Ultra Wideband-based wireless synchronization of IEEE 1588 clocks

Gergely Hollósi, and István Moldován

Abstract—**Time-Sensitive Networking (TSN) requires clock synchronization superior to the well-known Network Time Protocol (NTP). The IEEE 802.1AS-2020 used for synchronization in TSN networks is based on the IEEE 1588-2019 standard (also known as Precision Time Protocol, PTP) defines methods and tools to perform sub-microsecond time synchronization over vari- ous communication channels. However, the IEEE 1588 implementation is commonly used with wired communication protocols, although there are use cases that could gain an advantage from a wireless solution. This paper investigates the possibility of PTP clock synchronization through wireless Ultra Wideband (UWB) communication. UWB excels where other wireless technologies are lacking: it provides high accuracy timestamping even if multipath propagation is present. The method is evaluated using commercial, well-accessible cheap hardware, resulting in the order of 10-nanosecond accuracy. The paper also highlights the main error components and requirements for improving wireless PTP synchronization.**

Index Terms—**IEEE 1588, UWB, PTP, TSN, clock synchronization.**

I. Introduction

Precise and accurate synchronization is gaining more and more attention in actual industrial internet-of-things (IIoT) and a couple of different use-cases, including real-time applications like nuclear fusion control, mobile communication, substation automation, and, modern manufacturing plant [1], [2]. The well-known Network Time Protocol (NTP) was designed to perform time transfer in IT services, and its accuracy is in the millisecond range. However, new services require sub-millisecond accuracy, e.g., microsecond or, in some cases, nanosecond accuracy [2], [3]. The IEEE 1588-2019 standard [4] (commonly known as the Precision Time Protocol – PTP) addresses the topic of sub-microsecond accurate clock synchronization over a communication network. In most cases, these accurate clock synchronization solutions are implemented on a wired network with PTP. There are many wireless implementations based on GPS synchronization, which are also capable of sub-microsecond clock synchronization, but they have significant limitations for indoor applications. Therefore, wired PTP solutions are favored for indoor clock synchronization applications. Still, there are advantages of a wireless clock synchronization solution in an indoor environment for validation and wireless master clock purposes. The industrial standard to validate clock synchronization accuracy in PTP networks is based on PPS (Pulse Per Second) signals. During the validation, the PPS signal of the clock of the synchronized device can be compared to the PPS signal of a reference clock, from which the synchronization accuracy can be measured. For a grand master reference, GPS is widely applied, however, it is prevalent that GPS cannot be an option in an indoor environment, while the wired network's PPS signal transmission requires additional cabling and network configuration. Such challenges open the path for wireless PTP validation where there is no need for additional cabling and offer a flexible solution. Besides validation, wireless PTP can be utilized as a master clock to eliminate non-transparent PTP devices from the PTP network, also there is no need for cabling; existing network devices do not have to be replaced and it can be significantly cheaper compared to wired PTP system implementations.

Present paper proposes a method to synchronize local PTP clocks by means of wireless UWB communication. While synchronizing UWB clocks is a widely studied topic, using UWB communication for synchronization of PTP enabled clocks of Ethernet devices is a rarely researched area. The main advantage of the presented method is that the PTP clocks then readily can be used as master clocks in a PTP network.

The paper is structured as follows: The related works present the core concepts of IEEE 1588 including its application areas, wired and wireless PTP implementations and the introduction of the UWB technology. Besides the general methodology, there are many specific implementation challenges, which are presented in a separate section. The evaluation section shows the results of the UWB-based PTP clocks' synchronization errors and describes the origins of these errors.

II. Related works

The PTP standard provides tools to synchronize a so-called slave clock to a master clock. Here, only the end-to-end solution is presented, as follows. The standard uses two essential messages to achieve end-to-end synchronization (see Fig. 1), a *Sync* message and a *Delay_Req* message. The *Sync* message is sent by the master, and the master notes the transmission (TX) timestamp (t_1) of the message. The slave device receives the message and saves the receive (RX) timestamp (t_2) of the message. In case of one-step implementation, the TX timestamp is embedded in the *Sync* message, while for two-step implementation an auxiliary message (*Follow_Up*) is used to send the TX timestamp to the slave. While the clock drift (the frequency difference) can be derived from the

The authors are with the Department of Telecommunications and Media Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Budapest, Hungary.
(e-mail: hollosi.gergely@vik.bme.hu)

DOI: 10.36244/ICJ.2023.2.4

Sync message, an additional *Delay_Req* message is required to estimate the clock offset. The *Delay_Req* is sent by the slave at t_3 and received by master at t_4, which is sent back to the slave in an auxiliary message *Delay_Resp*. From the recorded timestamps, the clock offset and clock drift can be calculated.

To achieve micro- or nanosecond accuracy, software-only solutions are insufficient, thus hardware-aided methods are preferred. Hardware-aided methods make it possible to precisely record the TX and RX timestamps for packets transmitted to or received from the network; however, the synchronization algorithm can be implemented in software. Moreover, there is the White Rabbit project – included in IEEE 1588-2019 – where sub-nanosecond accuracy can also be achieved on a specialized network. White Rabbit is a project delivering to create an Ethernet-based network with low-latency, deterministic packet delivery and network-wide, transparent, high-accuracy timing distribution. The White Rabbit Network is based on Ethernet (IEEE 802.3), Synchronous Ethernet (SyncE) and PTP standards. The measured PTP performance demonstrates sub-nanosecond accuracy over a 5km fiber optic link with a precision below 10 ps [3].

Accurate clock synchronization and PTP's main application area is in the industrial domain, especially in Time-Sensitive Networking (TSN). TSN is a set of standards under development by the Time-Sensitive Networking task group of the IEEE 802.1 working group. Time-Sensitive Networking Task Group specifies TSN functionalities as a set of standards that provide deterministic services through IEEE 802 networks to enable bounded low packet loss, guaranteed packet delivery, bounded low latency and low packet delay variation. TSN targets a variety of particular aspects, among them timing and synchronization aspects. Synchronization is an essential building block to meet these requirements, and IEEE 802.1AS-2020 standard [5] defines a PTP profile to use in TSN networks.

Even though wired solutions are the main direction of PTP implementations, there are existing solutions and research in the wireless PTP domain. Óscar et al. [1] proposes a timestamping method for time synchronization over WLAN standard conditions. The paper presents several simulations using the IEEE 802.11n physical layer and four wireless time-dispersive and time-variant channel models. They have also performed validation experiments using an 802.11g modem implemented over a high-performance Software Defined Radio hardware platform. Inaki et al. [6] presents an IEEE 802.1AS clock synchronization performance evaluation of an integrated wired-wireless TSN architecture. Wireless TSN is expected to be integrated with Ethernet TSN to create large-scale wired–wireless TSN networks. The paper presents two hardware architectures to enable clock synchronization distribution among the network domains. Another Wireless PTP implementation is presented in [7]. This paper evaluates the performance of the over-the-air time synchronization mechanism, which has been proposed in 3GPP Release 16. The paper analyses the accuracy of time synchronization through the boundary clock approach in the presence of clock drift and different air-interface timing errors related to reference time indication. The paper also investigates the frequency and scalability aspects of over-the-air time synchronization. The performance evaluation reveals the conditions required for accuracy of 1μs or below in TSN time synchronization.

In case of IEEE 802.11 (commonly known as WiFi) the software based PTP solutions are capable of reaching time synchronization error in the order of 10 microseconds using wireless links [8], [9]. However, more precise synchronization can be achieved using the WiFi receiver information. Since IEEE 802.11-2006 standard, the protocol supports the fine timing measurement (FTM) for ranging purposes, which can be used to precise time-of-flight measurement [10]. The IEEE 802.1AS standard [5] also mentions the WiFi as the possibly media-dependent layer, however, available implementations are lagging behind.

There are a couple of explicit hardware solutions or circuits to help the implementation of the PTP protocol for Ethernet-based (IEEE 802.3) networks. However, wireless products – while theoretically capable of – do not provide timestamping features or hide the information which is needed to calculate accurate RX and TX timestamps. Furthermore, the propagation characteristics of the radio channel – especially in non-line-of-sight situations – generates jitter in receive timestamps [11]. Fortunately, Ultra Wideband (UWB) technology is perfectly suitable for timestamp reception and transmission of packets since indoor localization requires sub-nanosecond accuracy to provide location in centimeter precision. UWB is applied in various use-cases [12]–[14], but its main application area is indoor, centimeter-based localization using ranging techniques. UWB technology is applied also in time synchronization and synthonization use-cases [15]–[17], mainly for localization purposes (e.g. in time-difference-of-arrival systems). Marcelo et al. [18] achieves 5-ns RMS results in a UWB network, while the Wicsync solution reports errors below 3 nanoseconds [19]. However, all the solutions synchronizes the clocks in the UWB clock domain which avoids the need for clock domain change. UWB applies very short pulses, which help to estimate the channel impulse response (CIR) to extract the first component arrived, hence providing precise timestamps for the first (and hopefully the line-of-sight) component. The IEEE 802.14.5 standard [20] specifies the ranging counter as the clock for timestamping events, which has a resolution of around 15 picoseconds; however, the timestamping accuracy is in the order of nanoseconds.

III. Methods

Our goal is to design a device pair capable of providing accurate synchronization using UWB communication. There are two target use-cases as shown in Fig. 2:

- wireless synchronization between master and slave PTP devices and
- wireless PPS signal transmitter.

The first one aims to provide means to synchronize a local PTP master clock to a remote grand master clock using UWB communication, which can be applied in situations e.g. where

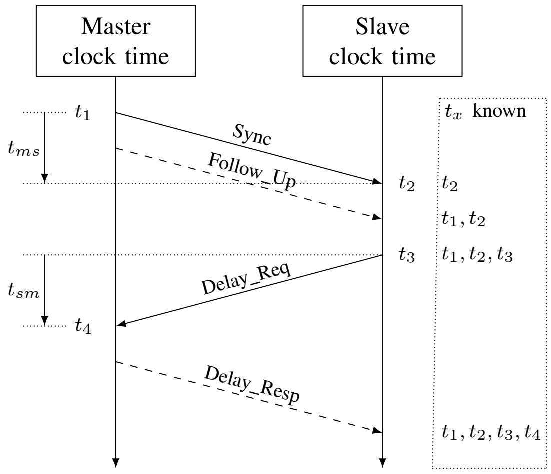
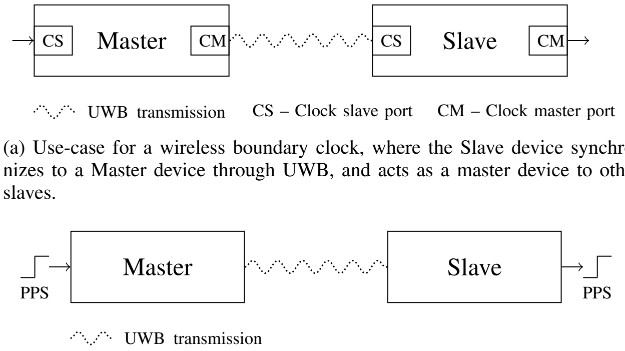


Fig. 1. Basic PTP message exchange [4]



(a) Use-case for a wireless boundary clock, where the Slave device synchronizes to a Master device through UWB, and acts as a master device to other slaves.

(b) Wireless PPS signal transmitter.

Fig. 2. The target use-cases for the UWB based PTP synchronization.

wiring would be hard and expensive between two subnetworks, or there are no transparent clocks between the subnetworks. In this initial version we are focused on the second use-case where the aim is to transmit a PPS signal from the master to the slave device, e.g. for diagnostic purposes to check the synchronization between remote devices without transmitting the PPS signal through wire.

In our scenario, the master device assumed the PTP master role, in order to synchronize the slave clock to the master.

Naturally, we use the high precision timestamping feature of the UWB device to accurately timestamp the PTP messages on the UWB radio link. However, this time instead of the physical distance, the delay is calculated.

To calculate the offset value from the PTP master, the master-to-slave and the slave-to-master offset need to be defined (see Fig. 1):

$$\begin{aligned} d_{\text{m2s}} &= t_1 - t_2 \\ d_{\text{s2m}} &= t_3 - t_4 \end{aligned} \tag{1}$$

From this, the offset from the master can be calculated:

$$\Delta t = d_{\text{m2s}} - d_{\text{prop}} = d_{\text{m2s}} - \frac{d_{\text{m2s}} + d_{\text{s2m}}}{2} \tag{2}$$

, where d_{prop} is the propagation delay. However, this calculation requires the TX and RX events to be timestamped by the PTP clock.

A. Switching local clock domain between SoCs

The main issue regarding the synchronization of PTP clocks using UWB communication is the different clock domains of the UWB transmitter and the PTP clock. Note, that clock domain in this section means the different local clock speeds of the SoC and the UWB transmitter, and not the clock domain concept of PTP! Fig. 3 shows a typical UWB transmitter solution with its clock domains, where the UWB transmitter is controlled by a general SoC (System-on-a-Chip). While UWB communication typically requires a quality clock signal (i.e. UWB is very sensitive to the phase noise), SoC microcontrollers are satisfied with cheap oscillators. To synchronize the PTP clock, the timestamp of RX and TX events needs to be timestamped in the PTP clock domain. The question is, if the exact timestamp of some event is known in the clock domain of the UWB transmitter, how the timestamp of the same event can be calculated in the clock domain of the PTP clock?

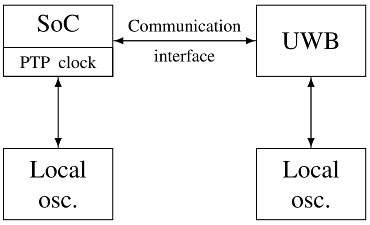


Fig. 3. Typical ultra wide-band hardware solution using different clock domains on the UWB transmitter and the SoC

Having two ideal clocks in different clock domains, the general clock model can be written for both clocks:

$$C^{\text{SoC}} = \int_0^t f^{\text{SoC}}(t)\, dt + o^{\text{SoC}} \tag{3a}$$

$$C^{\text{UWB}} = \int_0^t f^{\text{UWB}}(t)\, dt + o^{\text{UWB}} \tag{3b}$$

Here C is the actual clock counter value, $f(t)$ is the actual frequency, o is the offset of the clock and t is the time. Using crystal oscillators or even TCXOs, it is straightforward that the frequency drift between f^{SoC} and f^{UWB} accumulates very fast. One solution is to use expensive OCXO oscillators with low frequency error (i.e. 1 ppb), however, it is easier to use the same clock for the SoC and the UWB transmitter, i.e. $f^{\text{SoC}}(t) = f^{\text{UWB}}(t)$. In this case, subtracting Eq. 3b from Eq. 3a:

$$C^{\text{SoC}} = C^{\text{UWB}} + (o^{\text{SoC}} - o^{\text{UWB}}) = C^{\text{UWB}} + \Delta o \tag{4}$$

, where Δo is the offset between the SoC and the UWB chip clocks.

However, even with common clock source the offsets in each clock domain cannot be controlled, since the PLLs (Phased Locked Loop) and other circuits make the clock

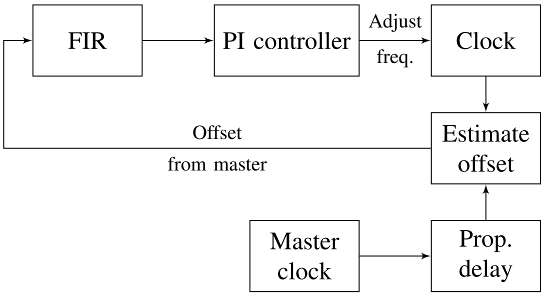


Fig. 4. The clock servo algorithm used to synchronize the PTP clock. To eliminate clock offset and drift, a PI controller is applied.

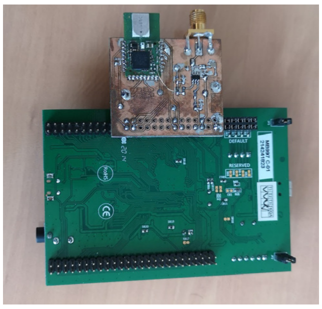


Fig. 5. The evaluation device is based on the STM32F407 evaluation board, and extended by a user-made board containing the DWM1000 module and the clock circuits with an optional SMA connector for the external clock signal.

initialization somewhat stochastic, so Δo can change at each reset. Denote Δo_m the offset at the master and Δo_s the offset at the slave. Define

$$\begin{aligned} t_1 &= C_1^{\mathrm{UWB}} + \Delta o_m \\ t_2 &= C_2^{\mathrm{UWB}} + \Delta o_s \\ t_3 &= C_3^{\mathrm{UWB}} + \Delta o_s \\ t_4 &= C_4^{\mathrm{UWB}} + \Delta o_m \end{aligned} \tag{5}$$

Combine Eq. 5 and Eq. 2, which yields

$$\begin{aligned} \Delta t = C_1^{\mathrm{UWB}} - C_2^{\mathrm{UWB}} + \Delta o_m - \Delta o_s - \\ - \frac{C_1^{\mathrm{UWB}} - C_2^{\mathrm{UWB}} + C_3^{\mathrm{UWB}} - C_4^{\mathrm{UWB}}}{2} \end{aligned} \tag{6}$$

Unfortunately, the difference $\Delta o_m - \Delta o_s$ is unknown and dependent on the initial offset of the clock, which is mainly random. Ideally, Δo_m should equal Δo_s, and there shall be some process providing constant Δo offsets. To achive this, some kind of synchronization facility needs to be provided by the UWB transmitter, e.g. DW1000 chip is able to reset the internal timebase triggered by an external pulse. A concrete example is presented in Section IV; however, the exact method of synchronization depends on the actual hardware capabilities used in the implementation.

B. Clock servo algorithm

For adjusting the PTP clock, a simple PI controller is applied (see Fig. 4.). There are a couple of other, more sophisticated solutions to implement the clock servo algorithm, but for the goals of this paper, the simple PI controller is well-suited since the errors stay direct enough to be evaluated from the viewpoint of timestamping.

The controller itself is driven by the offset from the master error signal (Δt), and passed through a FIR filter, which smooths the random errors of timestamp measurement. The PI controller outputs the control signal, which is used to tune the frequency of the PTP clock.

IV. IMPLEMENTATION

The implementation is based on a STM32F407 SoC, which is a powerful, widely used ARM Cortex M4-based device used in embedded systems. The chip has an Ethernet MAC with a built-in frequency tunable PTP clock, which is able to timestamp RX and TX events on the MAC layer. The UWB communication is provided by the well-known Qorvo DW1000 chip, which has excellent timestamping and external synchronization capabilities. The evaluation devices are designed by the authors, and can be seen in Fig. 5.

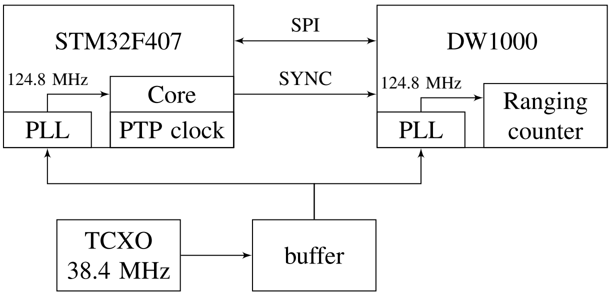


Fig. 6. The synchronization circuit for evaluating the clock synchronization algorithm. The SoC (STM32F407) and DW1000 UWB transmitter are communication through an SPI interface, and the clock synchronization is performed by the Sync signal. The ranging counter and the SoC system clock (and also the PTP clock) are running on 124.8 MHz.

Fig. 6 shows the connections between the components. The clock is generated by a 38.4 MHz thermal-controlled oscillator (TCXO) with a very low phase noise (-132 dBc/Hz at 1 kHz). The clock is distributed between the components by a PL133 clock distributor IC, which also provides low additive phase noise (in the order of femtoseconds). The DW1000 chip uses a PLL to generate the 124.8 MHz signal to drive the ranging counter. The ranging counter is 40 bits wide, providing a resolution of 15 picoseconds. However, the actual timestamping happens on the 124.8 MHz system clock (i.e. the last 9 bits are zeros), and the chip uses a special algorithm based on the channel impulse response to fine-tune the timestamp [21]. While the STM32F407 SoC can operate up to 168 MHz, to simplify implementation the STM32F407 SoC is also programmed to generate a $f_{\mathrm{SYS}} = 124.8$ MHz signal, and the PTP clock and also the Cortex core are driven by this clock.

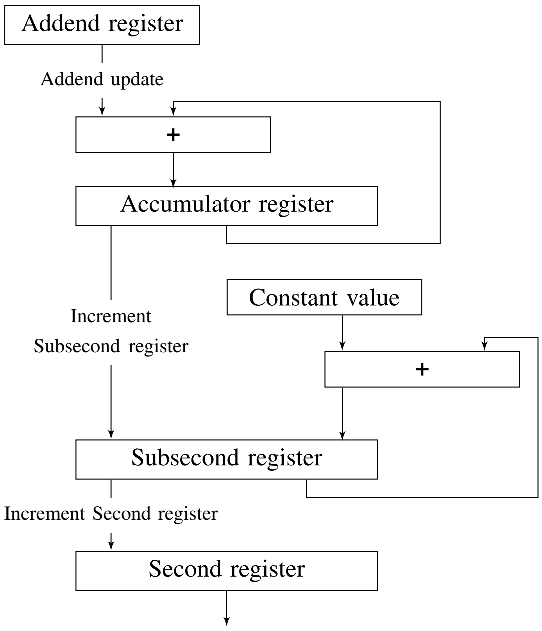


Fig. 7. The operation of the PTP clock. The PTP clock consists of a second and a subsecond register. The subsecond is incremented by a constant when the accumulator register is overflown. [22]

A. The hardware PTP clock

The hardware PTP clock (PHC – PTP Hardware Clock) consists of a 32-bit wide second register and a 31-bit wide subsecond register, allowing a resolution of 0.46 nanoseconds [22]. While the PTP clock is driven by the 124.8 MHz clock, the counter is virtually running somewhat slower. The frequency of the counter can be tuned by a so-called addend register (C_{addend}) shown in Fig. 7, which is added to the accumulator register at each clock cycle, effectively changing the clock frequency as

$$f_{\text{PTP}} = \xi \cdot f_{\text{SYS}} \cdot \tau \quad \xi = \frac{C_{\text{ADDEND}}}{2^{32}} \quad \tau = \frac{C_{\text{SS}}}{2^{31}} \tag{7}$$

, where C_{SS} is the value of the 8-bit subsecond register, τ is the resolution of the PTP clock and ξ is the frequency tune coefficient.

Changing the addend register, the clock frequency changes. To select the correct values for the addend and the subsecond register, one should compromise the resolution of the PTP clock and the limits of the clock frequency tuning. While the former provides better accuracy, the latter limits the control signal from the PI controller.

B. Synchronization with the DW1000 chip

The main issue in synchronization is that the packet receive and transmit time instants shall be known by the PTP clock. Typically, the PTP clock is integrated in the hardware that receives and transmits the packets. Also, the DW1000 UWB chip provides timestamps using his own counter (with his own offset Δo), however, the clock of the DW1000 chip cannot be tuned, thus it is not suitable to use as a PTP clock. To synchronize a PTP clock outside of the DW1000 clock domain, external synchronization facilities shall be used. External synchronization is implemented by a special SYNC signal in the DW1000 chip. There are two modes used: one-shot transmit synchronization mode (OSTS) and the one-shot timebase reset (OSTR).

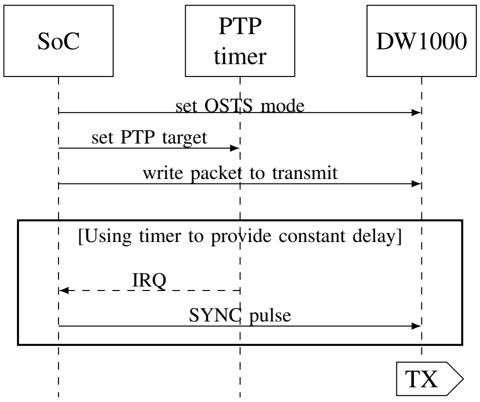


Fig. 8. The method of TX synchronization. The transmission starts after the SYNC pulse with a constant delay.

One shot timebase reset mode allows a reset to be applied to the timebase counter used for timestamping in DW1000 at a deterministic and predictable time relative to a synchronisation event (i.e. the SYNC pin). The DW1000 chip will reset the counter at a repeatable time to typically less than 100ps variation. Besides OSTR mode, one-shot transmit synchronization mode provides for the transmission of a frame at a well-defined time (typically 12 ps) relative to the assertion of the SYNC pin of the DW1000 chip. To learn more of the mechanism of OSTS and OSTR mode, the DW1000 user manual has a detailed description [21].

1) Transmission synchronization:

1) Transmission synchronization: For scheduled transmission, OSTS mode is used. For transmitting a packet at a specified moment in the future, the PTP clock target is set to give an interrupt (see Fig. 8). However, interrupts on Cortex M4 are not ensured to be real-time, so it introduces a jitter of a couple of clock cycles. To avoid the jitter, the PTP interrupt triggers a timer, which constructs a fixed-sized SYNC pulse in constant time. At the rising edge of the SYNC pulse, the DW1000 is guaranteed to start transmission in a fixed time delay. Notice that the transmission timestamp can be included in the message, so one-step method can be used which does not need to send a *Follow_Up* message.

2) Receive synchronization: Receive synchronization has to provide constant Δo offset (see Section III-A). The DW1000 chip is able to reset its timebase against a rising edge of the SYNC signal, using the external synchronization mode OSTR. When setting the SYNC signal, the PTP clock shall be read and stored as t_{sync} (see Fig. 9). In a Cortex M4 processor, the two 32-bit registers of the PTP clock and the GPIO change cannot be performed simultaneously; however, it can be done

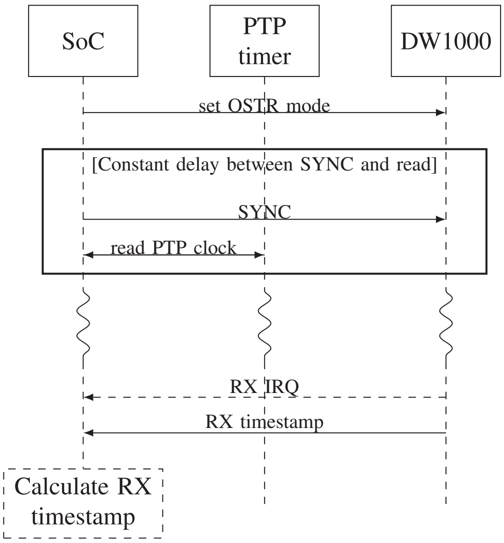


Fig. 9. The method of RX synchronization. The transmission starts after the SYNC pulse with a constant delay.

with a constant delay with disabled interrupts.

Later on, at reception, the DW1000 timestamp can be read out, and the PTP timestamp for the receive event can be calculated as

$$t_{\mathrm{PTP}} = t_{\mathrm{SYNC}} + \frac{C_{\mathrm{DW}}}{2^9} \cdot \xi \cdot f_{\mathrm{SYS}} \cdot \tau \tag{8}$$

, where C_{DW} is the 40 bit DW1000 timestamp.

C. Timestamping errors

In the implementation, there are a couple of potential error or jitter sources.

1) PTP clock resolution: One of the most basic jitter source is the resolution and frequency of the PTP clock. The limited resolution (see Eq. 7) of the PTP clock results in a classical sampling noise; however, the situation is worse as the frequency tuning by the addend register deviates from the sampling points. Increasing the effective frequency of the PTP clock results in a decrement in the sampling noise.

2) Wireless propagation: The stochastic behavior of wireless propagation results in timestamping errors, i.e. the first component that arrived is wrongly detected and timestamped. To avoid this issue, UWB devices observe the channel impulse response to refine the timestamp, acquiring subnanosecond accuracy of timestamping. However, pure NLOS (non line-of-sight) propagation can add higher timestamp offset, which is hard to detect.

3) Clock domain synchronization error: The main source of error results from the clock domain switch between the SoC and the UWB transmitter. The SYNC signal of the DW1000 chip is sampled at the rising edge of the 38.4 MHz clock; however, SoC runs on a faster clock, 124.8 MHz. The SoC toggle the SYNC signal on its own clock, which can result in some jitter since SYNC signals at different rising edges of the SoC clock sampled at the same rising edge of the DW CLK (see Fig. 10). This error can introduce around 26 nanoseconds of jitter or uncertainty in timestamping, due to the 38.4 MHz sampling clock.

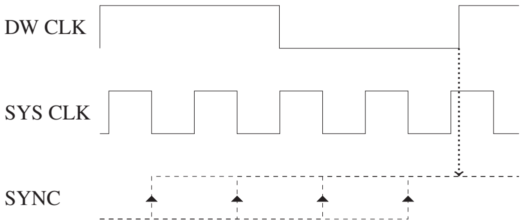


Fig. 10. Jitter resulting from clock domain change. The SYNC signal is sampled at the rising edge of the DW clock, but driven by the SYS clock.

TABLE I
THE SETTINGS USED FOR EVALUATION

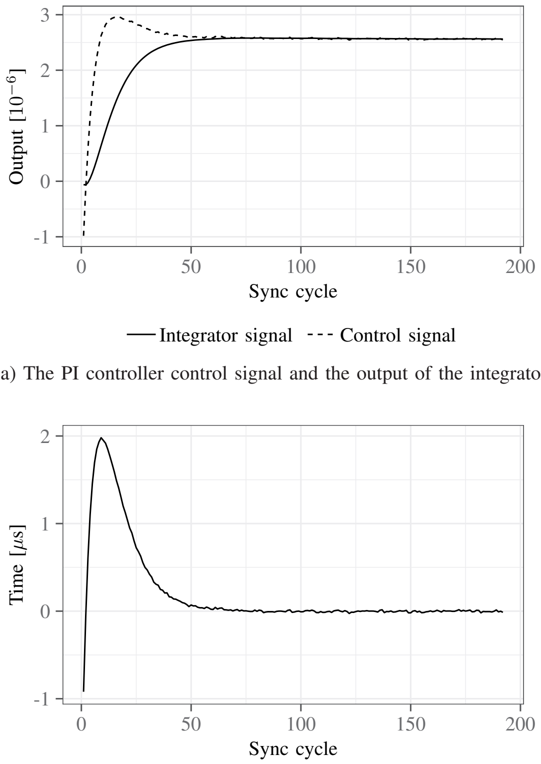
Subsecond update	0x14
Addend	0xDC41363A
Sync period	500 ms
SYSCLK	124.8 MHz
K_i (PI)	15
K_p (PI)	1
FIR	[0.5,0.5]
UWB channel	5
UWB bitrate	850 kbps
UWB PRF	64 MHz
UWB preamble syms	1024

V. EVALUATION

The evaluation is based on the implemented synchronization method presented in Section IV. For evaluation, two devices were used, one is a free-running master device, and the other one is the slave device synchronizing to the master device. The setting of the arrangement is summarized in Table I.

After a cold start, the PI controller starts to converge to the synchronous state, as shown in Fig. 11. The figure shows that the error signal's steady state converges to zero in the order of nanoseconds, and the control signal for the PTP clock converges to a steady state. The control signal has a small overshoot, and the convergence happens in 70-80 sync cycles corresponding to 35-40 seconds. The parameters of the PI controller were selected for the system to be stable with a high integrator coefficient. The high integrator coefficient helps to avoid the overreaction resulting from timestamping and clock domain change errors. However, the paper do not target the design method of the optimal controller for transient behaviour, only concentrates on the steady state.

After reaching the steady state, 1000 measurements were made by a DSO-X 3054A oscilloscope, using the Pulse Delay Measurement function to analyze and record the delay between the PPS signals of the master and slave PTP clock (see Fig 12). The result can be seen in Fig. 13 as a histogram. It is clearly seen that the error between the two PPS signals does not exceed 30 nanoseconds in absolute value; however, the distribution is skewed, and there are less negative errors than positive errors. 90 percent of the absolute errors are below



(a) The PI controller control signal and the output of the integrator

(b) The error signal of the PI controller

Fig. 11. The error signal and the output of the PI controller of the clock servo. The time axis shows the synchronization cycles, where one cycle is 500 ms. It can be clearly seen, that the error signal is in the order of nanoseconds.

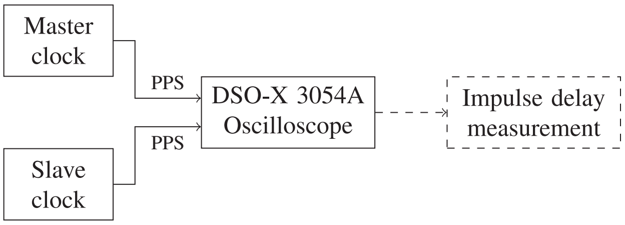


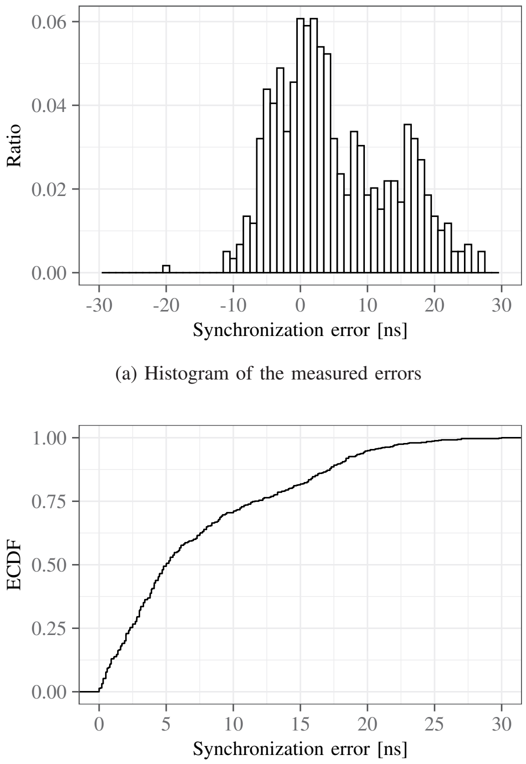
Fig. 12. The measurement method of the clock synchronization error between the master and slave devices.

20 nanoseconds, while 70 percent of the absolute errors are below 10 nanoseconds.

The skewness of the distribution assumes that the timestamping error is asymmetric, i.e. it is not exactly the same in the case of master-to-slave and slave-to-master. The errors are likely to originate from a couple of sources presented before:

1) The error resulting from clock domain change.
2) The PTP clock resolution.
3) Multipath propagation.

It is worth noticing that the magnitude of the error can be explained by the clock domain change error in itself and is well below the requirements of an industrial profile TSN network (i.e. 250ns).



(a) Histogram of the measured errors

(b) The empirical cumulative distribution plot of the absolute error

Fig. 13. Distribution of the error measured between the two PPS signals. The histogram is based on 1000 sync cycle after the controller has locked.

VI. Conclusions

The paper investigates the feasibility of PTP clock synchronization using the wireless Ultra Wideband technology, which provides an accurate timestamping feature for packet transmission and reception. Both the methodology and the implementation-specific issues are presented. The results show that the synchronization accuracy is in the order 10 nanoseconds. This clock synchronization accuracy is on the same scale as the commercially available wired solutions. It can be concluded that – even in its initial form – the presented UWB implementation can be an excellent solution for an accurate wireless PTP master clock synchronization or as a validation method for clock synchronization due to its low cost and wireless properties. However, for the validation of synchronization accuracy and precision, the presented solution is useful only in PTP wireless environments or in PTP networks with PTP unaware devices, since the resolution is the same as the accuracy of the state-of-the-art methods. As a bonus, localization is also provided as the original use case for UWB.

The main causes of the synchronization error are also presented, anticipating the possibility of further improving the accuracy of the synchronization. Specifically, the clock domain switch error has the same magnitude as the overall synchronization error, thus estimating the constant delay at the clock domain switch the synchronization accuracy can be drastically improved.

Regarding future work, the synchronization error of the master and slave device can be further reduced with fine tuning of PI controller or with the implementation of a more sophisticated solution such as Kalman-filtering. Another possible future research direction would be the evaluation and measurements of the different LOS and NLOS scenarios and the characteristics of the distance between the master and slave device. Furthermore, as the UWB technology has significant limitations beyond a certain distance, there is some initial research on a multi-hop UWB PTP system. Such a system can provide clock synchronization on the order of 10 ns over many times the UWB radio range. However, in this case, the synchronization errors are accumulated, offering an exciting research topic.

References

[1] O. Seijo, J. A. López-Fernández, H.-P. Bernhard, and I. Val, "Enhanced Timestamping Method for Subnanosecond Time Synchronization in IEEE 802.11 Over WLAN Standard Conditions," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5792–5805, 2020, doi: 10.1109/TII.2019.2959200.

[2] Z. Idrees, J. Granados, Y. Sun, S. Latif, L. Gong, Z. Zou, and L. Zheng, "IEEE 1588 for Clock Synchronization in Industrial IoT and Related Applications: A Review on Contributing Technologies, Protocols and Enhancement Methodologies," *IEEE Access*, vol. 8, pp. 155 660–155 678, 2020, doi: 10.1109/ACCESS.2020.3013669.

[3] M. Lipiński, T. Włostowski, J. Serrano, and P. Alvarez, "White rabbit: a PTP application for robust sub-nanosecond synchronization," in *2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, 2011, pp. 25–30, doi: 10.1109/ISPCS.2011.6070148.

[4] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588- 2019 (Revision of IEEE Std 1588-2008)*, pp. 1–499, 2020, doi: 10.1109/IEEESTD.2020.9120376.

[5] "IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications," *IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011)*, pp. 1–421, 2020, doi: 10.1109/IEEESTD.2020.9121845.

[6] I. Val, O. Seijo, R. Torrego, and A. Astarloa, "IEEE 802.1AS Clock Synchronization Performance Evaluation of an Integrated Wired–Wireless TSN Architecture," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 2986–2999, 2022, doi: 10.1109/TII.2021.3106568.

[7] H. Shi, A. Aijaz, and N. Jiang, "Evaluating the performance of over-the-air time synchronization for 5g and tsn integration," in *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2021, pp. 1–6, doi: 10.1109/BlackSea-Com52164.2021.9527833.

[8] T. Adame, M. Carrascosa-Zamacois, and B. Bellalta, "Time- Sensitive Networking in IEEE 802.11be: On the Way to Low-Latency WiFi 7," *Sensors*, vol. 21, no. 15, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/15/4954. doi: 10.3390/s21154954

[9] J. Haxhibeqiri, X. Jiao, M. Aslam, I. Moerman, and J. Hoebeke, "Enabling TSN over IEEE 802.11: Low-overhead Time Synchronization for Wi-Fi Clients," in *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, vol. 1, 2021, pp. 1068–1073, doi: 10.1109/ICIT46573.2021.9453686.

[10] "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 Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016, doi: 10.1109/IEEESTD.2016.7786995.

[11] G. Hollósi, "Distribution of ultra wideband (UWB) receive timestamps in dense indoor environment based on the Saleh-Valenzuela channel model," in *2022 14th International Conference on Communications (COMM)*, 2022, pp. 1–5, doi: 10.1109/COMM54429.2022.9817167.

[12] A. Jiménez and F. Seco, "Finding objects using uwb or ble localization technology: A museum-like use case," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017, pp. 1–8, doi: 10.1109/IPIN.2017.8115865.

[13] W. Shule, C. M. Almansa, J. P. Queralta, Z. Zou, and T. Westerlund, "UWB-Based Localization for Multi-UAV Systems and Collaborative Heterogeneous Multi-Robot Systems," *Procedia Computer Science*, vol. 175, pp. 357–364, 2020, mobiSPC. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920317324. doi: 10.1016/j.procs.2020.07.051

[14] M. Drobczyk, C. Strowik, and C. Philpot, "A Wireless Communication and Positioning Experiment for the ISS Based on IR-UWB," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6, doi: 10.1109/WCNC.2017.7925487.

[15] S. Rinaldi, A. Musatti, A. Depari, P. Ferrari, A. Flammini, and E. Sisinni, "An Experimental Characterization of Chain of PLLs for Wired Clock Synchronization of UWB Anchors for Indoor Location," in *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2022, pp. 1–6, doi: 10.1109/I2MTC48687.2022.9806698.

[16] R. Zou, N. Wu, Z. Qu, and 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.

[17] P. Ferrari, P. Bellagente, A. Depari, A. Flammini, M. Pasetti, S. Rinaldi, and E. Sisinni, "Resilient time synchronization opportunistically exploiting uwb rtls infrastructure," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022, doi: 10.1109/TIM.2021.3132354.

[18] M. Segura, S. Niranjayan, H. Hashemi, and A. F. Molisch, "Experimental demonstration of nanosecond-accuracy wireless network synchronization," in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 6205–6210, doi: 10.1109/ICC.2015.7249312.

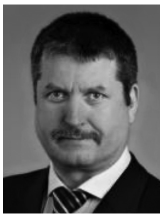
[19] B. Xue, Z. Li, P. Lei, Y. Wang, and X. Zou, "Wicsync: A wireless multi-node clock synchronization solution based on optimized UWB two-way clock synchronization protocol," *Measurement*, vol. 183, p. 109760, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026322412100717X. doi: 10.1016/j.measurement.2021.109760

[20] "IEEE standard for low-rate wireless networks," *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, 2020, doi: 10.1109/IEEESTD.2020.9144691.

[21] D. Ltd., DW1000 User Manual, *"How To use, configure and program the DW1000 UWB transmitter"*, Qorvo, 2017.

[22] ST, *Reference manual, STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm-based 32-bit MCUs*, 2021.

Gergely Hollósi is a researcher at Dept. of Telecommunications and Media Informatics (TMIT) of Budapest University of Technology and Economics (BME). Gergely received his M.Sc. in the Budapest University of Technology and Economics (BME) in 2009. He is actively working and researching on computer vision, image processing, machine learning algorithms and indoor localization systems.



István Moldován is a Research Fellow at the Budapest University of Technology and Economics in the Department of Telecommunications and Media Informatics. In 1996, he received an M.Sc. degree in Automation and Industrial Informatics from the Technical University of Tirgu-Mures, Romania. His research interests include network management, embedded systems, simulation and performance evaluation of computer networks. He is lecturing on communication networks.