Impactless Association Methods for Wi-Fi based Time-Sensitive Networks

Pablo Avila-Campos, Jetmir Haxhibeqiri, Ingrid Moerman, Xianjun Jiao and Jeroen Hoebeke

IDLab, Department of Information Technology at Ghent University - imec, Technologiepark-Zwijnaarde 126, Ghent, 9052, Belgium.

*Corresponding author(s). E-mail(s): pabloesteban.avilacampos@ugent.be;
Contributing authors: jetmir.haxhibeqiri@ugent.be; ingrid.moerman@ugent.be; xianjun.jiao@ugent.be; jeroen.hoebeke@ugent.be;

Abstract

Current safety-critical applications, found in domains such as automotive networking, automated industries, and control systems, require deterministic and latency-bounded end-to-end communication, currently not supported by best-effort networks. Time-Sensitive Networking (TSN) is the most feasible approach to fulfill such requirements. Originating from the wired domain, their application to the wireless domain is now attracting attention. Besides some advances in time synchronization and scheduling, essential policies such as association with a wireless TSN (W-TSN), have not been explored. This paper analyses the necessary tools for a prospective client to accomplish association with the W-TSN without affecting ongoing time-sensitive traffic. We introduce three beacon-based methods to pre-synchronize and pre-schedule nodes willing to associate with the network. On top of the first full-stack Software Defined Radio (SDR) based IEEE802.11/Wi-Fi design, the proposed approaches are developed and tested in a real-world environment. A delay analysis shows a worst-case association delay of up to 1.8s, and high accuracy synchronization on prospective clients' association frame receptions, with 100% of on-time arrivals, even with challenging scheduling timeslots as short as 128 μs.

Keywords: W-TSN, IEEE 802.11, association, beacons

1 Introduction

The automation industry has continuously been looking for solutions for deterministic communications i.e., guaranteed packet delivery with bounded latency, low delay variation, and low packet loss. Such features would translate into higher flexibility, versatility, and efficiency in future industrial manufacturing. Time-Sensitive Networking (TSN) is presented as the key technology to achieve this vision. Although TSN might be thought of as a protocol, it is in fact a set of enhancements to the IEEE802.1 standard aimed at meeting the high latency and reliability needs of current and future industrial networks. Its wireless counterpart, W-TSN, despite the advantages of bringing mobility and fast deployment to TSN, is still in an early development stage [1].

TSN was born in 2007. Back then, high-end media communications were mainly based on analog point-to-point connections which often resulted in large and confusing amounts of cables. Different approaches based on computer networks such as Audio over Ethernet or Audio over IP were used to try to solve this. However, apart from the lack of interoperability, these solutions did not fulfill the required Quality of Service (QoS). As a result, the Audio Video Binding (AVB) Task Group was created to help the standardization of transmission of media content through Ethernet [2]. The AVB group tackled the problem with different standards such as IEEE 802.1AS [3], which introduces an enhanced network synchronization mechanism, IEEE 802.1Qbv [4] to schedule traffic, reserve bandwidth, and streams, and IEEE 802.1CB [5] to provide redundancy. Finally, AVB became the TSN task group [1], and the aforementioned standards are now the foundation of TSN [6].

The TSN tool set is divided into four components, the first one being reliability, which focuses on high availability using path reservation, control, and replication. Next, resource management targets TSN configuration through components such as stream reservation, link-local registration to distribute information through the network, and data modeling. The aim is to achieve zero-congestion loss, closely related to the third component bounded latency. TSN tackles this by introducing known techniques such as traffic shaping, cyclic queuing, and frame preemption. The last component is time synchronization which, as mentioned before, was already introduced by the AVB task group [7]. Together, these components pretend to cover current and next-generation application communication requirements over the Ethernet standard.

TSN standards have been developed considering Ethernet as the exclusive communication medium. Despite recent advances in wireless technologies, such as 5G and Wi-Fi 6, the industry is still skeptical that the same level of synchronization, latency, security, and reliability performance can be achieved over wireless [8]. Certainly, wireless includes extra challenges such as Radio Frequency (RF) interference due to its shared medium and mobility nature

[1] http://www.ieee802.org/1/tsn

which provokes re-transmissions and thus latency increase. Also, limited bandwidth and transmission rates in some technologies plus resource constraints with limited buffer space further aggravate the situation. Despite this, solutions are being studied to bring TSN-like capabilities to wireless technologies and efforts to enhance the reliability of wireless through network design, frequency planning, QoS, and now TSN is showing impressive results [8].

There are two leading candidates for technologies to integrate TSN. On the one hand, 5G is being proposed as a bridge for connecting TSN nodes capable of keeping the high time synchronization required as well as mapping TSN traffic into 5G QoS flows [9]. On the other hand, Wi-Fi is taking big steps towards determinism with the introduction of gigabit data rates and the addition of Orthogonal Frequency-Division Multiple Access (OFDMA) in Uplink (UL) and Downlink (DL), which reduces channel contention. However, still, the high-end required time synchronization of 1 μs is not at the wired-TSN level [10].

Apart from bringing existing TSN components to the wireless world, it is also imperative to consider primary wireless network processes such as bootstrapping. Bootstrapping refers to the process of establishing initial network connectivity, including association and configuration on a wireless device. It involves setting up the necessary parameters and protocols to enable wireless communication and access to the network. This process becomes particularly critical in a TSN environment, as the network must ensure uninterrupted traffic flow while the bootstrapping process takes place. In 802.11, prospect clients will use Clear Channel Assessment (CCA) before transmitting to avoid collisions; however, this is not fully compatible with W-TSN. As it is shown in Figure 1, the W-TSN has assigned time slots to the clients. In the case of a new client wanting to associate, two possibilities may arise. On the one hand, if Client 1 uses CCA, it will delay its transmission, increasing its latency and hence jitter. On the other hand, if Client 1 does not use CCA, a collision will take effect, thus retransmission will be required. In both cases, determinism is affected, and the network is no longer capable of providing bounded latency.

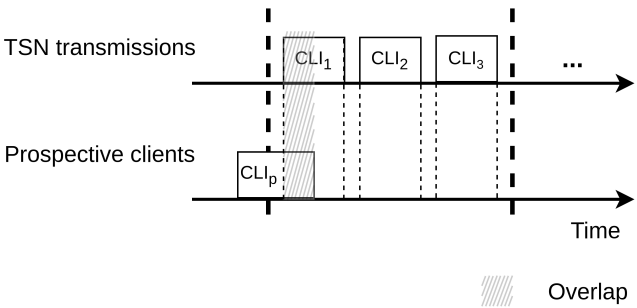


Fig. 1: Wireless association collision

As previously mentioned, achieving high reliability and bounded latency are key objectives of TSN. Therefore, the network must offer compliant solutions for fundamental processes like bootstrapping to ensure these goals are met. Our strategy to solve this problem is to grant TSN features to clients even before they achieve a connection with the network. By using beacons, prospect clients are designated a time slot and a good enough time synchronization mechanism, avoiding overlaps on other client transmissions and keeping compatibility with IEEE802.11. As an extension of our previous work in [11] and [12], two new pre-synchronization mechanisms both compatible with Access Point scheduled transmissions have been implemented. Furthermore, information in beacons is encoded to reduce its size, and an analysis of the impact of W-TSN association over ordinary association has been appointed. Also, a second prospective client has been included in the tests to clearly demonstrate the pre-schedule concept. To the best of our knowledge, we are the first to give a functional solution to the association in a Wi-Fi-based W-TSN. This solution was developed and tested on top of a TSN-enhanced version of *openwifi*, a Linux mac80211 compatible full-stack IEEE802.11/Wi-Fi design based on Software Defined Radio (SDR) [13].

The paper's specific contributions can be summarized as follows:

- We present a highly efficient association procedure that is fully compatible with TSN (Time-Sensitive Networking) and specifically designed for Wi-Fi networks.
- As part of this association procedure, we suggest utilizing beacon frames to enable scheduling and time synchronization for non-associated clients.
- To optimize functionality, we propose an efficient encoding scheme within the beacon frames, which allows prospective clients to determine their time slot for association.
- Lastly, we introduce three distinct time pre-synchronization methods that maintain time synchronization during the association procedure for prospective clients.

The remainder of the paper is organized as follows. First, in section 2, related work on Wireless TSN association is presented, followed by a background introduction to association and the main TSN features required for association in section 3. Next, in section 4, the proposed association procedure and how it integrates into the current IEEE 802.11 association process are explained. In section 5, the developed beacon-based time synchronization methods are explained. In section 6, the results including synchronization accuracy, threshold analysis, schedule correctness, and association delays are shown. Finally, section 7 concludes this work.

2 Related Work

Before getting into the W-TSN association, it is important to provide sufficient background regarding the current research in the scope of this work. Therefore, this section is divided into three parts. The first will introduce wireless broadband technologies suitable to provide TSN capabilities. Next, a review of recent W-TSN implementations is given, focused on how every work tackles association. Finally, the recent advances in synchronization and scheduling with similar approaches are reviewed.

2.1 TSN Capable Wireless Technologies

Among the various requirements a wireless technology must include to support time-sensitive traffic, are traffic differentiation, low transmission latency, high precision in time synchronization, vendor neutrality, and reliability [14]. It is also important to consider that despite meeting the previous conditions, the biggest obstacle all wireless technologies have to overcome to properly deliver time-sensitive traffic is the variant nature of the wireless medium. Nevertheless, in this context, 5G and Wi-Fi are the clear candidates for realizing W-TSN. Both technologies' standardization bodies have been continuously working towards some sort of Ultra Reliable Low Latency Communications (URLLC), directly related to the aforementioned TSN requirements.

On the 5G side, the 3GPP Release-16 introduces support to TSN applications demanding determinism and reliability. Here, for a centralized TSN controlling architecture, the 5G system becomes a transparent bridge, hiding the core and access network from the TSN. Then, control plane functions such as session management function (SMF), policy control function (PCF), and Access and Mobility Management Function (AMF), create QoS policies to allow the user traffic to be handled in a time-sensitive manner [9, 15]. Concerning the association of a new node, no big changes have been introduced in recent releases. The prospect client randomly selects a preamble to transmit association frames. The association process takes place over the Physical Random-Access Channel (PRACH), dedicated to that purpose [16]. Hence, given the frequency domain separation, the effect of association on time-sensitive traffic may be considered non-existent.

At the IEEE802.11 or Wi-Fi side, mechanisms to better support time-sensitive traffic have long been lacking. As part of the IEEE802.11e and IEEE802.11aa amendments, the inclusion of time-critical Access Categories (AC), i.e. traffic differentiation, tries to improve the QoS of streams. However, the lack of flexible scheduling policies avoids guaranteeing time-sensitive features. Furthermore, the MAC layer of IEEE802.11 has always depended on a random access method with exponential back-off, which is unable to ensure a bounded latency because of its unpredictable nature [14, 17]. Nevertheless, with the advent of IEEE802.11ax or Wi-Fi 6, new features such as the introduction of OFDMA, the low-latency operation mode, and Target Wake Time

(TWT) mechanisms provide significant latency reduction and improved efficiency [15, 18, 19]. The latest flavor, i.e. IEEE 802.11be or Wi-Fi 7, is still under definition. However, it will bring additional mechanisms such as traffic differentiation and prioritization, multi-Access Point operation, and the use of the 6 GHz band with wider 320 MHz channels to enhance transmission rates, bringing low latency and high reliability, enabling new scenarios such as TSN [20]. With respect to the association, Section 4 describes in detail the current procedure that has not changed nor been adapted to TSN requirements. As explained later, our work focuses on studying a TSN-capable association mechanism, that fills in the current gap.

2.2 W-TSN Implementations

Currently, there is an important interest in implementing TSN features in the wireless world. However, most practical implementations do not consider basic processes such as association. On the contrary, most of them focus on the wired-wireless interface, synchronization, and scheduling [21–24]. There are several studies closely related to our research. For example, in the study conducted by Cruces et al. [25], they propose a TSN that incorporates a discovery and configuration stage based on Carrier Sense Multiple Access (CSMA), followed by a switch to a Time Division Multiple Access (TDMA) system during the operational stage. However, when compared to our work, their focus primarily lies on the operational stage rather than addressing the challenge of incorporating new clients into the W-TSN. While their discovery stage provides a solution for the initial setup, it does not offer a comprehensive solution for integrating new clients who wish to join the W-TSN during an operational stage.

Superframes are also implemented by [26] in W-SHARP, a wireless technology for real-time industrial applications. In this case, communication is performed using superframes, while time synchronization is achieved with beacons at the beginning of the superframes. In this case, the stations estimate the time offset and pass it through a proportional-integral(PI) filter. Different from this work, we present three different means of synchronization that directly drop *late* beacons. This helps to avoid the typical delays imposed by PI filters and leads to a faster association process. Moreover, even though the W-SHARP work has similarities with IEEE 802.11, it is not compatible with it.

In [27], the authors propose a Time-Frequency Division Multiple Access (TFDMA) based system called RETIS which implements different techniques to improve reliability and latency in wireless transmissions. Bringing the idea from the Wireless networks for Industrial Automation-Factory Automation (WIA-FA) protocol, beacons are used for time synchronization and scheduling. The authors briefly introduce how new nodes would use the information in the beacons to become part of the network, but contrary to our work there is not a clear description of how it will handle channel access delays for time synchronization or the compatibility with IEEE802.1Qbv, which defines traffic shaping in wired TSN.

2.3 Association-related Time Synchronization and Scheduling

As aforementioned, there are big efforts in research toward precise time synchronization. Here, works such as [28, 29] and [30] present and evaluate high-accuracy alternatives focused on wired-wireless architectures in line with 802.1AS. They explore different messaging alternatives, including beacons. However, compared with our work, they still lack solutions for when the client is still not part of the network or depends on complex and high resource consumption time-compensation algorithms.

In conclusion, several studies about time synchronization and traffic scheduling, particularly in TDMA-based systems, have been presented. However, those do not take into account an association phase that must be carried out without interfering with traffic in dynamically assigned time slots.

3 Wireless Association and TSN Background

This section aims to introduce the concepts related to wireless association, and TSN features necessary for association. Initially, it will explain the current IEEE 802.11 association procedure, providing a foundation for understanding the proposed modifications to make it TSN-friendly. Time synchronization and time-aware scheduling are particularly vital in a deterministic association process [3, 31]. Therefore, this section will also explore the fundamental concepts underlying these two features in both wired and wireless communications, offering a comprehensive understanding of their significance.

3.1 Current Association Procedure

The IEEE 802.11 standard, also known as Wi-Fi, defines a set of standards related to Local Area Networks (LAN), and protocols in Media Access Control (MAC), and Physical Layer (PHY). Among the services defined in the standard, the association is of importance at the earliest stage. Each station must enforce an association procedure to join an access point before it can communicate with other nodes in the network [32].

The first step a prospective client has to complete to associate is to select a target Access Point (AP). The selection is done by scanning the available channels. These channels depend on client configuration, hardware capabilities, and country regulations. IEEE 802.11 standard currently includes two scanning options: active and passive. For active scanning on every channel, the client first transmits a probe request frame and waits for a default dwell time of 500 ms for a probe response from APs in coverage. Opposite to this, passive scanning only listens for AP beacons on every channel for the dwell time [33].

Table 1: Time Synchronization Requirements [34, 35]

Use Case	Sync. Accuracy (μs)	E2E Latency (ms)
Factory Automation	<1	<1
Smart Grid	<1-20	<10-100
Vehicular Communication	<10	<3-10
Audio/Video	<12	<150
Wireless Sensor Networks	<1	<1

Once the target AP has been selected, the association starts. As shown in Figure 2, there are three connection states in IEEE 802.11. Every state change requires a frame transmitted by the client to the AP and a response. Hence to get to State 3, the client will transmit authentication and an association request which needs to be replied to by the AP [33].

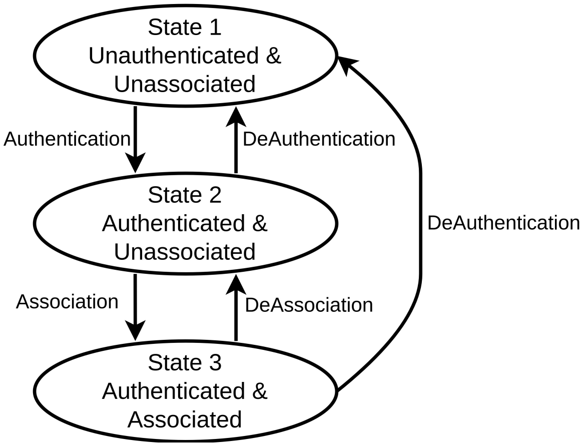


Fig. 2: Association State Machine

3.2 Time Synchronization in TSN

Time synchronization has always been a crucial aspect of communications. Currently, it is becoming more critical as emerging technologies demand more efficiency from the networks, which starts with synchronization. For instance, in Wi-Fi 7, accurate time synchronization is essential to coordinate APs to support Multi-User (MU) transmissions in UL and DL [20]. Table 1 presents the synchronization accuracy and latency needed for some popular use cases.

Time-sensitive networks are built on the fact that a common time reference is shared across the network. As a basis, IEEE 802.11AS [3] defines a profile of generalized Precision Time Protocol (gPTP), one of the most widely used protocols for TSN. The IEEE 802.1AS characteristics may be summarized into three main goals: 1) In LAN-attached station environments an accuracy level within 1 μs. 2) One-time reference for the LAN/Subnet, along a LAN-agnostic approach with segments using 802.3 Ethernet, 802.11 Wi-Fi, and other technologies with defined specific measurements for each. 3) A plug-and-play approach, with automatic Grand Master (GM) selection, and clock tree reconfiguration if GM is lost. Currently, 802.1AS objectives do not contemplate a time synchronization mechanism that does not require a link establishment between the client and the network. Thus it is crucial to present solutions that will benefit essential policies such as association.

3.3 Scheduling in TSN

The other key feature of TSN, next to synchronization, is scheduling. One of the proposed mechanisms for TSN is found in IEEE 802.1Qbv which defines a so-called Time-Aware Scheduler (TAS) [4]. The first function of the TAS is to establish equal discrete time periods called *cycle*. After that, traffic is classified into queues with assigned time slots within the *cycle*. Time-aware shaping provides predetermined time slots for various data traffic classes established in a Gate Control List (GCL). This allows for the scheduling of numerous data streams while keeping a bounded latency. Due to the requirement of the scheduler for synchronization, all network participants are aware of when which priority can be transmitted and processed. In addition to IEEE 802.1Qbv, other sub-standards have been developed or are in the process of being developed to include traffic-shaping policies and scheduling algorithms. These standards are more of a building kit than a complete solution.

Figure 3 illustrates the scheme of the scheduler described above. This scheduler plays a crucial role in orchestrating the traffic transmission from nodes by allocating different time slots, thereby avoiding interruptions. In the wireless domain, this scheduling mechanism can be likened to a TDMA approach. To establish a solid foundation for our work on Wi-Fi W-TSN and, more specifically, bootstrapping, the openwifi nodes we use, already incorporate this gating mechanism. This implementation is carried out within the FPGA, guaranteeing seamless compatibility with wired TSN standards. For further information, please refer to the following resource: [36].

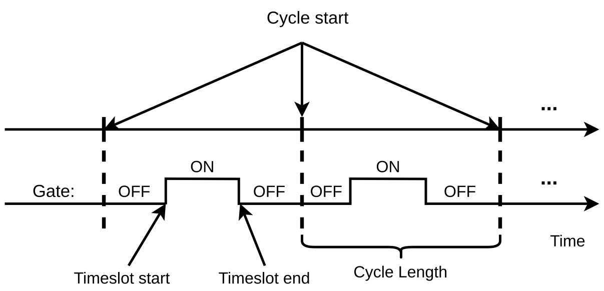


Fig. 3: Scheduling gating system

From a broader perspective, meeting application requirements necessitates the calculation of end-to-end schedules for the nodes involved. While existing research in this field primarily focuses on wired TSN, which relies on fixed schedules, the integration of wireless introduces challenges due to its inherently variable nature. On one hand, the variable transmission time resulting from channel quality-dependent data rates necessitates the utilization of adaptable time slots. On the other hand, strategies must be devised to enhance reliability while simultaneously meeting the application requirements.

4 Beacon Based W-TSN Bootstrapping

The central aspect of our proposal revolves around the utilization of beacons, which play a crucial role in introducing features such as scheduling and time synchronization to potential clients, effectively integrating them into the W-TSN even before they establish a formal association. This section aims to provide an overview of how the beacons, once received by prospective clients, seamlessly integrate with the TSN during the association process. Additionally, to ensure the proper functioning of the beacons, a management system is an integral part of our solution. The role of the controller in defining the behavior of the beacons will be discussed, followed by an outline of how beacons carry pre-synchronization and pre-schedule information, which are key elements enabling the W-TSN association for prospective clients.

4.1 IEEE 802.11 association integration to TSN

Keeping the procedure flow described in Section 3.1, our W-TSN association method integrates with IEEE 802.11 by solely using passive scanning. This way early prospect client transmissions are avoided. Then, as shown in Figure 4, both authentication and association request frames are transmitted by the prospective client in a TSN-scheduled fashion, transparent to the higher layers. To achieve this, the prospective client first needs to obtain a schedule and time synchronization from the captured beacons. As these two TSN features are used only during the association stage we will call them pre-schedule and pre-synchronization from now on.

4.2 TSN Management for Boostrapping

The IEEE 802.1Qcc standard [37] introduces three TSN management configuration models: centralized, decentralized, and hybrid. Among these models, our study focuses on the centralized model due to its numerous advantages, including simplified network management, enhanced resource allocation, flexibility, and scalability. In the centralized model, nodes within the network convey their application requirements to a Centralized User Controller (CUC). The CUC then communicates with a Centralized Network Controller (CNC) responsible for distributing the network resources among the nodes [38].

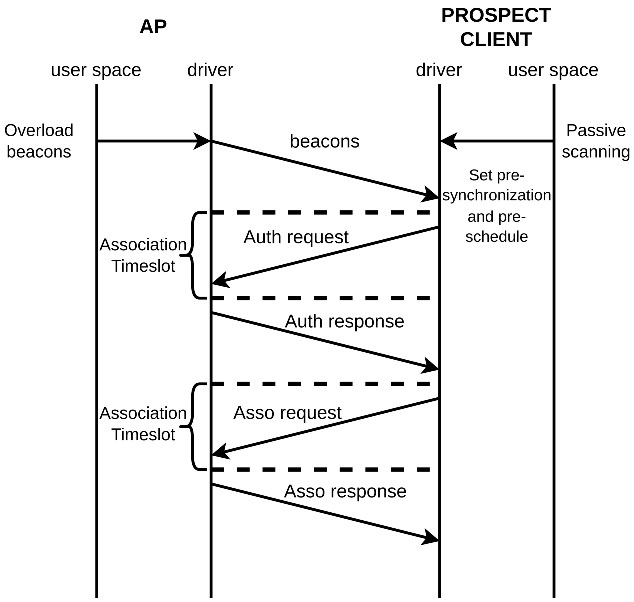


Fig. 4: Association Process

In terms of association, the CNC plays a crucial role by allocating association time slots to APs based on the available resources, as depicted in Figure 5 and implemented in this study. The association time slot is shared among all potential clients seeking to associate with an AP. Consequently, the size of the shared association time slot should be adjusted according to the anticipated number of associations. This ensures that sufficient time is allocated to accommodate the expected client associations effectively.

As part of the bootstrapping process, once the new client has been associated with the W-TSN AP, the network will follow the typical process, providing an IP address and an enhanced time synchronization protocol such as gPTP, described in Section 3.2, will take over, ceasing pre-synchronization [39]. Lastly, the client and the network will negotiate a schedule depending on the application's needs and available resources.

4.3 Beacon overloading and encoding

Beacons are one of the management frames of IEEE 802.11. The AP periodically transmits beacons to announce its presence and help associated clients with time synchronization. Beacons are the only frames that are received by prospective clients and could be used to trigger an association. As shown in Figure 6, among other information, beacons carry the AP timestamp, which in a TSN describes the network time. They also contain the beacon interval, typically configured to 100 Time Units (TU) with a TU being equal to 1.024

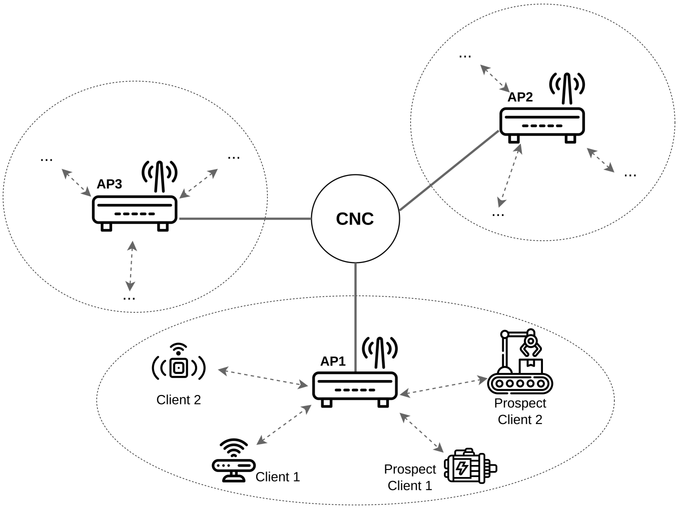


Fig. 5: Association Management Architecture

ms, so a beacon is expected to be transmitted every 102.4 ms. Further, capability information brings information about the type of network, support of features, and more. Finally, up to 2320 bytes of information elements can be included, part of which we will make use of [33].

As previously stated, the association requires two actors. Pre-synchronization and a pre-schedule or association time slot. Once the AP is provided with the pre-schedule, this is broadcasted to prospective clients in the vendor-element of beacons. The pre-schedule is described in four fields of the vendor element: cycle length index (j), time slot length index (k), time slot bitmap index start (w_n), and time slot bitmap index end (w_m) as seen in Figure 6.

Typically, manufacturers utilize the vendor element in beacons to incorporate customized information or facilitate unique functionalities within their devices. Consequently, the proposed modification to include schedule information aligns seamlessly with the standard specifications, ensuring compatibility across systems.

To reduce the number of occupied bits in the vendor-element, the pre-schedule is encoded using the following nomenclature: j and k, each with 3 bits, represent the cycle and time slot length respectively as seen in Equation 1. Then, the cycle length may vary from 512 to 65536 μs and the time slot duration from 128 to 16384 μs as can be seen in Equations 2 and 3, respectively:

$$S = j[3bits] + k[3bits] + w_n[9bits] + w_m[9bits] \tag{1}$$

$$C_L = 512 * 2^j \ [\mu s] \tag{2}$$

where

$$j = 0, .., 7$$

Table 2: Pre-schedule Encoding Notation

Variable	Description
S	Pre-synchronization schedule carried by the beacon
j	Encoded cycle length value, where j = 0,..,7
k	Encoded time slot length value, where k = 0,..,7
w_n	Time slot position index start, where $0 \leq w_n$ $0 \leq (4 * 2^{j-k} - 1)$
w_m	Time slot position index end, where $0 \leq w_m$ $0 \leq (4 * 2^{j-k} - 1)$
C_L	Decoded cycle length value, where C_L = 512,..,65536μs
TS_L	Decoded time slot length value, where C_L = 128,..,16384μs
TS_{Start}	Decoded time slot start value
TS_{End}	Decoded time slot end value

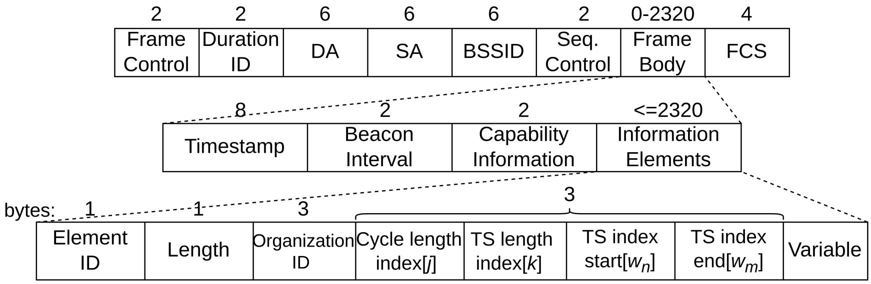


Fig. 6: Beacon Stuffing in Vendor Proprietary Element

$$TS_L = 128 * 2^k \ [\mu s] \tag{3}$$

where

$$k = 0, .., 7$$

with

$$TS_L \leq C_L$$

Next, the time slot position index start and end, are encoded as w_m and w_n with 9 bits each. Both indexes mark the beginning and the end of the period during which association is allowed as seen in Figure 7. As described by Equation 4, the start and end index may range from 0 to $4 * 2^{(j-k)} - 1$.

$$0 \leq w_m \leq w_n \leq 4 * 2^{(j-k)} - 1 \tag{4}$$

where

$$k \leq j + 2$$

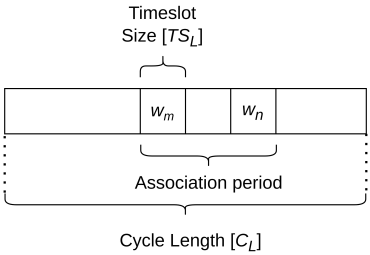


Fig. 7: Bitmap Encoding

Finally, Equations 5 and 6 define the decoding operation done in the prospect client. Once w_m and w_n are received, the prospective client calculates on a time basis the transmission gate opening and closing as previously presented in Figure 3. The local time is ruled by the Time Synchronization Function (TSF), which is a timer with modulus 2^{64} counting in increments of microseconds.

$$TS_{Start} = TS_L * w_n \ [\mu s] \tag{5}$$

$$TS_{End} = TS_L * (w_m + 1) \ [\mu s] \tag{6}$$

5 Pre-synchronization methods

Providing accurate time synchronization between the network and prospective clients is required to ensure the correct usage of the association schedule. As this time synchronization is accomplished during association, it is referred to as pre-synchronization. To accomplish pre-synchronization, three different methods have been fashioned. They can be divided into two groups. The first considers that the AP beacon transmissions are not scheduled, hence, a new beacon will be generated according to a fixed interval, as described in subsection 4.3, and broadcasted from the AP to stations in coverage. On the other hand, the second considers a full W-TSN scenario where beacons, as any other DL traffic is scheduled, following the mechanism described in subsection 3.3.

The first method is called *Early/Late Beacon Detection (ELBD)* and it is part of the first group. The second group comprises the other two methods: *Slice-Based Cycle Detection (SBCD)* and *Follow-up based*. By different techniques, all methods attain time synchronization with prospective clients, making TSN association feasible.

As seen in Equation 7, beacon delay might be divided into a fixed delay (δ), and a variable delay (X). Beacons function as management frames and thus are exempt from the backoff mechanism. Instead, the variable delay (X) in beacon transmission primarily occurs due to Clear Channel Assessment

(CCA). This delay occurs when there is an ongoing concurrent transmission, causing the beacon transmission to be postponed. In the subsequent proposed time synchronization methods, efforts are made to either filter or compensate for this variable delay. On the other hand, the fixed delay (δ) is a distinct aspect addressed in detail in the following section.

$$Delay = \delta + X \tag{7}$$

Beacon delay compensation

The received beacon presents fixed delays that the prospective client must adjust to achieve a correct pre-synchronization. Such delays are caused by channel access, beacon transmission time, and processing between the driver level and the physical layer at the transmitter side and vice-versa at the receiver side. As can be seen in Equation 8, δ is the overall corrected delay that is added to the received beacon time information at the driver level of the receiver side before setting it to the prospective client.

$$\begin{aligned} \delta &= DIFS + B_T + P \\ &= 28 + \frac{148 * 8}{6} + 19.7 \\ &= 28 + 197.33 + 19.7 \\ &= 245.03\ \mu s \end{aligned} \tag{8}$$

The DCF interframe space (DIFS) is the initial delay element, and it provides a technique for assigning priorities to frames in the distributed coordination function [33] of IEEE 802.11. It is the least amount of time a beacon frame must wait before being transmitted, and it varies depending on the standard used. The 802.11g is used in our implementation, therefore $DIFS$ takes 28 μs according to [40]. The beacon transmission time is B_T, and it is calculated using the beacon frame size of 148 bytes and the default data rate of 6 Mbps at which beacons are transmitted. Finally, the client processing time P was monitored and taken into account while processing and filtering beacon frames at the driver.

5.1 Early/late beacon detection (ELBD)

Due to the variable nature of the wireless channel, and the Clear Channel Assessment (CCA) mechanism that must be followed by the IEEE802.11 nodes, the beacon arrival time at the prospective clients is not always consistent. As a result, if the Timestamp field shown in Figure 6 is directly overwritten into the prospective client, this could result in synchronization inaccuracies and thus unsynchronized client transmissions.

The purpose of *ELBD* is to eliminate ineffective beacons. Relating to Equation 7, this method filters beacons affected by the variable delay X. As shown in Figure 9a, this filter uses the arrival time difference($\Delta_{arrival}$) of a pair of received beacons at the prospect's client driver, and checks if the difference is within a defined range (R) in a moving window manner. To define R, the received beacon frame's *Beacon Interval - bcn.int* field is extracted, and an *error* deviation value is added, as indicated in Equation 9. If the received beacon pair matches the criteria, the client's timestamp is set using the last beacon's internal timestamp. A deep explanation of how the *error* value is calculated may be found in Section 6.2.

$$R = bcn.int \pm error \tag{9}$$

This filter, unlike previous filtering approaches found in literature [26], does not require a convergence period, which is critical for a quick W-TSN association operation. The algorithm 1 shows the operation of the filter. The term TSF stands for Time Synchronization Function, which represents the timestamp. First, when a beacon is received, the current TSF is saved to calculate $\Delta_{arrival}$. If the $\Delta_{arrival}$ condition of two continuous beacon arrivals is met, the client's time is updated adding the delay compensation δ described in Equation 8, and lastly the $gate.flag$ will allow association packets to be transmitted.

5.2 Slice-based Cycle Detection (SBCD)

Besides channel access delays, having a schedule at the AP means that beacons will not follow a default transmission interval (102.4 ms), but instead will rely on the schedule as shown in Figure 9b. The *SBCD* mechanism benefits from the cyclic operation of the schedule to achieve pre-synchronization.

To operate, apart from the pre-schedule, the variable part of the beacon's vendor element shown in Figure 6 carries the AP schedule. This way the prospective client knows when the beacon is expected to arrive. As it is not possible to predict how long the beacon had to wait before being transmitted, X in Equation 7, in this method, the timestamp in the beacon is not used. Instead, the client compares the $\Delta_{arrival}$ with the *ap.cycle* of the AP schedule.

This comparison is performed by calculating the module operation between the $\Delta_{arrival}$ and *ap.cycle*, from two consecutive arrivals, as shown in Equation 10. In this manner, the prospect client filters beacons that are not being transmitted at the beginning of the AP time slot. Subsequently, once a pair of beacons meets the condition, as shown in the Algorithm 2, the client timestamp is set to 0 plus $ap.ts_{start}$, and the delay compensation described in Equation 8 is also added. $ap.ts_{start}$ is different from 0 when the AP time slot did not start with the cycle.

$$\Delta_{mod} = mod(\Delta_{arrival}, ap.cycle) \tag{10}$$

5.3 Follow-up based

The last pre-synchronization mechanism can be defined as the most generic as it supports non-scheduled and scheduled beacons. It exploits the fact that once a frame is transmitted, the physical layer notifies the driver and higher layers. The follow-up method timestamps every beacon transmission, overcoming the channel delay issue we described previously in Section 5.1, and loads the next beacon with this timestamp information ($bcn.tx.TSF$).

In contrast to previous pre-synchronization methods, this approach takes a different approach by compensating for the variable delay X from Equation 7. Instead of filtering out ineffective beacons, this method utilizes pairs of beacons, where one beacon carries information about the variable delay X of the previous beacon.

As shown in Algorithm 3, the first condition to calculate the current time is to confirm a pair of beacons are consecutive. For this, Δ_{tx} is compared with $bcn.int$, which comes as a beacon field. Then, as seen in Figure 8, the $corrected.TSF$ can be calculated using Equation 11.

$$\begin{aligned} corrected.TSF = {} & \Delta_{arrival} + bcn.tx.TSF[i-1] + \delta \\ & - DIFS \end{aligned} \tag{11}$$

Equation 11, calculates the correct time by adding the $\Delta_{arrival}$, the transmission time, which is part of δ, and the timestamp of the moment the previous beacon was transmitted. $DIFS$ is already considered in the $bcn.tx.TSF$ hence needs to be subtracted from Equation 11.

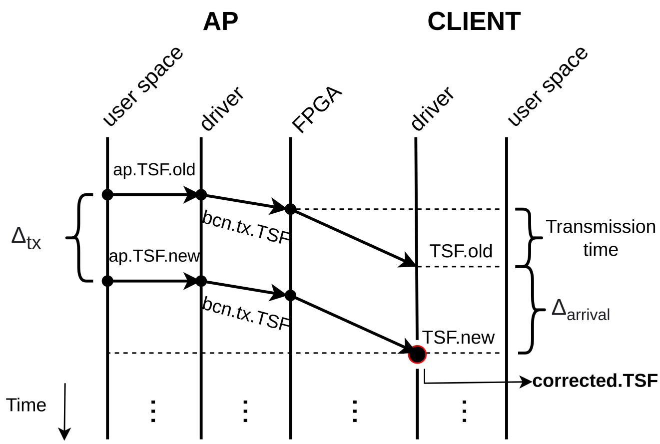
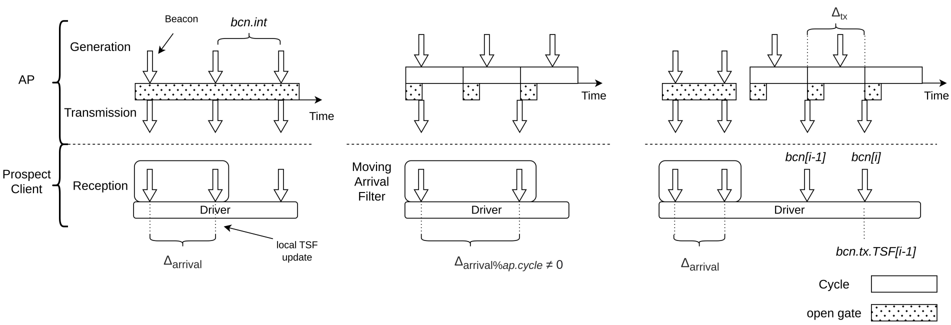


Fig. 8: Follow-up pre-synchronization timeline

After calculating the $corrected.TSF$, it is then set to the prospective client. This update in time is also applied to previous time information, hence more beacons can be used for pre-synchronization. A timeline diagram of the correction equation is shown in Figure 8. As it is shown, beacons are generated at the AP user space, transmitted from the FPGA, and finally processed at the client's driver.

Figure 9c, presents the advantage of the follow-up method over the previous ones as it can work for both scheduled and unscheduled AP beacon transmissions. This Figure also illustrates Δ_{tx} and $\Delta_{arrival}$ in the generation and reception of beacons.



(a) ELBD (b) SBCD (c) Follow-up

Fig. 9: Pre-synchronization methods

Algorithm 1 ELBD pre-synchronization

Input: new received beacon, *error*
Output: pre-synchronization

Initialization :
Extract $ap.TSF$, $bcn.int$ ▷ A beacon is received, and fields are extracted.
Save $TSF.now$ ▷ The beacon arrival TSF is saved.
$\Delta_{arrival} = TSF.now - TSF.old$ ▷ The TSF difference between two beacon arrivals is calculated.
Moving Arrival Filter
if $bcn.int - error \leq \Delta_{arrival} \leq bcn.int + error$ **then** ▷ Check if $\Delta_{arrival}$ is within the filter limits.
    $client.TSF = ap.TSF + \delta$ ▷ The client TSF is updated.
    $gate.flag = 1$ ▷ The gate flag indicates the client was successfully pre-synchronized.
end if
$TSF.old = TSF.new$ **return** $client.TSF, gate.flag$

Algorithm 2 SBCD pre-synchronization

Input: new received beacon, *error*
Output: pre-synchronization

```
Initialization :
Extract ap.TSF, ap.cycle, ap.ts_start        ▷ A beacon is received, and fields are extracted.
Save TSF.now                                 ▷ The beacon arrival TSF is saved.
Δ_arrival = TSF.now − TSF.old                ▷ The TSF difference between two beacon arrivals is calculated.
Δ_mod = mod(Δ_arrival, ap.cycle)             ▷ Δ_arrival is compared with the AP cycle using the module operation.
Moving Arrival Filter
if 0 ≤ Δ_mod ≤ error then                    ▷ Check if Δ_arrival and AP cycle are multiples.
    client.TSF = 0 + ap.ts_start + δ         ▷ The client TSF is updated considering the AP time slot position.
    gate.flag = 1                            ▷ The gate flag indicates the client was successfully pre-synchronized.
end if
TSF.old = TSF.new return client.TSF, gate.flag
```

5.4 Implementation and Standard Compliance

The bootstrapping implementation covers mainly the driver of both the AP and prospective client. Table 3 provides a summary of the necessary changes in the driver for pre-synchronization and pre-scheduling. Notably, only the Follow-up method requires a driver enhancement at the AP which enriches the beacons with $bcn.tx.TSF[i-1]$ as detailed in Section 5.3 and Algorithm 3. Regarding driver adjustments to the prospective STA, each method necessitates distinct approaches. These varied approaches have been elaborated upon in the corresponding sections, namely 4 and 5. As shown in Section 4.1, the association procedure remains transparent to higher layers. Furthermore, it is important to emphasize that both W-TSN and the proposed bootstrapping procedure are fully compliant with standards. However, it is worth noting that integrating TSN with legacy devices could potentially impact TSN's performance, as discussed in the Introduction of this document.

6 Results

6.1 Setup description

The specified association mechanism has been implemented on an FPGA-based SDR platform utilizing IEEE 802.11. The openwifi IEEE802.11/Wi-Fi baseband chip/FPGA design was utilized for evaluating the proposed solution, which was implemented in the Software Defined Radio (SDR) ADRV9361-Z7035, which combines the Analog Devices AD9361 integrated RF Agile

Algorithm 3 Follow-up pre-synchronization

Input: new received beacon,
Output: pre-synchronization

Initialization :
Extract $ap.TSF.now$, $bcn.int$, $bcn.tx.TSF[i-1]$ ▷ A beacon is received, and fields are extracted.
Save $TSF.now$ ▷ The beacon arrival TSF is saved.
$\Delta_{arrival} = TSF.now - TSF.old$ ▷ The TSF difference between two beacon arrivals is calculated.
$\Delta_{tx} = ap.TSF.now - ap.TSF.old$ ▷ The TSF difference between two beacon transmissions is calculated.
Moving Arrival Filter
if $\Delta_{tx} \leq 1.5 * bcn.int$ **then** ▷ Check if the last two received beacons are consecutive.
  Save $TSF.current$ ▷ To track current client TSF.
  $corrected.TSF = \Delta_{arrival} + bcn.tx.TSF[i-1] + \delta - DIFS$ ▷ Corrected TSF as in Eq. 11.
  $TSF.old = TSF.now + corrected.TSF - TSF.current$ ▷ Calculate the corrected old TSF at client.
  $client.TSF = corrected.TSF$ ▷ The client TSF is updated.
  $gate.flag = 1$ ▷ The gate flag indicates the client was successfully pre-synchronized.
else
  $TSF.old = TSF.now$
end if
$ap.TSF.old = ap.TSF.new$ **return** $client.TSF, gate.flag$

Table 3: Bootstrapping Implementation

		AP	**Prospective STA**
Pre-scheduling		No modifications	Requires driver modifications
Pre-synchronization	**ELBD**	No modifications	
	SBCD	No modifications	
	Follow-up	Requires driver modifications	

Transceiver™with the Xilinx Z7035 Zynq-7000. The carrier card ADRV1CRR-BOB/FMC was used to add an Ethernet interface [13].

Despite the 802.11g standard being employed for developing and testing, the association procedure has not been modified in newer iterations of Wi-Fi such as 802.11ax or 802.11be, hence the proposed association technique is still valid [36]. Even though using 802.11ax will also provide more transmission degrees of freedom when using OFDMA, during the association phase, however, resource use is constrained to the time domain. Apart from accurate time synchronization, UL OFDMA requires AP trigger frames. Hence for the association, the AP would need to realize whether there are one or more prospect stations willing to associate in order to trigger their transmissions in different OFDMA sub-carriers, which is far from ideal and might delay the process.

As shown in Figure 10, the infrastructure mode wireless network comprised two nodes: one access point (AP) and two clients/stations, all of which were connected to a PC functioning as a CNC and a Dashboard/MQTT Broker.

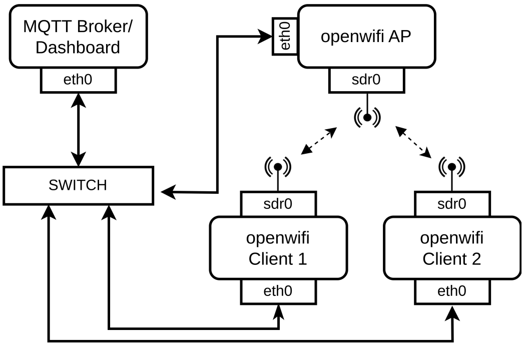


Fig. 10: Setup Architecture

All the experiments presented next were performed in an office environment with neighbor channels in use and a fixed distance of 10cm between nodes. Also, a signal level lower than -50 dBm was ensured in all nodes. Despite the short testing distance and good signal level, as long as the nodes are in coverage, no pre-synchronization nor pre-scheduling will be altered as propagation delay is negligible for this procedure.

6.2 Threshold definition

In both *ELBD* and *SBCD*, algorithms an *error* value is included in the main time update condition. This variable allows controlling the beacon filtering used for pre-synchronization. Thus, this *error* value becomes a threshold allowing or avoiding pre-synchronization with the received beacons.

To define the optimal threshold value, two variables have been considered. First, the association delay is a key performance indicator of how transparent the W-TSN association process is to higher layers. In [32], an in-depth study on the delays in setting up an IEEE802.11 link and how it affects user experience can be found. In that work, the association process (association+authentication) results in about 35% of the total set-up time which can take up to 30s, but in most cases takes 7s. Hence, 2.45s will be the baseline for our measurements.

The association delay values shown next, are measured from the reception of the first beacon until the association (association+authentication) is successful. Every value is the result of 20 successful association processes. Furthermore, a worst-case fixed association time slot of 128 μs within a cycle of 65535 ms was loaded in beacons. Also, in the $SBCD$ case, the AP was set to the same worst-case schedule to restrict beacon transmissions.

Tables 4 and 5, show different threshold levels as well as the median delay caused by association. The number of beacons (modal value) needed to complete a first pre-synchronization is also shown.

Table 4: Association delays with ELBD

Threshold(us)	Association Delay(s)	Beacons
2	1.284	3
3	1.159	5
5	1.029	3
10	1.039	3
20	1.039	3
50	1.038	4
100	1.037	3

The second variable to contemplate is the percentage of beacons used. Although this will highly depend on the channel access delays, it is helpful to infer the filtering means of the algorithm.

Looking into Figures 11 and 12, for both methods, as was expected, as the threshold level increases, the percentage of used beacons does as well. However, in the case of $SBCD$, the increase is sharper than in $ELBD$. Furthermore, as described in section 5, it is not possible to compare both approaches as they are based on different concepts.

Regarding the follow-up method, two tests were performed, one with and one without the worst-case schedule at the AP. The median association delays were 1.8 and 1.75 seconds, respectively. Additionally, the mode beacon count was 3 for both cases.

In conclusion, the introduction of pre-synchronization with worst-case schedules results in association delays of up to 1.284s for $ELBD$, 1.048s for $SBCD$, and 1.8s for the follow-up approach. These values remain lower than the 2.45s reported by [32]. This clearly demonstrates that the inclusion

Table 5: Association delays with SBCD

Threshold(us)	Association Delay(s)	Beacons
0	1.015	4
1	1.048	3
2	1.015	3
10	0.983	3

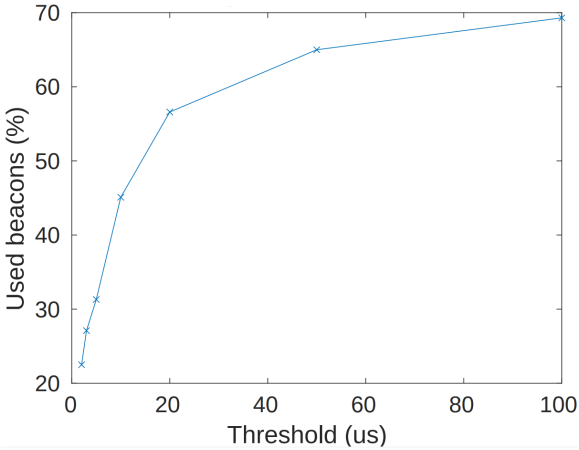


Fig. 11: Used Beacons Percentage in ELBD

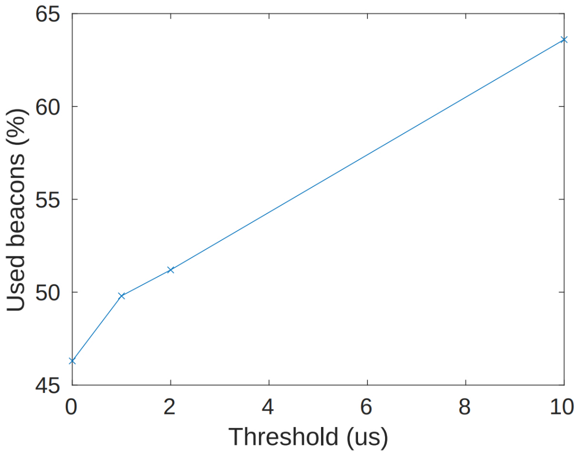


Fig. 12: Used Beacons Percentage in SBCD

of pre-synchronization does not significantly delay the standard association process.

Additionally, it is important to note that both pre-synchronization methods, $ELBD$, and $SBCD$, allow for the definition of a threshold level that can be tailored to the percentage of used beacons in a given application. Although the number of beacons in this case is similar, resulting in a similar association delay, the threshold represents a trade-off between pre-synchronization accuracy and association delay. This trade-off depends on specific variables such as beacon generation frequency, AP schedule, and channel congestion.

Lastly, it is crucial to take into account scenarios in which multiple stations need to utilize the association time slot. Contention between stations might be another association delay source. However, an appropriate selection of the association time slot size and strategies such as grouping, used in IEEE 802.11ah will diminish such delays [41].

Furthermore, it is crucial to highlight the correlation between the possibility of simultaneous connections and the interaction between the W-TSN schedule and the starting timing of the prospective client association procedure. Particularly noteworthy is the impact within shorter cycles and the consideration of the minimal packet prerequisites for association, which would lead to a reduction in the probability of concurrent connections.

6.3 Pre-synchronization

There are different methods to measure time synchronization accuracy. However, even though a driver-level implementation is done, due to the limitations of beacons, time synchronization accuracy is limited. Thus, the aim of pre-synchronization is not high accuracy, but instead a good enough synchronization at small time slot sizes of 128 μs. Furthermore, the intention behind offering multiple pre-synchronization methods is to provide users with the flexibility to choose the approach that best suits their needs and aligns with the specific characteristics of their network environment.

Hence, to evaluate how good pre-synchronization is, the arrival time of ping request frames of similar size and the data rate of association frames were used. To elaborate, by using the architecture of Figure 10, once both clients were pre-synchronized, they transmitted ping request frames within the association time slot, which were logged at the AP. Equation 12 was used to analyze the arrival TSF of such frames into a cycle-based measurement.

$$T_{Of} = mod(T_{Ar}, C_L) \tag{12}$$

where T_{Of} represents the arrival time within the cycle in μs, T_{Ar} the frame arrival TSF, C_L the designated cycle length, and $mod()$ is the modulo operation. Once calculated, T_{Of} is pushed to the MQTT broker/Dashboard, which saves and presents the frame arrivals.

Figure 13, presents both clients' arrival offset to the AP using follow-up-based pre-synchronization. The horizontal red lines limit the time slots of 128 μs. The AP updates in real-time the pre-schedule, which then needs to be adopted by the prospective clients as well. As can be seen, both client receptions are always within the assigned time slot limits.

6.4 Pre-schedule

The correct behavior of the pre-scheduling mechanism is highly dependent on pre-synchronization. The time reference acquired by the time synchronization helps the prospective client to define the beginning of a cycle and thus, the opening and closing of the gates using the received pre-schedule. Figure

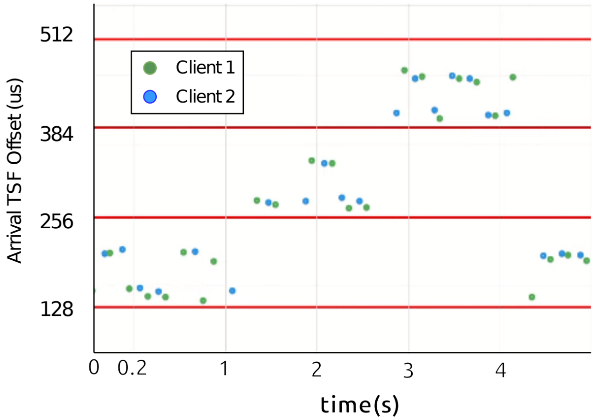


Fig. 13: Arrival offset changing pre-schedule

14 presents four different experiments where the prospective client changed its pre-schedule using the *ELBD* pre-synchronization method. This Figure illustrates the arrival TSF at the AP across different association time slots, highlighting the pre-scheduling capabilities for adaptation. Each point on the graph corresponds to a specific TSF arrival time, demonstrating the ability to dynamically adjust and adapt within different association time slots.

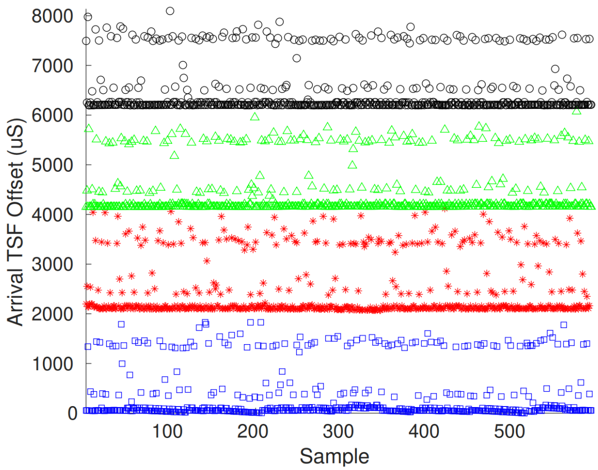


Fig. 14: Arrival offset in different association timeslots

7 Conclusion and Future Work

By developing three pre-synchronization techniques to support the wireless association of prospective TSN clients, this study established that it is possible to achieve impactless association by providing TSN features to unconnected clients. Pre-scheduling is accomplished by loading AP broadcast beacons with the association time slot offset in time. The aforementioned pre-synchronization techniques grant the prospective client a time reference to transmit the association frames in the assigned time slots. The first pre-synchronization mechanism considers the AP transmitting beacons at a fixed interval, while the second and third include a schedule at the AP. For both, the prospective client had to compensate for delays related to contention, processing, and time slot access. The third pre-synchronization method outperforms the previous ones as it operates in a scheduled and non-scheduled scenario. Future work will investigate association time slot sizes that adapt to the different working stages of the network, changing channel conditions, as well as priority designation mechanisms. Further studies are required to support needed features such as impactless roaming between APs and optimization of the association process to achieve lower delays.

Acknowledgment

This research was funded by the Flemish FWO SBO S003921N VERI-END.com (Verifiable and elastic end-to-end communication infrastructures for private professional environments) project and by the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" program.

Declarations

- Funding
  This study was funded by the Flemish FWO SBO S003921N VERI-END.com (Verifiable and elastic end-to-end communication infrastructures for private professional environments) project and by the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" program.
- Conflict of interest/Competing interests
  Author Pablo Avila-Campos declares that he has no conflict of interest. Author Jetmir Haxhibeqiri declares that he has no conflict of interest. Author Ingrid Moerman declares that she has no conflict of interest. Author Xianjun Jiao declares that he has no conflict of interest. Author Jeroen Hoebeke declares that he has no conflict of interest.
- Ethics approval
  This article does not contain any studies with human participants or animals performed by any of the authors.
- Consent to participate

Not applicable
- Consent for publication
  Not applicable
- Availability of data and materials
  The datasets generated during and/or analyzed during the current study are not publicly available due to a confidentiality agreement but are available from the corresponding author upon reasonable request.
- Code availability
  Not applicable
- Authors' contributions
  Not applicable

References

[1] Cisco: White paper: time-sensitive networking: a technical introduction. Cisco Public (2017)

[2] 802.1BA-2011 IEEE standard for local and metropolitan area networks-audio video bridging (AVB) systems, 31 (2011)

[3] 802.1AS-2020 - IEEE Standard for Local and Metropolitan Area Networks-Timing and Synchronization for Time-Sensitive Applications. Technical report (2020)

[4] 802.1Q-2018 IEEE standard for local and metropolitan area network-bridges and bridged networks (2018)

[5] 802.1CB-2017 - IEEE standard for local and metropolitan area networks–frame replication and elimination for reliability (2017)

[6] Lim, H.-T., Herrscher, D., Waltl, M.J., Chaari, F.: Performance analysis of the ieee 802.1 ethernet audio/video bridging standard. SIMUTOOLS '12, pp. 27–36. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL (2012)

[7] Farkas, J., Finn, N., Ericsson, P.T., Broadcom, H.: IEEE 802.1 Time-Sensitive Networking (TSN). Technical report (2017)

[8] Bush, S.: Avnu Alliance® White paper wireless TSN-definitions, use cases & standards roadmap (2020)

[9] 5G-ACIA: Integration of 5G with time-sensitive networking for industrial communications. Technical report, 5G Alliance for Connected Industries and Automation (5G-ACIA) (2021)

[10] Cisco: IEEE 802.11ax: The sixth generation of Wi-Fi. Technical report (2020)

[11] Avila-Campos, P., Haxhibeqiri, J., Moerman, I., Hoebeke, J.: Impactless beacon-based wireless TSN association procedure. 2022 IEEE 18th International Conference on Factory Communication Systems (WFCS), 1–8 (2022). https://doi.org/10.1109/WFCS53837.2022.9779186

[12] Avila-Campos, P., Haxhibeqiri, J., Moerman, I., Hoebeke, J.: Beacon-based wireless TSN association. IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 1–2 (2022). https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798105

[13] Jiao, X., Liu, W., Mehari, M., Aslam, M., Moerman, I.: Openwifi: A Free and Open-Source IEEE802.11 SDR Implementation on SoC. IEEE Vehicular Technology Conference **2020-May** (2020). https://doi.org/10.1109/VTC2020-SPRING48590.2020.9128614

[14] Kinabo, A.B.D., Mwangama, J.B., Lysko, A.A.: An evaluation of broadband technologies from an industrial time-sensitive networking perspective. 2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021, 715–722 (2021). https://doi.org/10.1109/ICACCS51430.2021.9441748

[15] Atiq, M.K., Muzaffar, R., Seijo, O., Val, I., Bernhard, H.P.: When IEEE 802.11 and 5G meet time-sensitive networking. IEEE Open Journal of the Industrial Electronics Society **3**(January), 14–36 (2022). https://doi.org/10.1109/OJIES.2021.3135524

[16] ETSI: LTE; Evolved universal terrestrial radio access (E-UTRA); Physical channels and modulation (3GPP TS 36.211 version 12.5.0 Release 12) (2015)

[17] Adame, T., Carrascosa-Zamacois, M., Bellalta, B.: Time-sensitive Networking in IEEE 802.11be: On the Way to Low-Latency WiFi 7. Sensors (Basel, Switzerland) **21**(15) (2021). https://doi.org/10.3390/S21154954

[18] Kronauer, G., Seewald, M., Smith, M., Regev, A.: Avnu Alliance® White paper in wireless TSN : market expectations, capabilities & certification (February) (2022)

[19] Cavalcanti, D.: White paper: wireless TSN – definitions , use cases & standards roadmap. Avnu Alliance, 1–16 (2020)

[20] Adame, T., Carrascosa-Zamacois, M., Bellalta, B.: Time-sensitive networking in ieee 802.11be: On the way to low-latency wifi 7. Sensors **21**(15) (2021). https://doi.org/10.3390/s21154954

[21] Sudhakaran, S., Montgomery, K., Kashef, M., Cavalcanti, D., Candell,

R.: Wireless time sensitive networking for industrial collaborative robotic workcells. IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS **2021-June**, 91–94 (2021). https://doi.org/10.1109/WFCS46889.2021.9483447

[22] Sudhakaran, S., Mageshkumar, V., Baxi, A., Cavalcanti, D.: Enabling qos for collaborative robotics applications with wireless TSN. 2021 IEEE International Conference on Communications Workshops, ICC Workshops 2021 - Proceedings, 0–5 (2021). https://doi.org/10.1109/ICCWorkshops50388.2021.9473897

[23] Seijo, O., Iturbe, X., Val, I.: SHARP: Implementation of a hybrid wired-wireless TSN network to enable flexible smart factories. IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS **2021-June**, 95–98 (2021). https://doi.org/10.1109/WFCS46889.2021.9483597

[24] Seijo, O., Iturbe, X., Val, I.: Tackling the challenges of the integration of wired and wireless TSN with a technology proof-of-concept. IEEE Transactions on Industrial Informatics **3203**(c), 1–1 (2021). https://doi.org/10.1109/tii.2021.3131865

[25] Cruces, C., Torrego, R., Arriola, A., Val, I.: Deterministic hybrid architecture with time-sensitive network and wireless capabilities. IEEE International Conference on Emerging Technologies and Factory Automation, ETFA **2018-Septe**, 1119–1122 (2018). https://doi.org/10.1109/ETFA.2018.8502524

[26] Seijo, O., Lopez-Fernandez, J.A., Val, I.: W-SHARP: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications. IEEE Transactions on Industrial Informatics **17**(5), 3651–3662 (2021). https://doi.org/10.1109/TII.2020.3007323

[27] Cecil, R., Setka, V., Tolar, D., Sikora, A.: RETIS-Real-Time sensitive wireless communication solution for industrial control applications. IDAACS-SWS 2020 - 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, Proceedings (2020). https://doi.org/10.1109/IDAACS-SWS50031.2020.9297074

[28] Baniabdelghany, H., Obermaisser, R., Khalifeh, A.: Extended synchronization protocol based on IEEE802.1AS for improved precision in dynamic and asymmetric TSN hybrid networks. 2020 9th Mediterranean Conference on Embedded Computing, MECO 2020, 8–11 (2020). https://doi.org/10.1109/MECO49872.2020.9134100

[29] Val, I., Seijo, O., Torrego, R., Astarloa, A.: IEEE 802.1AS Clock Synchronization Performance Evaluation of an Integrated Wired-Wireless TSN Architecture. IEEE Transactions on Industrial Informatics **18**(5), 2986–2999 (2022). https://doi.org/10.1109/TII.2021.3106568

[30] Romanov, A.M., Gringoli, F., Sikora, A.: A precise synchronization method for future wireless TSN networks. IEEE Transactions on Industrial Informatics **17**(5), 3682–3692 (2021). https://doi.org/10.1109/TII.2020.3017016

[31] IEEE 802.1Qav-2009 - IEEE standard for local and metropolitan area networks– virtual bridged local area networks - forwarding and queuing enhancements for time-sensitive streams. Technical report (2010)

[32] Pei, C., Wang, Z., Zhao, Y., Wang, Z., Meng, Y., Pei, D., Peng, Y., Tang, W., Qu, X.: Why It Takes so Long to Connect to a WiFi Access Point. Proceedings - IEEE INFOCOM (2017). https://doi.org/10.1109/INFOCOM.2017.8057164

[33] Coleman, D.D., Westcott, D.A.: CWNA: Certified Wireless Network Administrator Official Study Guide, 4th edn. SYBEX Inc., USA (2021)

[34] Mahmood, A., Ashraf, M.I., Gidlund, M., Torsner, J., Sachs, J.: Time synchronization in 5G wireless edge: requirements and solutions for critical-MTC. IEEE Communications Magazine **57**(12), 45–51 (2019). https://doi.org/10.1109/MCOM.001.1900379

[35] Gundall, M., Huber, C., Melnyk, S.: Integration of ieee 802.1as-based time synchronization in ieee 802.11 as an enabler for novel industrial use cases. ArXiv **abs/2101.02434** (2021)

[36] open-sdr/openwifi: open-source IEEE 802.11 WiFi baseband FPGA (chip) design: driver, software. https://github.com/open-sdr/openwifi

[37] IEEE P802.1Qcc - IEEE draft standard for local and metropolitan area networks–media access control (MAC) bridges and virtual bridged local area networks amendment: stream reservation protocol (SRP) enhancements and performance improvements, 1–236 (2018)

[38] Nsiah, K.A., Alkhouri, K., Sikora, A.: Configuration of wireless TSN networks. IDAACS-SWS 2020 - 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, Proceedings, 3–7 (2020). https://doi.org/10.1109/IDAACS-SWS50031.2020.9297066

[39] Aslam, M., Liu, W., Jiao, X., Haxhibeqiri, J., Miranda, G., Hoebeke, J., Marquez-Barja, J., Moerman, I.: Hardware efficient clock synchronization

across Wi-Fi and ethernet-based network using PTP. IEEE Transactions on Industrial Informatics **18**(6), 3808–3819 (2022). https://doi.org/10.1109/TII.2021.3120005

[40] 802.11 IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan networks - specific requirements - Part 11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Std 802.11b-1999, 1–96 (2000)

[41] Seferagic, A., Moerman, I., De Poorter, E., Hoebeke, J.: Evaluating the suitability of IEEE 802.11ah for low-latency time-critical control loops. IEEE Internet Things J. **6**(5), 7839–7848 (2019). https://doi.org/10.1109/JIOT.2019.2916579