithm based on linear regression that is able to compensate for the error produced by the different ToF values estimated at every node.

3. Timestamps acquisition and ToF estimation

This section presents the protocol proposed to perform the acquisition of the timestamps in the UWB transceivers and the microcontrollers of the network nodes. The protocol considers that there is a master node responsible of maintaining and disseminating the network global time. The master periodically broadcasts packets for the other nodes to synchronize their local clocks. All the network nodes are assumed to be within the radio range of the master and keep their own local clocks in a timer of the microcontroller. The transmission and reception of a packet produces the acquisition of timestamps at the UWB transceiver, but the capture of the timestamps at the microcontroller is performed only on the reception. This is due to the time capturing process implemented in the microcontroller, determined by SFDLed output pin of the UWB transceiver. The SFDLed is activated only in reception and as it was characterized in reference [14]. As it is shown in [14], the activation of the SFD line on two UWB transceivers placed side by side exhibited an average time difference and standard deviation of 2.3 ns and 22.4 ns, respectively. Therefore, the uncertainty in the SFDLed line activation is low enough to allow the implementation of precise synchronization protocols with the timestamps captured at the internal timers of the microcontroller.

3.1. Synchronization packets transmission

As it is shown in Fig. 1, the synchronization algorithm will require an accurate estimation of ToF of the RF signal received from the master and for this, the protocol takes advantage of the UWB transceivers to get it. As it is stated in the previous section, the ADS-TWR algorithm provides the most precise estimate of the ToF. It requires requires the exchange of three packets and the acquisition of 6 timestamps, (T1 to T6) in Fig. 1, at the UWB transceivers.

The estimation of the ToF can be performed as it is explained in [23]:

$$TOF = \frac{T_{round1} \cdot T_{round2} - T_{reply1} \cdot T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}} \tag{1}$$

For our goal, ADS-TWR must be performed sequentially between the master and each node in the network. The ADS-TWR requires the transmission of three packets per node, however an optimized protocol is proposed

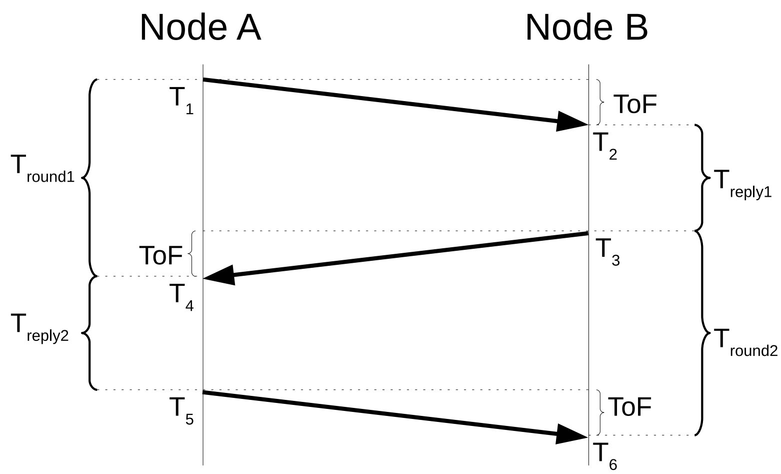

Figure 1: Three way packet exchange.

to reduce the total amount of packets. The sequence of synchronization packets transmitted to capture the timestamps is depicted in the Fig. 2. Every node is assigned an address which establishes the order in which they perform the packet exchange with the master. In the header of each packet, the master includes the address of the intended receiver that responds conforming a two way packet exchange. The third packet servers to capture the last timestamps of the ADS-TWR procedure, but it is also the first packet transmission for the next node. Thus, using this protocol only two packets per node are needed to implement the ADS-TWR algorithm, with the exception of the last node that requires a one final packet more to gather the 6 timestamps. As it is represented in Fig. 2, the UWB transceivers provide timestamps in both cases, transmissions and receptions, these values are denoted as $T_{UWBn,m}$ where n is the node address and m is the timestamp number. In contrast, the microcontroller can only capture timestamps during packet reception, producing the marks $T_{MCUn,m}$, where n and m have the same meaning. The master regularly initiates a new round of synchronization packets transmission with a Synchronization Period (SP) that is set in advance.

3.2. ToF estimation

The protocol depicted in Fig. 2 allows every node to estimate the ToF of the RF signal during the packet exchange with the master. However, since

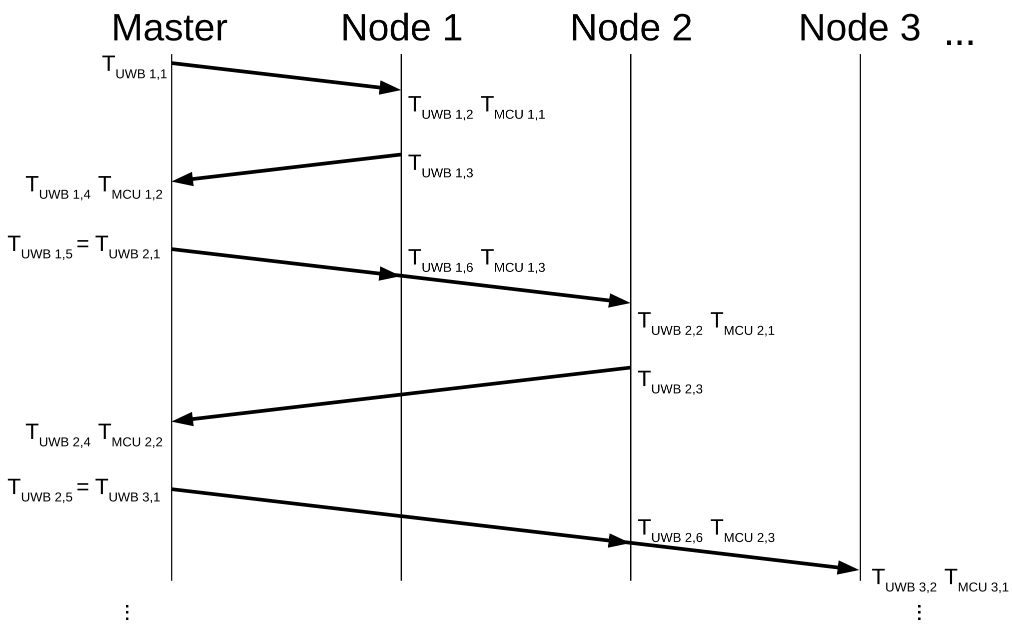

Figure 2: Sequence of synchronization packets transmitted to capture the timestamps at every node according to the proposed protocol.

the nodes are assumed to be static this value remains stable throughout the network operation time. Thus, it can be established a bootstrap procedure in which the nodes estimate the ToF using the ADS-TWR algorithm. Once all the nodes determine their ToF, the protocol in Fig. 2 changes to the OWR scheme since the nodes responses to the master synchronization packets are no longer needed. The master is aware of the end of this process by the lack of response packets from the nodes. At this stage, when all the nodes have completed the estimation of the ToF, the protocol changes and the master transmits only one broadcast packet periodically to synchronize all the nodes, reducing significantly the total amount of packets sent.

Notice that the ToF estimation stage is divided in two steps. The first one comprises the time required by the timers of the UWB transceivers to reach a steady operation. As it was reported by other authors [34] and the manufacturer of the transceiver [35], the instability at the beginning is related to the warm-up period of the crystal oscillators when each node starts its operation. The duration of this period can be determined by checking the internal temperature of the UWB transceiver. Once the UWB chip reaches a steady operating temperature, the UWB clock becomes stable and the ToF estimation is more precise. Since the crystal oscillator stability depends on the variations of the internal temperature of the UWB transceiver, during this step the node waits for the stabilization of this temperature. Then, when the node boots up, the temperature rises rapidly, and after some minutes it reaches a steady value that remains stable during the rest of the network operation time. So, the detection of the end of the warm-up period is performed by the nodes sampling the UWB internal temperature and calculating an approximation to its first derivative. The instant at which the approximation to the first derivative becomes zero determines the end of this period allowing the node to start the estimation of the ToF avoiding the initial fluctuations.

The second step is focused on is focused on the calculation of the ToF applying Eq. 1 to the UWB timestamps acquired with the synchronization packets. With every new transmission initiated by the master, the nodes calculate a new ToF value. These transmissions are repeated periodically after the SP, which constitutes a configuration parameter of the protocol. Nonetheless, these ToF values must be considered as random samples, since the timestamps acquisition process is affected by uncertainties in the signal reception and the SFD detection. After getting a sequence of samples, the nodes can calculate the average to achieve a more precise estimate of the real ToF. Thus, the goal of this step is to obtain an accurate estimate of the mean value of the ToF, denoted as μ_{ToF}, from the samples acquired $ToF(i)$. The estimate is calculated as:

$$\overline{ToF} = \sum_{i=1}^{n} \frac{ToF(i)}{n} \tag{2}$$

The divergence of this estimate from μ_{ToF} is given by the confidence interval. Thus, if $2\Delta_{ToF}$ is the confidence interval and $(1-\alpha)$ is the corresponding confidence level with $0 < \alpha < 1$, both quantities are related as:

$$P(\mu_{ToF} - \Delta_{ToF} \leqslant \overline{ToF} \leqslant \mu_{ToF} + \Delta_{ToF}) = 1 - \alpha \tag{3}$$

which implies that after taking a given number of samples n, $\overline{ToF}$ lies within the confidence interval $2\Delta_{ToF}$ in a proportion equal to $(1 - \alpha)\%$. When ToF_i are observations of an independent and normally distributed random variable, the confidence interval for the unknown average μ_{ToF} is:

$$2\Delta_{ToF} = \overline{ToF} \pm t_{n-1,1-\frac{\alpha}{2}} \cdot \widehat{\sigma_{ToF}} \tag{4}$$

where $t_{n-1,1-\frac{\alpha}{2}}$ is the $(1-\frac{\alpha}{2})$ critical value for the Student's t distribution of a two-sided test with $(n-1)$ degrees of freedom and $\widehat{\sigma_{ToF}}$ is the estimate of the standard error of the mean:

$$\widehat{\sigma_{ToF}} = \sqrt{\sum_{i=1}^{n} \frac{(ToF(i) - \overline{ToF})^2}{n(n-1)}} \tag{5}$$

Eq. 4 can also be used in the case of independent and identically distributed random variables, since according to the central limit theorem, as the number of samples n increases the distribution of $\overline{ToF}$ approaches the normal distribution. Thus, when the number of samples is $n > 100$, the condition of being independent and identically distributed random variables is sufficient to obtain an accurate result. Furthermore, when $n > 30$ the Student's t distribution of Eq. 4 can be replaced by the standard normal distribution and the value $t_{n-1,1-\frac{\alpha}{2}}$ becomes $z_{1-\frac{\alpha}{2}}$, which represents the critical value for the standard normal distribution with a two-sided test and a confidence level $(1-\alpha)$. From the tables of the standard normal distribution, it is observed that for a two-sided test and a confidence level of 95% the critical value is ± 1.96. Applying this approximations, the confidence interval amplitude (CIA) can be defined as:

$$CIA = 1.96 \cdot \widehat{\sigma_{ToF}} \tag{6}$$

The calculation of the CIA provides an assessment of the accuracy of the ToF estimate and indicates whether the number of samples is sufficient to achieve the required precision. Thus, after the warm-up step, the nodes start the acquisition of ToF samples for the estimation of the mean of the ToF applying Eq. 2. In general, as the number of samples increases the CIA becomes narrower and the ToF estimate becomes more accurate. This bootstrap procedure ends when the CIA is below a specific threshold that guarantees the achievement of the expected accuracy. Although this method allows the automatic adjustment of the number of ToF samples required to estimate the mean, a minimum number of 100 samples must be taken to meet the approximations introduced in the algorithm.

4. Time synchronization algorithm

In this section, the algorithm for calculating the parameters of the synchronization model is described. As it has been mentioned, the network nodes integrate a microcontroller that keeps the local clock in an internal timer. This timer receives periodic impulses from a quartz crystal, which drifts from its nominal frequency due to the changing environmental conditions. Thus, the relative deviation that exists between the clocks of different nodes changes over time. Nevertheless, the relationship between the local clock of a node and the clock of the master can be assumed to be linear over short periods of time [21]. However, the parameters of the linear model change due to the different drift affecting each clock. Thus, the values of two clocks can be related by a linear model:

$$y(t) = \beta \cdot x(t) + \alpha \tag{7}$$

where $y(t)$ and $x(t)$ are the two clocks, t is the real time, β is the slope and α is the intercept. The parameters of this linear model change over time and must be recalculated.

The objective of the algorithm proposed in this work is to obtain a linear model to turn the local clocks of the nodes into a common global clock. This global clock is maintained at the master which broadcasts it with the transmission of the synchronization packets received by the rest of the nodes. The collection of timestamps for calculating the linear model only requires the transmission of one broadcast packet. So, at this stage, after the completion of the ToF estimation, the protocol implements the OWR transmission scheme with a given SP. To obtain the linear model, the timestamps of the global clock must be directly related to the timestamps captured in the internal timers of the microcontrollers, which maintain the local clocks of the nodes. Since these timestamps are captured only at the reception of a packet, they can only be linked directly to the samples of the master UWB clock. Therefore, the master UWB timer will serve as the global common clock of the network to which all the local clocks of the nodes will be synchronized. The relationship between the master UWB clock and the nodes local clocks following a linear model can be expressed as:

$$T^{M}_{UWB}(t) = \beta_k \cdot T^{k}_{MCU}(t) + \alpha_k \tag{8}$$

where $T^{M}_{UWB}(t)$ is the master UWB clock, $T^{k}_{MCU}(t)$ is the k node local clock, β_k is the slope and α_k the intercept for the node k. However, this equation does not take into account the delay of RF signal. Thus, this expression can be refined introducing the ToF of the RF signal:

$$T^M_{UWB}(t) = \beta_k \cdot (T^k_{MCU}(t) - \overline{ToF}_k) + \alpha_k \tag{9}$$

being $\overline{ToF}_k$ the mean estimate of the ToF of the RF signal transmitted from the master to the k node. With this model the relationship between the timestamps collected at the master UWB timer and the microcontrollers local clock can be expressed as:

$$T^M_{UWB}(i) = \beta_k \cdot (T^k_{MCU}(i) - \overline{ToF}_k) + \alpha_k + \eta_i \tag{10}$$

where $T^M_{UWB}(i)$ and $T^k_{MCU}(i)$ are the timestamps collected after the transmission of the $i-th$ synchronization round and η_i is the error associated to the capturing process. This last term represents the uncertainty that affects the acquisition of the timestamps. As this process is completed entirely by hardware elements, just at the detection of the SFD field of the packet received, it is not required the intervention of software components that can introduce random delays in the timestamping process. Nevertheless, there are still sources of small errors that affect the precision of the timestamps acquisition, such as the RF signal multi-path or the randomness of SFDLed pin activation. Thus, this term is considered as additive white Gaussian noise $\eta \sim N(0, \sigma^2)$, being σ its standard deviation.

After collecting a sequence of WS samples formed by pairs of timestamps $(T^M_{UWB}(i), T^k_{MCU}(i))$ a least squares algorithm can be applied to obtain the estimates $\hat{\beta_k}$ and $\hat{\alpha_k}$ of the linear model. The constant WS indicates the window size of the linear regression applied. The least squares method does not require that the samples follow a specific statistical distribution. With the arrival of a new packet from the master, the nodes recalculate the linear model applying the least squares algorithm using a rolling window of size WS over the last pairs of timestamps collected.

Once the estimates $\hat{\beta_k}$ and $\hat{\alpha_k}$ are calculated by a node, it can predict future values of the global common clock from its own local clock samples. Thus, nodes can convert their local clock samples T^k_{MCU} into global clock values $\hat{T}^M_{UWB,k}$ by applying the equation:

$$\hat{T}^M_{UWB,k} = \hat{\beta_k} \cdot (T^k_{MCU}) + \hat{\alpha_k} \tag{11}$$

5. Performance Analysis

This section shows the obtained results in the synchronization of real nodes employing the proposed algorithm. The platform utilized is based on the microcontroller ATSAMD21G18A from Microchip [36] and the UWB DW1000 transceiver from Qorvo [37]. Both components are connected using the SPI port and configured using the Arduino-dw1000 library [38]. As it was described in section 3, the acquisition of the timestamps in the microcontroller involves the activation of the SFDLed output pin of the UWB transceiver, which is connected to the capture input of a timer. This timer has been configured to be incremented by a clock signal with a frequency of 78.67 MHz. The setup for the experiment comprised four nodes that were deployed in an indoor corridor in the building of ETSE, University of Valencia. Fig. 3 shows an image of the nodes assembled. A more detailed schematic of the components and their interconnection can be found in [14].

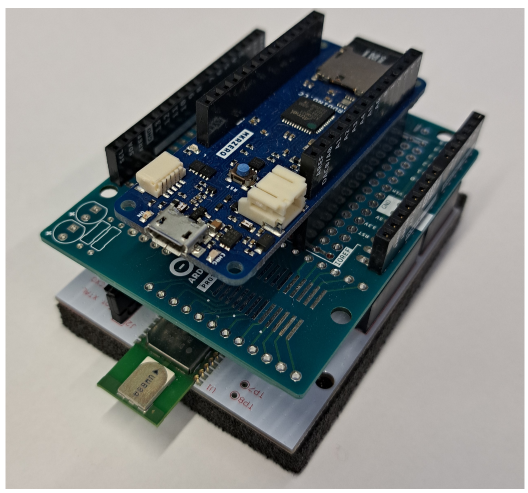

Figure 3: Image of the node assembled.

The placement of the nodes was selected to evaluate the impact of the ToF cancellation on the accuracy of the synchronization algorithm. To this end, the nodes of the network were deployed following the scheme depicted in Fig. 4. As it can be seen, there is a master node and two network nodes. Initially, the two network nodes were placed together at a position that was three meters away from the master. After the evaluation of this setup, one of the network nodes was moved away, changing gradually the distance between them from 0 to 25 meters in steps of 5 meters. During the conducted experiments the master transmitted synchronization packets with a period of 200 ms. These packets were received at the two network nodes with a distinct ToF delay due to the different distances from the master to them. The evaluation of the algorithm required the inclusion of a fourth node, which retransmits the packets of the master after a fixed delay. This fourth node, denoted as repeater in Fig. 4, was placed in all the experiments in the middle point of the line connecting the two network nodes. The inclusion of the repeater node facilitates the evaluation of the synchronization algorithm since the packets sent from the repeater are received at both network nodes simultaneously. Thus, the timestamps captured at both network nodes with these packets should provide the same global time at both nodes. The repeater was programmed to retransmit the synchronization packets from the master after 100 ms to avoid collisions. The goals of the experiments were focused on testing the validity of the ToF estimates and the accuracy of the synchronization algorithm.

Figure 4: Deployment of the four nodes employed in the evaluation. Node 2 was moved away from node 1 in steps of 5 meters covering a total distance from 0 to 25 meters. Repeater was always placed at the midpoint between the two nodes.

5.1. Evaluation of the ToF estimation

In this section, the evaluation of the ToF estimation algorithm is presented. As it was detailed in section 3.2, the nodes have to complete two steps. The first one covers the time needed by the UWB transceiver oscillator to stabilize. The duration of this period can be determined checking the internal temperature of the UWB transceiver. As it has been experimentally measured by testing the operation of several DW1000 transceivers, the internal temperature rises up very rapidly during the warm-up and remains stable after this period. The end of the warm-up interval is detected checking the internal temperature of the UWB transceiver. The algorithm acquires the internal temperature of the UWB which is sampled every SP. The vector with the temperature values is filtered with a triangular moving average filter to eliminate the sampling noise. After that, an approximation of the derivative of the temperature is calculated by taking the difference between consecutive samples and dividing by the SP. The end of the warm-up period is detected when the derivative is zero.

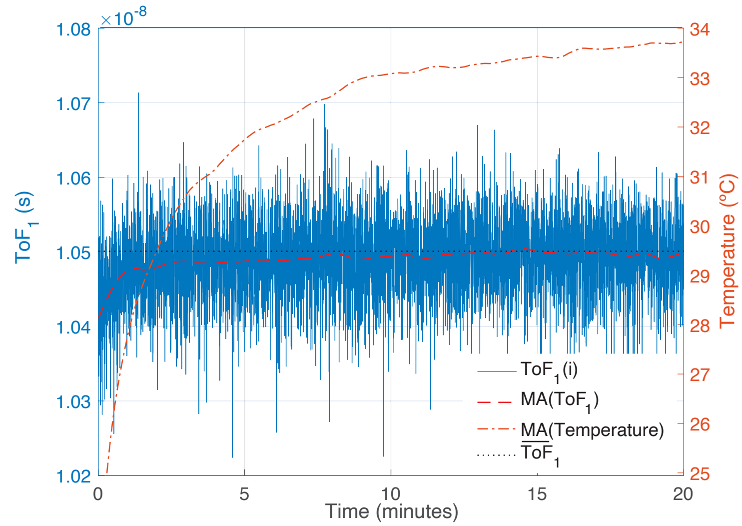

Figure 5: ToF_1 and moving average filtered internal temperature (MA(Temperature)) of node 1 placed 3 meters away from the master. ToF values include: a) $ToF_1(i)$, b) moving average filtered ToF_1 (MA(ToF_1)) and c) $\overline{ToF}_1$.

The experimental data collected to test the validity of the proposed warm-up period detection procedure corresponds to the case where the two network nodes were placed at distances of 3 meters and 8 meters from the master. The experiment lasted 6 hours in a corridor with a constant temperature of 24 ºC. The packet exchange implemented to calculate the ToF followed the scheme presented in Fig. 2 and the internal temperature of the nodes was acquired with a sampling period of 200 ms. Fig. 5 and 6 present for the each node the instantaneous ToF, the filtered ToF obtained applying a triangular moving average filter and the mean of the ToF value computed considering all the samples taken during the test. Additionally, the moving average filtered internal temperature of the nodes is also plotted. Considering these graphs, it can be seen that the temperature and ToF estimate increase rapidly at the

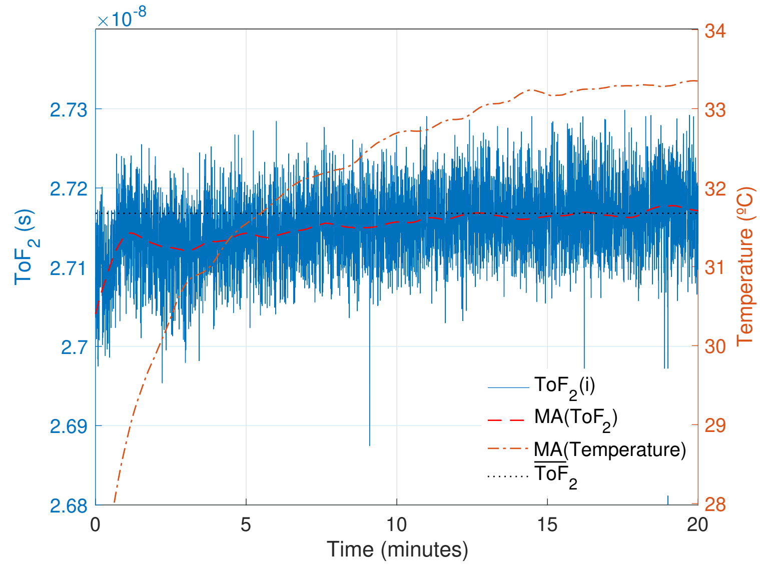

Figure 6: ToF_2 and moving average filtered internal temperature (MA(Temperature)) of node 2 placed 8 meters away from the master. ToF values include: a) $ToF_2(i)$, b) moving average filtered ToF_2 (MA(ToF_2)) and c) $\overline{ToF}_2$.

beginning, but after a few minutes both parameters remain stable. Therefore, the conclusion to be drawn from these graphs is that there is an significant correlation between them. Thus, the lower fluctuation of the temperature will lead to a more precise determination of the end of warm-up stage. The detection of this instant can be performed searching the first sample where the approximation of the temperature derivative is zero. As it is depicted in Fig. 7, for both nodes this occurs after about 10 minutes of operation.

Once the end of the warm-up is found the second step of the ToF estimate can start. This process involves the calculation of the ToF and the CIA as they were introduced in section 3.2. The goal is to determine the minimum number of samples that can provide a representative estimate of the ToF. These samples are posterior to the end of the warm-up period and are used to calculate the estimate of the ToF using Eq. 2. This second step finishes when the CIA calculated with Eq. 6 is below a certain threshold which is sufficient to guarantee the precision required in the ToF estimation. In this evaluation, the threshold for the CIA was selected as 1 click of the UWB timer. When the CIA is below this threshold the ToF estimation stage ends and the sample mean calculated with Eq. 2 becomes the estimate that will

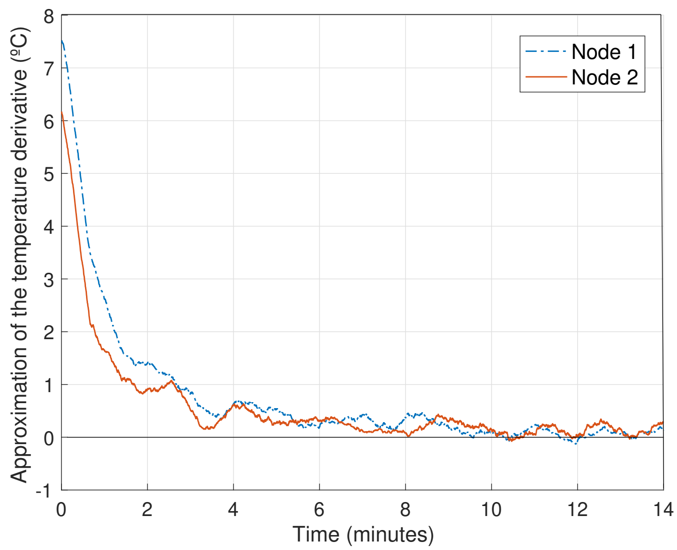

Figure 7: Approximation of the internal temperature derivatives of the UWB transceivers.

be used during the clock synchronization. Fig. 8 and 9 show the CIA for both nodes and the difference between the sample mean of ToF with the number of samples taken to that point and the mean considering all the samples collected during the 6 hours of the experiment. The point were the algorithm collected the first 100 samples has been marked with a vertical line. This is the minimum number of samples that the algorithm has to take due to the approximations considered in Eq. 6. In this experiment, since the CIA for both nodes is lower than 1 UWB timer click after the first 100 samples, that is the point where the ToF estimation ends. As it can be seen, the error introduced taking this estimate with $n = 100$ is practically negligible compared with the estimate obtained considering all the samples collected during the experiment.

5.2. Evaluation of the clock synchronization

As it has been previously mentioned, the objective of the proposed algorithm is the attainment of a linear model capable of turning the local time of the nodes into the global common clock. After the estimation of the ToF, the next experiments evaluate the accuracy of the synchronization algorithm introduced in section 4. The deployment of the nodes was carried out considering several distances, as it is represented in Fig. 4, and the SP was 200 ms.

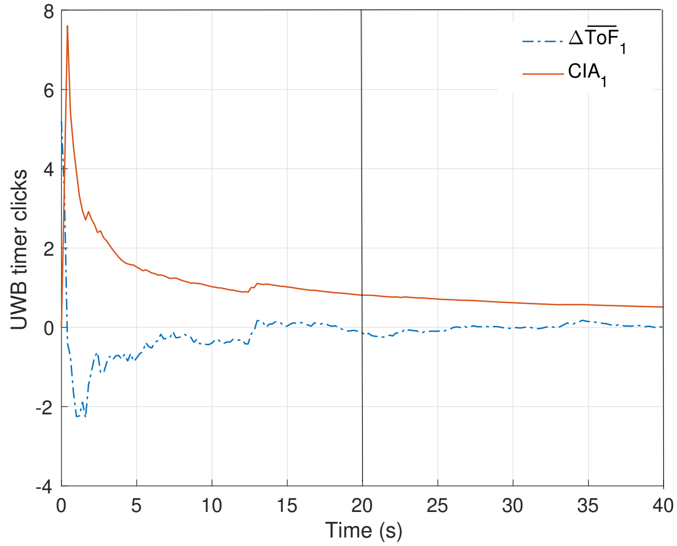

Figure 8: Values of CIA_1 and $\Delta\overline{ToF}_1$ for node 1 placed 3 meters away from the master. $\Delta\overline{ToF}_1$ is the difference between the sample mean $\overline{ToF}_1$ with the number of samples obtained to that instant and the mean considering all the samples collected in the experiment. Vertical line indicates the point with n = 100 samples. The UWB timer click for the DW1000 transceiver is 15.6 ps.

The estimation of the linear regression parameters relied on a rolling window containing the last WS samples. Every sample i was formed by a couple of timestamps one from the master UWB timer, which represents the global common clock, and the microcontroller timer of the k node, considered as its local clock, $(T^M_{UWB}(i), T^k_{MCU}(i))$. The linear model applied follows the expression in Eq. 9 and the estimation of $\hat{\beta}_k$ and $\hat{\alpha_k}$ is updated with every new synchronization packet from the master. The WS value has been fixed to 3, as it was observed in previous experiments [14] that achieves the best results. The experiments lasted 6 hours for each placement and the temperature of the corridor was stable with values between 24ºC and 25ºC all the time.

The assessment of the synchronization accuracy is based on the calculation of the difference of the predicted global times provided by the two network nodes for the timestamps captured after the repeater transmissions. Since the repeater is equidistant from the two network nodes, they receive its packets simultaneously. The timestamps obtained after the arrival of the repeater packets are captured at the same instant on both nodes. Therefore,

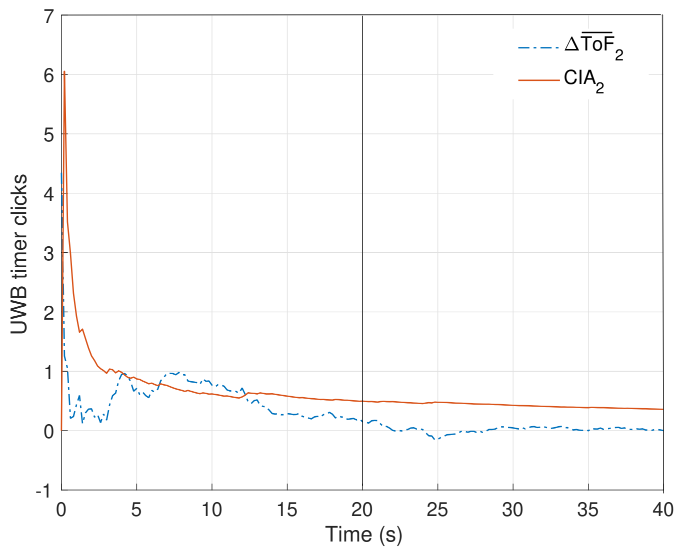

Figure 9: Values of CIA_2 and $\Delta\overline{ToF}_2$ for node 2 placed 8 meters away from the master. $\Delta\overline{ToF}_2$ is the difference between the sample mean of ToF_2 with the number of samples taken to that point and the mean considering all the samples collected in the experiment. Vertical line indicates the point with n = 100 samples.

they should provide the same global time values for these timestamps. The predicted global times at the two networks nodes are calculated as:

$$\hat{T}^M_{UWB,1}(i) = \hat{\beta}_1 \cdot T^1_{MCU}(i) + \hat{\alpha}_1 \tag{12}$$

$$\hat{T}^M_{UWB,2}(i) = \hat{\beta}_2 \cdot T^2_{MCU}(i) + \hat{\alpha}_2 \tag{13}$$

where $\hat{T}^M_{UWB,1}(i)$ and $\hat{T}^M_{UWB,2}(i)$ are the global time values calculated at the two nodes after capturing the local clocks timestamps $T^1_{MCU}(i)$ and $T^2_{MCU}(i)$, when the repeater has transmitted the $i-th$ packet. With these values the absolute difference between the global time predicted by the two nodes can be calculated:

$$|e_r(i)| = |\hat{T}^M_{UWB,1}(i) - \hat{T}^M_{UWB,2}(i)| \tag{14}$$

The evaluation of the synchronization accuracy is carried out calculating the average of the sequence of $|e_r(i)|$ values obtained after each repeater

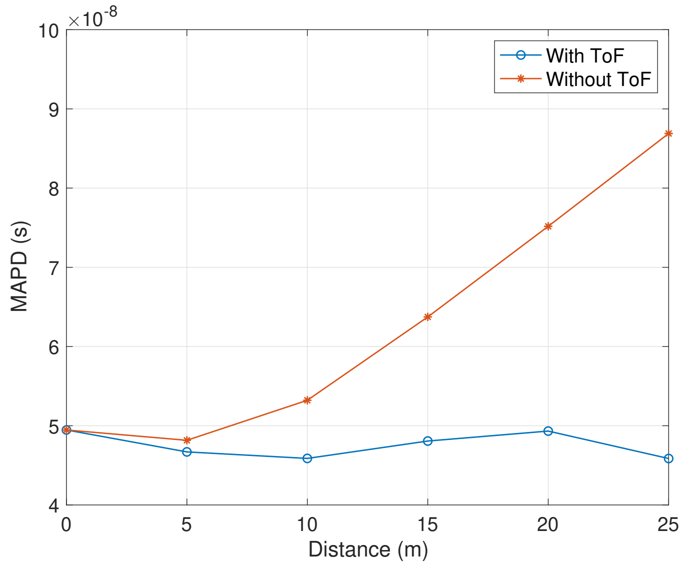

Figure 10: Mean absolute prediction difference (MAPD) of the global clock at two network nodes using the nodes local clocks. The distance between the nodes changes from 0 to 25 meters. Two linear models with and without ToF were considered.

transmission. The resulting term is denoted as the Mean Absolute Prediction Difference (MAPD).

The evaluation has also considered the case in which the linear regression is calculated without including the ToF between the master and the node. In this case, the linear model follows the expression given in Eq. 8 and the regression is performed using only the timestamps gathered ($T^M_{UWB}(i)$, $T^k_{MCU}(i)$), without the ToF estimate. The comparison of the results obtained with both approaches determines the improvement achieved with the inclusion of the ToF delay in the model. The results shown in Fig. 10 comprise various positions for the node 2, which is moved away to change the distance between the two network nodes in an interval from 0 to 25 meters. As it can be noticed, for distances greater than 5 meters the model without ToF produces higher MAPD values, being this difference more pronounced as the separation between the nodes increases. Finally, with a separation of 25 meters, the MAPD value for the regression without ToF doubles the value achieved with the model that considers this delay. The difference in synchronization accuracy at this last position coincides with the ToF delay to cover the 25 meters. Table 2 presents the statistics of the results achieved

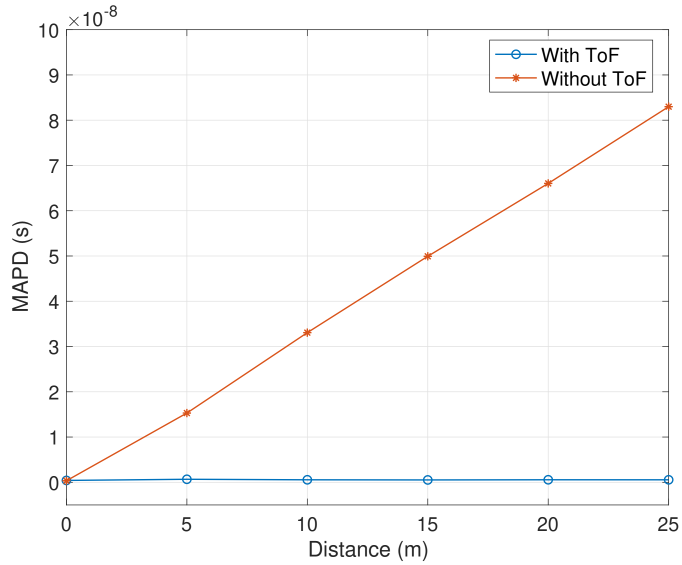

Figure 11: Mean absolute prediction difference (MAPD) of the global clock at two network nodes using the nodes UWB clocks. The distance between the nodes changes from 0 to 25 meters. Two linear models with and without ToF were considered.

for the synchronization at the MCU. All the files containing the timestamps collected in these experiments are available online[1].

Although the main goal of this work is the synchronization of the local clocks of the nodes with the UWB clock of the master, the same experiments were performed using the UWB clocks of the nodes. The objective of this evaluation is to corroborate that the same improvement can also be achieved in the synchronization between the timers of the UWB transceivers. The procedure followed in this experiment was the same but changing the timestamps coming from the microcontroller timers to the UWB timers of the nodes. Thus, the least squares algorithm is computed with the terms $(T^M_{UWB}(i), T^k_{UWB}(i))$, where $T^k_{UWB}(i)$ is the timestamp captured at the UWB transceiver of the k node after the transmission of the i synchronization packet from the master. Fig. 11 shows the results of this experiment and coincides with the previous case in the divergence of the estimate of the global clock with the distance. Thus, when the nodes are 25 m apart, the MAPD

[1]https://www.uv.es/jjperezs/com/Data.zip

Table 2: Statistics of the synchronization results achieved at the MCU for all the distances considered when the ToF is cancelled (With ToF) and not cancelled (Without ToF). Values shown are: Mean Absolute Prediction Difference (MAPD), Standard Deviation of the Absolute Prediction Difference (STD) and 99th Percentile of the Absolute Prediction Difference (P99).

	With ToF						Without ToF					
Distance	**0m**	**5m**	**10m**	**15m**	**20m**	**25m**	**0m**	**5m**	**10m**	**15m**	**20m**	**25m**
MAPD (ns)	49.4	46.6	45.8	48.0	49.3	45.8	49.4	48.3	53.2	63.7	74.0	86.8
STD (ns)	37.5	35.3	34.8	36.4	37.6	33.3	37.5	36.7	39.8	45.4	49.8	40.6
P99 (ns)	160.7	152.4	148.7	156.7	161.3	155.9	157.1	156.9	168.1	190.1	192.3	188.2

Table 3: Statistics of the synchronization results achieved at the UWB for all the distances considered when the ToF is cancelled (With ToF) and not cancelled (Without ToF). Values shown are: Mean Absolute Prediction Difference (MAPD), Standard Deviation of the Absolute Prediction Difference (STD) and 99th Percentile of the Absolute Prediction Difference (P99).

	With ToF						Without ToF					
Distance	**0m**	**5m**	**10m**	**15m**	**20m**	**25m**	**0m**	**5m**	**10m**	**15m**	**20m**	**25m**
MAPD (ns)	0.41	0.67	0.55	0.52	0.56	0.55	0.34	16.0	33.0	49.9	66.3	82.9
STD (ns)	0.28	0.31	0.35	0.30	0.35	0.39	0.24	0.35	0.48	0.35	0.48	0.48
P99 (ns)	1.20	1.42	1.52	1.30	1.51	1.63	1.03	16.9	34.2	50.7	67.5	84.1

is less than 1 ns for the model with ToF and rises to 83 ns for the model without ToF. This error comes from the different ToF of the RF signal and coincides with the time required for the signal to cover this distance. Table 3 presents the statistics of the results achieved for the synchronization at the UWB.

5.3. Comparison of the proposed algorithm with the related work

The main contribution of this work is the proposal of a new algorithm that using the DW1000 transceiver can synchronize accurately the timers inside the micrcontrollers of the network nodes. As it can be seen in Table 2, with timers operating with 78.67 MHz clocks, the mean error committed is tens of nanoseconds, regardless of the distance between the master and slave nodes. These results can be compared with those provided by protocols found in the literature and summarized in Table 1 whose objective is the synchronization of a processor timer. As it can be seen, the proposed algorithm achieves a synchronization accuracy of the processor timer much better than the cited works and outperforms these references. It should also be noted that the network nodes have been designed using only low-cost components. Thus, the computational complexity of the synchronization algorithm has been kept low to facilitate its implementation in devices with very limited resources.

6. Conclusions

The work presented in this paper proposes a new synchronization algorithm for WSN based on UWB communications. This technology allows the acquisition of precise timestamps after the arrival of the synchronization packets at both the UWB transceiver and the microcontroller. The algorithm allows the synchronization of the local clocks of the nodes to the global common clock maintained in the UWB transceiver of the master. This synchronization permits the timestamping of local events and the data sampling process with a common global clock shared by all the network nodes. Thus, the protocol enables the implementation of collaborative tasks that need a precise time synchronization and the temporal alignment of the data coming from different nodes in applications requiring high sampling rates.

Since the ToF of the radio signal coming from the master is different for every node, the linear model obtained without considering this delay do not provide correct results when the distances from the master to the nodes is above 10 meters. Thus, the algorithm includes an initial stage where the nodes estimate this delay to cancel it during the synchronization process. To avoid the initial fluctuations of the ToF during the warm-up period of the UWB transceiver, the algorithm samples the internal temperature of the UWB transceiver and rules out the ToF values obtained before its stabilization. Once the temperature of the UWB transceiver becomes stable, the algorithm calculates the confidence interval of the sample mean to obtain an accurate estimation of the ToF with a reduced number of samples. During this stage, the master exchanges a sequence of three packets with each node to collect the six timestamps in the UWB transceivers required to apply the ADS-TWR algorithm. Since the transmission of the third packet from the master can be reused as the first packet for the following node, only two packets are actually needed for each node, with the exception of the last one that needs three packets. The presented results show that an accurate ToF estimate can be achieved with one hundred samples.

Once the ToF dependence is removed, the linear model relating the clocks of the master and the nodes provides accurate results regardless of their positions. During the synchronization stage the packet transmission scheme changes to the OWR, because only one broadcast packet is enough to synchronize all the nodes in the radio range of the master. The linear model is obtained applying the well-known least squares method over a rolling window comprising the last three pairs of timestamps captured at the master UWB transceiver and the node local clock. When local clocks of the nodes are implemented with an internal timer of the microcontrollers, the difference between the global clock predicted in two nodes is in order of tens of nanoseconds. In the case of using the UWB timers to establish the local clocks of the nodes, this difference becomes less than one nanosecond.

7. Acknowledgment

This work was supported by grant PID2020-113785RB-100 funded by MICIU/AEI/10.13039/501100011033. Grant PID2021-126823OBI00 funded by MICIU/AEI/10.13039/501100011033. Grant TED2021-131040B-C33 funded by MICIU/AEI/10.13039/501100011033 and the European Union Next Generation EU/PRTR. Also, grants CIAICO/2022/179, CIAEST/2022/50 and CIAEST/2022/64 funded by Generalitat Valenciana, Spain.

8. References

References

[1] M. Delamo, S. Felici-Castell, J. J. Pérez-Solano, A. Foster, Designing an open source maintenance-free Environmental Monitoring Application for Wireless Sensor Networks, Journal of Systems and Software 103 (2015) 238 – 247. doi:10.1016/j.jss.2015.02.013.

[2] R. Sánchez-Corcuera, A. Nuñez-Marcos, J. Sesma-Solance, A. Bilbao-Jayo, R. Mulero, U. Zulaika, G. Azkune, A. Almeida, Smart cities survey: Technologies, application domains and challenges for the cities of the future, International Journal of Distributed Sensor Networks 15 (6) (2019) 1550147719853984.

[3] G. S. Tewolde, Sensor and network technology for intelligent transportation systems, in: 2012 IEEE International Conference on Electro/Information Technology, 2012, pp. 1–7. doi:10.1109/EIT.2012.6220735.

[4] Y. Zhang, L. Sun, H. Song, X. Cao, Ubiquitous wsn for healthcare: Recent advances and future prospects, IEEE Internet of Things Journal 1 (4) (2014) 311–318.

[5] V. C. Gungor, B. Lu, G. P. Hancke, Opportunities and challenges of wireless sensor networks in smart grid, IEEE transactions on industrial electronics 57 (10) (2010) 3557–3564.

[6] M. Majid, S. Habib, A. R. Javed, M. Rizwan, G. Srivastava, T. R. Gadekallu, J. C.-W. Lin, Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review, Sensors 22 (6) (2022) 2087.

[7] K. F. Hasan, Y. Feng, Y.-C. Tian, Gnss time synchronization in vehicular ad-hoc networks: Benefits and feasibility, IEEE Transactions on Intelligent Transportation Systems 19 (12) (2018) 3915–3924. doi:10.1109/TITS.2017.2789291.

[8] T. Schmid, R. Shea, Z. Charbiwala, J. Friedman, M. B. Srivastava, Y. H. Cho, On the interaction of clocks, power, and synchronization in duty-cycled embedded sensor nodes, ACM Trans. Sen. Netw. 7 (3) (2010) 24:1–24:19. doi:10.1145/1807048.1807053.

[9] Z. Zhong, P. Chen, T. He, On-demand time synchronization with predictable accuracy, in: INFOCOM, 2011 Proceedings IEEE, 2011, pp. 2480–2488. doi:10.1109/INFCOM.2011.5935071.

[10] I. S. Association, Ieee std 802.15. 4-2011, ieee standard for local and metropolitan area networks—part 15.4: Low-rate wireless personal area networks (lr-wpans) (2011).

[11] L. Zwirello, T. Schipper, M. Harter, T. Zwick, Uwb localization system for indoor applications: Concept, realization and analysis, Journal of Electrical and Computer Engineering 2012 (2012) 1–11.

[12] I. S. Association, Ieee 802.15. 4-2015-ieee standard for low-rate wireless networks (2015).

[13] I. S. Association, Ieee standard for low-rate wireless networks–amendment 1: Enhanced ultra wideband (uwb) physical layers (phys) and associated ranging techniques (ieee std 802.15. 4z) (2020).

[14] J. J. Pérez-Solano, S. Felici-Castell, A. Soriano-Asensi, J. Segura-Garcia, Time synchronization enhancements in wireless networks with ultra wide band communications, Computer Communications 186 (2022) 80–89. doi:10.1016/j.comcom.2022.01.012.

[15] Qorvo, DWM1001 System Overview And Performance, `https://www.qorvo.com/products/d/da007974`, accessed: 17/02/2023 (2017).

[16] S. Ganeriwal, R. Kumar, M. B. Srivastava, Timing-sync protocol for sensor networks, in: Proceedings of the 1st international conference on Embedded networked sensor systems, SenSys '03, ACM, New York, NY, USA, 2003, pp. 138–149. doi:10.1145/958491.958508.

[17] J. Elson, L. Girod, D. Estrin, Fine-grained network time synchronization using reference broadcasts, SIGOPS Oper. Syst. Rev. 36 (SI) (2002) 147–163. doi:10.1145/844128.844143.

[18] M. Maróti, B. Kusy, G. Simon, A. Lédeczi, The flooding time synchronization protocol, in: Proceedings of the 2nd international conference on Embedded networked sensor systems, SenSys '04, ACM, New York, NY, USA, 2004, pp. 39–49. doi:10.1145/1031495.1031501.

[19] S. Yoon, C. Veerarittiphan, M. L. Sichitiu, Tiny-sync: Tight time synchronization for wireless sensor networks, ACM Trans. Sen. Netw. 3 (2) (Jun. 2007). doi:10.1145/1240226.1240228.

[20] S. Ganeriwal, I. Tsigkogiannis, H. Shim, V. Tsiatsis, M. Srivastava, D. Ganesan, Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks, Networking, IEEE/ACM Transactions on 17 (3) (2009) 843–856. doi:10.1109/TNET.2008.2001953.

[21] J. J. Pérez-Solano, S. Felici-Castell, Adaptive time window linear regression algorithm for accurate time synchronization in wireless sensor networks, Ad Hoc Networks 24 (2015) 92–108. doi:10.1016/j.adhoc.2014.08.002.

[22] P. Sommer, R. Wattenhofer, Gradient clock synchronization in wireless sensor networks, in: 2009 International Conference on Information Processing in Sensor Networks, 2009, pp. 37–48.

[23] D. Neirynck, E. Luk, M. McLaughlin, An alternative double-sided two-way ranging method, in: 2016 13th Workshop on Positioning, Navigation and Communications (WPNC), 2016, pp. 1–4. doi:10.1109/WPNC.2016.7822844.

[24] V. Djaja-Josko, J. Kolakowski, A new method for wireless synchronization and tdoa error reduction in uwb positioning system, in: 2016 21st International Conference on Microwave, Radar and Wireless Communications (MIKON), 2016, pp. 1–4. doi:10.1109/MIKON.2016.7492077.

[25] V. Djaja-Josko, Novel method for the wireless synchronization of the anchors in the uwb localization system utilizing two reference nodes, in: 2020 23rd International Microwave and Radar Conference (MIKON), 2020, pp. 69–73. doi:10.23919/MIKON48703.2020.9253932.

[26] C. Xiaomao, F. Yiwei, L. Chunfei, Improved master-slave time synchronization algorithm based on tdoa positioning, in: Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering, EITCE 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 210–215. doi:10.1145/3443467.3443755.

[27] V. Navrátil, J. Krška, F. Vejražka, Concurrent bidirectional tdoa positioning in uwb network with free-running clocks, IEEE Transactions on Aerospace and Electronic Systems 58 (5) (2022) 4434–4450. doi:10.1109/TAES.2022.3161895.

[28] Q. Shi, X. Cui, S. Zhao, S. Xu, M. Lu, Blas: Broadcast relative localization and clock synchronization for dynamic dense multiagent systems, IEEE Transactions on Aerospace and Electronic Systems 56 (5) (2020) 3822–3839. doi:10.1109/TAES.2020.2979640.

[29] R. Zou, N. Wu, Z. Qu, J. Chen, Design and implementation of a high-precision wireless clock synchronization system based on uwb, in: 2022 4th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), 2022, pp. 1094–1099. doi:10.1109/ICMSP55950.2022.9859065.

[30] N. M. Senevirathna, O. De Silva, G. K. I. Mann, R. G. Gosine, Asymptotic gradient clock synchronization in wireless sensor networks for uwb localization, IEEE Sensors Journal 22 (24) (2022) 24578–24592. doi:10.1109/JSEN.2022.3213696.

[31] F. Bonafini, P. Ferrari, A. Flammini, S. Rinaldi, E. Sisinni, Exploiting time synchronization as side effect in uwb real-time localization devices, in: 2018 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), 2018, pp. 1–6.

[32] A. Dongare, P. Lazik, N. Rajagopal, A. Rowe, Pulsar: A wireless propagation-aware clock synchronization platform, in: 2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2017, pp. 283–292.

[33] B. Xue, Z. Li, P. Lei, Y. Wang, X. Zou, Wicsync: a wireless multi-node clock synchronization solution based on optimized uwb two-way clock synchronization protocol, Measurement (2021). doi:10.1016/j.measurement.2021.109760.

[34] J. Sidorenko, V. Schatz, N. Scherer-Negenborn, M. Arens, U. Hugentobler, Decawave ultra-wideband warm-up error correction, IEEE Transactions on Aerospace and Electronic Systems 57 (1) (2021) 751–760. doi:10.1109/TAES.2020.3015323.

[35] Qorvo, APS011 Application Note. Sources of Error in DW1000 Based Two-Way Ranging (TWR) Schemes, https://www.qorvo.com/products/d/da008446, accessed: 17/02/2023 (2015).

[36] Microchip, ATsamd21g18 microcontroller, https://www.microchip.com/en-us/product/ATsamd21g18, accessed: 10/06/2023.

[37] Qorvo, DW1000 UWB transceiver, https://www.qorvo.com/products/p/DW1000, accessed: 10/06/2023.

[38] Thomas Trojer, Github: Library that offers basic functionality to use Decawave's DW1000 chips/modules with Arduino, https://github.com/thotro/arduino-dw1000, accessed: 28/07/2021 (2018).