

High-Precision Wireless Synchronization: When Wi-Fi meets UWB

Louis Adriaens, Wei Liu, Jetmir Haxhibeqiri, Jeroen Hoebeke, Ben Van Herbruggen, Stijn Luchie,
Eli De Poorter and Pablo Esteban Avila Campos - IDLab, Ghent Univeristy - imec

Abstract—Seamless and reliable communication is crucial across personal and industrial domains in today’s interconnected world, ranging from multimedia streaming and augmented/virtual reality to smart manufacturing. This need extends to wireless communication, which supports mobile applications and offers greater flexibility. Time-Sensitive Networking (TSN) addresses stringent real-time demands with low latency, low jitter, and high availability, relying on precise time synchronization via the Precision Time Protocol (PTP). While PTP is widely adopted in wired networks, its usage in wireless networks is less common. This paper investigates the potential of Ultra-Wide Band (UWB) technology to achieve high-precision clock synchronization in Wireless-TSN (W-TSN). Traditionally used for indoor localization owing to its high-precision timestamping, UWB offers reliability and resilience against multipath fading. This research integrates UWB into Openwifi, an open-source Software Defined Radio (SDR) Wi-Fi chip design augmented with TSN capabilities, serving as the W-TSN solution. Significant improvements in synchronization accuracy were achieved by increasing the timestamping resolution and refining the clock drift correction algorithm within the Openwifi FPGA. This resulted in a Mean Absolute Error (MAE) of 13.1 ns and a 90th percentile (P_{90}) clock error of 25.0 ns. The integration of UWB technology further improved clock synchronization accuracy by roughly 25%, achieving an MAE of 9.98 ns with a P_{90} error of 20.0 ns. Finally, the study highlights UWB-based synchronization’s potential applications and benefits, including inherent localization capabilities, while noting limitations such as range constraints and added complexity.

Wireless Time-Sensitive Networking (W-TSN) · Precision Time Protocol (PTP) · Clock Synchronization · Ultra-Wide Band (UWB) · Openwifi

I. INTRODUCTION

Today, connectivity is a fundamental aspect of both personal and industrial applications. The demand for seamless and reliable communication networks has grown exponentially as industries increasingly rely on interconnected systems to optimize operations, innovate services, and enhance sustainability. This digital transformation, often referred to as Industry 4.0, is reshaping industrial automation by connecting computers, intelligent machines, people, and processes [1], [2]. From smart manufacturing and aerospace communication to robotic systems, multimedia streaming, and augmented/virtual reality, the need for robust and deterministic communication networks has never been more critical [3].

However, these requirements are essential not only for wired networks but also for wireless networks. Many applications demand the flexibility of wireless communication to connect mobile nodes without the constraints of physical cables. These wireless networks must meet the same high standards of reliability and robustness to support real-time

applications in dynamic and often harsh environments [1], [4].

In this context, Time-Sensitive Networking (TSN) [5] has emerged as a vital technology to support applications with stringent real-time constraints, such as low latency, low jitter, high availability, and high reliability [1]. The foundation of TSN’s efficacy is precise time synchronization [6], allowing devices to share a common understanding of time. This enables the establishment of deterministic schedules, ensuring devices coordinate transmissions and receptions within agreed-upon time frames to meet stringent requirements of time-sensitive applications.

The Precision Time Protocol (PTP) [7] is commonly employed to achieve synchronization among devices. Through the exchange and timestamping of packets, devices can calculate the differences between their clocks and make the necessary adjustments. While PTP is widely adopted in wired networks, its usage in wireless networks is less common. Wireless networks introduce additional challenges, such as the potential of packet loss due to collisions in the shared and dynamic medium, which can significantly impact synchronization performance. Furthermore, the precision of timestamping mechanisms varies; for example, software timestamping often introduces measurement noise due to processing delays. Additionally, issues such as clock frequency discrepancies (known as ‘clock drift’) and limitations in clock resolution further constrain synchronization precision and, consequently, the capabilities of TSN.

Emerging technologies like Ultra-Wideband (UWB) present intriguing possibilities in this context. Traditionally, UWB finds applications in areas like indoor localization [8], where the demand for precise tracking and positioning of equipment, materials, and products is crucial as industries progress towards more intelligent manufacturing processes [8]. Utilizing a broad frequency spectrum, UWB packets are short and resilient against multipath fading, a common wireless communication challenge [9].

Central to UWB’s positioning functionality are schemes like Time Difference of Arrival (TDoA), which use protocols such as Two-Way Ranging (TWR) to obtain accurate distance measurements through precise packet timestamping [10].

Given these robust timestamping capabilities, UWB presents an exciting opportunity to enhance time synchronization accuracy in Wireless-TSN (W-TSN). In addition, UWB positioning systems often coexist with other wireless technologies, such as Wi-Fi, the dominant indoor network. Many IoT devices have both UWB and Wi-Fi interfaces, typically operating separately. Integrating them allows UWB

to provide accurate positioning and serve as a dedicated channel for time synchronization, alleviating the burden on the main communication network. Therefore, this study investigates how integrating UWB technology with W-TSN can improve synchronization precision and explores the additional benefits of UWB's inherent localization capabilities in various applications.

To explore this integration, this research leverages Openwifi [11], a state-of-the-art, open-source Wi-Fi chip design implementation on an FPGA-Linux platform that uses SDR. Openwifi's flexibility and extensibility make it an ideal platform for integrating new technologies. It has been extended with TSN features, including synchronization and scheduling, providing a robust foundation for the W-TSN solutions [1]. The goal is to improve Openwifi's time synchronization capabilities by integrating UWB technology.

The UWB technology used in this research includes a DWMEVB3000 board [12] from Qorvo, which houses Decawave's DW3000 UWB transceiver, paired with a nrF52840 DK [13] from Nordic. These UWB modules provide high-precision hardware timestamping capabilities, which are crucial for obtaining accurate ranging measurements for localization and achieving the desired synchronization accuracy.

The remainder of this paper is structured as follows: Section II presents a summary of the relevant prior research. Following this, Section III details the integration of UWB modules with the Openwifi system, presenting various experiments aimed at enhancing clock synchronization accuracy. The main benefits and considerations of UWB-based clock synchronization methods are then discussed in Section IV, exploring various application scenarios. Finally, Section V summarizes the conclusions drawn from the study and discusses future work.

II. RELATED WORK

With the ever-increasing demand for real-time applications in the wireless domain, there has been a surge in research focusing on clock synchronization over WLAN. Mahmood et al. [14] provide a comprehensive survey on clock synchronization over IEEE 802.11, comparing methods like PTP and Network Time Protocol (NTP) while analyzing their performance.

Chen and Yang [15] delve into both software timestamping using Commercial-Off-The-Shelf (COTS) Wi-Fi devices and hardware PTP implementations utilizing the hardware TSF counter. Their findings indicate that achieving microsecond-level synchronization accuracy is feasible with both software and hardware PTP approaches.

Furthermore, research concerning UWB with time synchronization for wireless networks is gaining traction. De Dominicis et al. [16] present UWB-based sub-nanosecond accurate hardware timestamping, underscoring UWB's success due to its robustness against interference and multipath fading. Meanwhile, Anwar and Srivastava [17] propose leveraging IPv6 over Low power Personal Area Network (6LoWPAN) enabled UWB radios as a wireless Precise

Hardware Clock (PHC), achieving sub-nanosecond synchronization accuracy.

This research builds upon the work of Aslam et al. [6], who proposed a PTP implementation in IEEE 802.11 Wireless Local Area Networks (WLANs) on the Openwifi platform. Unlike traditional methods that either rely on software timestamping, yielding poor synchronization accuracy, or relying on a dedicated PTP hardware clock, their solution leverages the Timing Synchronization Function (TSF) clock inherent in the IEEE802.11 standard used for synchronization between access points and WLAN stations. By using the existing TSF clock with a resolution of $1 \mu s$, they bypass the need for a dedicated hardware clock, thereby reducing complexity and cost.

In single-hop WLAN scenarios, their proposed PTP solution yielded a mean clock offset of $-0.279 \mu s$, with a standard deviation of $0.820 \mu s$, and a 90th percentile (P_{90}) of $1.40 \mu s$. Notably, this performance meets the criteria set by the Wi-Fi Alliance for the *Wi-Fi Timesync* certificate, which mandates a P_{90} within $5.5 \mu s$. As such, their work provides a valuable benchmark for synchronization performance.

As previously mentioned, TSN features have been introduced to Openwifi, the SDR Wi-Fi system used throughout this study. Haxhibeqiri et al. [1] present various considerations that need to be addressed to incorporate TSN features into wireless networks, meeting the demand for real-time and deterministic communication while maintaining the flexibility of wireless networks. Based on these considerations, they realize a proof of concept TSN-capable Wi-Fi system based on Openwifi.

One of the crucial aspects highlighted in their work is accurate clock synchronization. Using the same clock synchronization solution presented by Aslam et al. [6], Haxhibeqiri et al. achieved a similar synchronization performance, reporting a P_{90} of $1.3 \mu s$. While Aslam's work focused purely on enabling synchronization on Openwifi, Haxhibeqiri's study extends this to explore additional aspects of wireless TSN, including scheduling and broader system integration, demonstrating how W-TSN can be established on a Wi-Fi network with clock synchronization capabilities.

This research strives to enhance the clock synchronization performance on Openwifi, particularly through the integration of UWB technology, leveraging its precise timestamping capabilities. The UWB clock has a resolution of $15.65 ps$, which offers a much greater timestamping resolution than Openwifi's TSF clock with a resolution of $1 \mu s$. Furthermore, considering the RF domain, UWB's sampling rate is much higher, utilizing a bandwidth of $499.2 MHz$ [18], compared to Wi-Fi's $20 Msps$ sampling rate [19]. There is thus significant room for improvement in clock synchronization accuracy when utilizing UWB. In addition to these timestamping capabilities, enhancements are made to the TSF block in FPGA to refine the timestamping and drift correction processes.

The potential of UWB for improving PTP clock synchronization has been explored by Hollósi and Moldován [20]. They highlight UWB's advantages, such as precise

a TWR exchange, albeit with different labels, as TWR typically uses the names `Poll`, `Response`, and `Final`.

However, due to the slow transmission speed of UART, directly integrating the Openwifi timestamps into the PTP-TWR message exchange would significantly delay the `Delay_Resp` message. To address this, an additional follow-up message is introduced to transmit the Openwifi timestamps after they are read out over UART. This adapted message exchange is illustrated in Fig. 3.

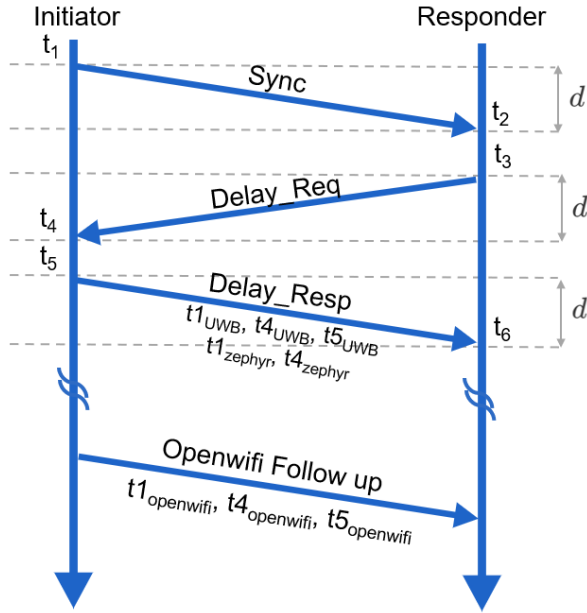


Fig. 3. Structure of the adapted PTP-TWR message exchange incorporating the Openwifi timestamps using an extra follow-up message.

In PTP, the responder uses the first four timestamps to calculate the clock offset, compensating for the incurred delay (d). Assuming a constant clock offset and symmetrical delay during the exchange, the delay can be computed as half of the Round Trip Time (RTT) minus the response time ($t_3 - t_2$).

The calculations proceed as follows: The master-to-slave difference (Δ_{m2s}) and slave-to-master difference (Δ_{s2m}) are determined:

$$\Delta_{m2s} = t_2 - t_1 = \tilde{o} + d$$

$$\Delta_{s2m} = t_4 - t_3 = -\tilde{o} + d$$

Here, \tilde{o} denotes the estimated offset, and d represents the one-way delay. These values are then derived as follows:

$$\tilde{o} = \frac{\Delta_{m2s} - \Delta_{s2m}}{2}$$

$$d = \frac{\Delta_{m2s} + \Delta_{s2m}}{2}$$

The slave clock corrects its clock by subtracting the estimated offset \tilde{o} . However, accurate time synchronization requires accounting for clock drift, which quantifies the rate

of change of the clock offset \tilde{o} over time. Clock drift is calculated as:

$$drift = \frac{(t_2 - t_{2,prev}) - (t_1 - t_{1,prev})}{t_1 - t_{1,prev}}$$

Here, $t_{1,prev}$ and $t_{2,prev}$ denote the timestamps from the previous PTP message exchange. The slave clock can then periodically adjust to match the desired frequency [6].

TWR, on the other hand, uses all six timestamps to calculate the delay d , as described by the following formula [22]:

$$d = \frac{(t_4 - t_1)(t_6 - t_3) - (t_3 - t_2)(t_5 - t_4)}{(t_4 - t_1) + (t_6 - t_3) + (t_3 - t_2) + (t_5 - t_4)}$$

This formula accounts for clock drift during the exchange, unlike the PTP exchange, which assumes a constant clock offset throughout. This consideration leads to a more accurate estimation of the delay, enhancing the precision and robustness of distance estimation in TWR.

B. Experimental Evaluation

The setup was initially tested with a two-second PTP-TWR message exchange interval using software timestamping within Openwifi. However, due to the overhead and latency involved in handling interrupts and reading TSF timer values in Linux kernel space, the resulting data exhibited significant noise and erratic behaviour. This led to unreliable PTP statistics, highlighting the limitations of software timestamping.

To address these issues, hardware timestamping was implemented. Modifications were made to the FPGA layer to support hardware timestamping by latching the TSF timer value into a register upon the UWB interrupt GPIO pin's rising edge. This registered value could then be accessed and read within the SDR interrupt handler in Linux kernel space, ensuring reliable timestamps and providing accurate PTP statistics.

This approach resulted in microsecond-level synchronization with a P_{90} of $1.0 \mu s$, aligning closely with the existing work of Aslam et al. [6], as discussed earlier in Section II, who achieved a P_{90} of $1.40 \mu s$. The key distinction is using UWB as a communication channel, though it does not yet incorporate UWB clocks or timestamps. The superior performance might also be due to using the same type of Openwifi boards on both ends, whereas Aslam et al. used two different types of Openwifi boards. Using identical boards with the same type of oscillators can ease synchronization effort, as they are manufactured with similar properties, hence they exhibit similar responses to environmental factors like temperature.

Subsequently, several steps were taken to improve the synchronization performance:

1) Improving the Hardware Timestamping Resolution:

The primary limitation was identified as the limited timestamp resolution of $1 \mu s$. To address this, modifications were made to Openwifi's TSF block in the FPGA. The TSF clock

is based on a 100 MHz counter, which generates a pulse every 100 cycles to achieve the desired 1 MHz frequency. By exposing this counter to the hardware timestamping process within the SDR’s FPGA, both the original TSF clock and the 100 MHz counter can be accessed in the interrupt handler within the Openwifi driver in Linux kernel space, effectively increasing the resolution by a factor of 100.

Although this significantly improved the accuracy of the measured PTP statistics, it did not enhance synchronization accuracy. Analyzing the obtained timestamp data revealed that the drift correction algorithm provided by Openwifi was the primary limiting factor. This algorithm achieves skew correction by periodically adjusting the TSF clock based on a provided *ppb* value. The skew correction period in microseconds (S_c) can be calculated using the formula from Aslam et al. [6]:

$$S_c = \left(\frac{10^9}{F_r \times ppb} \right) \times 10^6$$

For example, with a provided drift value of -6.305 ppm, translating to a *ppb* value of -6305, and assuming a TSF clock frequency (F_r) of 1 MHz, this yields an S_c value of -158604 μs . This means that the TSF counter is adjusted backwards by 1 μs every 158.604 ms, slowing down the responder board’s TSF clock to compensate for the drift with the initiator’s TSF clock.

However, applying corrections at 1 μs granularity roughly every 159 ms is too coarse for the two-second PTP-TWR message exchange interval, resulting in some intervals having more corrections than others. These inconsistent drift correction adjustments significantly impact the observed clock frequency, limiting the accuracy of drift correction for the TSF clock.

2) *Refining the Drift Correction Algorithm*: To enhance the drift correction accuracy, the algorithm was adapted to update the internal 100 MHz counter instead of the TSF counter, increasing the resolution from 1 μs to 10 ns. This adjustment allows drift corrections to occur 100 times more frequently.

The updated formula for the skew correction period in microseconds (S_c) is:

$$S_c = \left(\frac{10^9}{F_r \times ppb} \right) \times 10^4$$

For instance, considering again a drift value of -6.305 ppm, translating to a *ppb* value of -6305, and assuming a TSF clock frequency (F_r) of 1 MHz, the updated formula yields an S_c value of -1586 μs , which translates to a 10 ns adjustment every 1.58604 ms.

This refinement significantly improves the accuracy and frequency of drift corrections, yielding a substantial improvement in synchronization performance. The setup achieved a mean absolute error (MAE) of 13.1 ns with a P_{90} of 25.0 ns, representing a fiftyfold reduction in measured clock error between the Openwifi clocks.

3) *Incorporating UWB Timestamps*: Lastly, a multi-hop synchronization approach was used to integrate the UWB timestamps, treating the initiator’s and responder’s UWB clocks as intermediate synchronization steps. First, the offset and drift between the Openwifi TSF clock and the UWB clock at the initiator’s side are calculated. Both the Openwifi board and the UWB module timestamp t_1 during the transmission of the `Sync` message, allowing their difference to be determined by converting the timestamps to the same units. The drift is then determined by calculating the change in offset between successive exchanges. Next, the offset and drift between the two UWB modules are computed. Finally, mirroring the process on the initiator’s side, the offset and drift between the responder’s UWB clock and the Openwifi TSF clock are calculated using t_3 . This ensures both initiator and responder use the transmission of a packet for synchronization, maintaining consistency. The obtained offset and drift are then summarized over each hop to obtain the offset and drift between the two Openwifi clocks.

Integrating UWB timestamps with Openwifi led to a further improvement in synchronization accuracy by approximately 25%, resulting in a mean absolute error (MAE) of 9.98 ns and a P_{90} of 20.0 ns. Table I summarizes the synchronization results at each step, including the standard deviation and clock drift metrics. In addition, the table includes distance measurements obtained from TWR. These findings are based on synchronization measurements conducted over a 20-minute period with a two-second PTP-TWR message exchange interval.

TABLE I
THE MEASURED OPENWIFI TSF CLOCK SYNCHRONIZATION PERFORMANCE.

Metric	MAE	Standard Deviation	P_{90}
<i>Original Resolution Hardware Timestamping (1 MHz)</i>			
Offset	0.446 μs	0.538 μs	1.0 μs
Drift	275 ppb	383 ppb	496 ppb
Distance	43.0 cm	1.31 cm	-
<i>Improved Timestamping & Drift Correction Resolution (100 MHz)</i>			
Offset	13.1 ns	21.9 ns	25.0 ns
Drift	9.14 ppb	13.8 ppb	19.7 ppb
Distance	39.2 cm	1.75 cm	-
<i>With Additional UWB Timestamps</i>			
Offset	9.98 ns	13.2 ns	20.0 ns
Drift	7.15 ppb	9.32 ppb	14.8 ppb
Distance	44.5 cm	1.26 cm	-

IV. USE CASES

This section discusses the advantages and limitations of UWB-based clock synchronization methods, and explores various use case scenarios.

UWB offers reliability, robustness against multipath fading, and supports high-resolution timestamping, making it ideal for synchronization applications [9]. Its suitability extends to dynamic environments like augmented/virtual reality (AR/VR) and robotic systems, where mobility and

precise timing are crucial [3]. Additionally, its localization capabilities enable applications like asset tracking and indoor positioning systems where GPS signals are unreliable. For example, ranging information could support proactive roaming methods, significantly reducing handover time [23].

Moreover, UWB positioning systems often already coexist with Wi-Fi, with many IoT devices having both interfaces. By integrating both technologies, systems can optimize reliability using a dedicated UWB channel for synchronization, while easing congestion on the primary communication network.

Despite its advantages, UWB has limitations, including its limited range and difficulty to penetrate walls due to low transmit power. Although it's efficient for battery-powered devices, challenges arise in environments with thick walls or obstructions [9]. Additionally, the need for dedicated hardware adds a layer of complexity, prompting a thorough assessment to justify its implementation for specific applications.

Several specific potential scenarios are explored, offering unique insights and applications:

1) *AP(s) and Client(s)*: The AP initiates PTP-TWR message exchanges with its clients and serves as the master clock, enhancing synchronization reliability by using a dedicated UWB channel. Offering precise AP-to-client distance measurements, this scenario can support soft handover and roaming scenarios, reducing handover times by using proactive roaming methods [23]. Using multiple APs synchronized over a wired backbone, this system could provide seamless synchronization and connectivity between APs, further supporting mobility and reducing handoff delays.

2) *Synchronized UWB Infrastructure*: A synchronized UWB infrastructure involves UWB nodes exchanging PTP-TWR messages with a designated initiator node acting as the master clock. This node can be predetermined or dynamically selected through an election algorithm. Once synchronized, system clocks within the network can use this UWB infrastructure for synchronization, with the TSF clock in Openwifi acting as a slave clock. Furthermore, nodes with known positions can serve as "anchor" nodes, allowing triangulation techniques to determine the location of other nodes using distance measurements from at least three anchors [8].

3) *Wireless Ad Hoc UWB Network*: A Wireless Ad Hoc UWB Network (WANET) is scalable and self-organizing, allowing nodes to join and leave without pre-existing infrastructure [24]. UWB technology is suitable for such networks due to its reliability, high-resolution timestamps, and low complexity [25]. This scenario requires an election algorithm to designate an initiator node as the master clock, capable of handling dynamic topology changes [26]. Localization in an ad hoc setting is challenging, as anchor nodes are not predetermined. This may involve nodes advertising their ability to serve as anchors and the usage of distributed localization algorithms [27].

V. CONCLUSIONS & FUTURE WORK

This study addressed the need for precise clock synchronization in W-TSN, demonstrating that integrating UWB with W-TSN as a dedicated synchronization channel offers the potential for reliable and highly accurate, nanosecond-level clock synchronization. Leveraging UWB technology, the synchronization accuracy in Openwifi is improved through enhanced timestamp and drift correction resolutions, and the use of UWB timestamps in a multi-hop synchronization scheme. The final synchronization performance achieved an MAE of 9.98 ns with a P_{90} of 20.0 ns.

Additionally, UWB's inherent localization capabilities make it particularly suited for applications requiring both precise time synchronization and positioning. This research highlights the potential benefits and challenges associated with adopting UWB for synchronization in wireless networks, emphasizing the need for a thorough assessment to justify the added complexity and cost of its implementation for specific applications.

Future work in this research focuses on several key areas for improvement and expansion. Advanced synchronization algorithms, such as Kalman filters and enhanced drift correction methods, could improve precision and reliability. Supporting multiple responder nodes is essential for enabling localization through triangulation, which in turn paves the way for investigating integrated systems that combine synchronization and localization in practical applications. Proactive roaming strategies leveraging UWB ranging information could reduce handover times in multi-AP environments, ensuring seamless connectivity. Additionally, enabling ad hoc networking with UWB introduces challenges like robust election algorithms and distributed localization, warranting further exploration. Finally, addressing UWB's range limitations through multi-hop network configurations could extend its applicability to larger environments.

ACKNOWLEDGMENT

This work was partially supported by the imec ICON project VELOCe - (Agentschap Innoveren en Ondernemen project nr. HBC.2021.0657) and the Flemish FWO SBO S003921N VERI-END.com and FWO SBO S006224N SMART-STRIP.

REFERENCES

- [1] J. Haxhibeqiri, X. Jiao, E. Municio, J. M. Marquez-Barja, I. Moerman, and J. Hoebeke, "Bringing time-sensitive networking to wireless professional private networks: Filling gaps and bridging the innovation," vol. 121, no. 2, pp. 1255–1271, 2021.
- [2] M. Ghobakhloo, "Industry 4.0, digitization, and opportunities for sustainability," vol. 252, p. 119869.
- [3] P. Avila-Campos, J. Haxhibeqiri, X. Jiao, I. Moerman, and J. Hoebeke, "Removing the wires in time-sensitive networks," in *2022 61st FITCE International Congress Future Telecommunications: Infrastructure and Sustainability (FITCE)*, pp. 1–6.
- [4] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," vol. 11, no. 1, pp. 17–27, conference Name: IEEE Industrial Electronics Magazine.
- [5] "IEEE standard for local and metropolitan area networks—frame replication and elimination for reliability," pp. 1–102, conference Name: IEEE Std 802.1CB-2017.

- [6] M. Aslam, W. Liu, X. Jiao, J. Haxhibeqiri, G. Miranda, J. Hoebeke, J. Marquez-Barja, and I. Moerman, "Hardware efficient clock synchronization across wi-fi and ethernet-based network using PTP," vol. 18, no. 6, pp. 3808–3819, 2022, conference Name: IEEE Transactions on Industrial Informatics.
- [7] "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," pp. 1–499, conference Name: IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008).
- [8] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge University Press.
- [9] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," vol. 16, no. 5, p. 707, number: 5 Publisher: Multidisciplinary Digital Publishing Institute.
- [10] H. Xu and L. Yang, "Ultra-wideband technology: Yesterday, today, and tomorrow," in *2008 IEEE Radio and Wireless Symposium*, pp. 715–718, ISSN: 2164-2974.
- [11] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman, "openwifi: a free and open-source IEEE802.11 SDR implementation on SoC," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–2, ISSN: 2577-2465.
- [12] DWM3000EVB. Available: www.qorvo.com/products/p/dwm3000evb.
- [13] nRF52840DK. Available: www.nordicsemi.com/products/development-hardware/nRF52840-DK.
- [14] A. Mahmood, R. Exel, H. Trsek, and T. Sauter, "Clock synchronization over IEEE 802.11—a survey of methodologies and protocols," vol. 13, no. 2, pp. 907–922.
- [15] P. Chen and Z. Yang, "Understanding PTP performance in today's wi-fi networks," vol. 31, no. 6, pp. 3037–3050, conference Name: IEEE/ACM Transactions on Networking.
- [16] C. De Dominicis, A. Flammini, S. Rinaldi, E. Sisinni, A. Cazzorla, A. Moschitta, and P. Carbone, "High-precision UWB-based time-stamping," in *Control and Communication 2011 IEEE International Symposium on Precision Clock Synchronization for Measurement*, pp. 50–55, ISSN: 1949-0313.
- [17] F. M. Anwar and M. B. Srivastava, "Precision time protocol over LR-WPAN and 6lowpan," in *2017 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, pp. 1–6, ISSN: 1949-0313.
- [18] Decawave, *DW3000 Datasheet*, 2020.
- [19] [openwifi/doc/README.md](https://github.com/open-sdr/openwifi/blob/master/doc/README.md) at master · open-sdr/openwifi.
- [20] G. Hollósi and I. Moldován, "Ultra wideband-based wireless synchronization of IEEE 1588 clocks," vol. 15, no. 2, pp. 21–28.
- [21] X. Jiao, W. Liu, M. Mehari, H. Thijs, and A. Muhammad. (2023) open-source ieee802.11/wi-fi baseband chip/fpga design.
- [22] Decawave, *The implementation of two-way ranging with the DW1000*.
- [23] S. Feirer and T. Sauter, "Seamless handover in industrial WLAN using IEEE 802.11k," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1234–1239, ISSN: 2163-5145.
- [24] M. G. Rubinstein, I. M. Moraes, M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "A survey on wireless ad hoc networks," in *Mobile and Wireless Communication Networks*, G. Pujolle, Ed. Springer US, pp. 1–33.
- [25] H. H. Hoang and B. W. Kwan, "A cluster-based protocol for self-organizing UWB wireless ad hoc sensor networks," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 317–322, ISSN: 1062-922X.
- [26] S. Vasudevan, J. Kurose, and D. Towsley, "Design and analysis of a leader election algorithm for mobile ad hoc networks," in *Proceedings of the 12th IEEE International Conference on Network Protocols, 2004. ICNP 2004.*, pp. 350–360, ISSN: 1092-1648.
- [27] B. Denis, J.-B. Pierrot, and C. Abou-Rjeily, "Joint distributed synchronization and positioning in UWB ad hoc networks using TOA," vol. 54, no. 4, pp. 1896–1911, conference Name: IEEE Transactions on Microwave Theory and Techniques.