Wireless Time Synchronization in IEEE 802.11

Ulkar Aslanova, Leander Seidlitz*, Jonas Andre*
*Chair of Network Architectures and Services
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: ulkar.aslanova@tum.de, {leander.seidlitz, jonas.andre}@net.in.tum.de

***Abstract*—IEEE 802.11 Wireless Local Area Networks (WLANs) are extensively utilized for communication among multiple devices within a limited geographical area. In comparison to Ethernet-based networks, WLANs provide users with mobility and flexibility. However, wireless communication technologies have introduced new challenges, one of which is time synchronization, which is essential for network management and monitoring. While the synchronization issue has been extensively researched in conventional wired networks, the physical limitations of the wireless medium have presented a new set of difficulties. The lasting problem with the lack of synchronized timing support has been addressed in IEEE 802.11-2012. This amendment introduced two methods: Timing Advertisement (TA) and Timing Measurement (TM). This paper analyzes the time synchronization mechanisms included in the IEEE 802.11 standard, along with exploring non-IEEE solutions. It also examines the various factors that influence synchronization performance.**

***Index Terms*—IEEE 802.11 WLAN, Time Synchronization, Timing Synchronization Function (TSF), Timing Advertisement (TA), Timing Measurement (TM), Network Time Protocol (NTP), IEEE 1588 Precision Time Protocol (PTP)**

1. Introduction

Wireless communication has brought major changes to data networking and telecommunications. In today's world, wireless networks are widely employed in fields that formerly relied on traditional wired networks. One such network is the Wireless Local Area Network (WLAN), which offers users high bandwidth connectivity within a limited geographical area [1]. The adoption of IEEE 802.11 standards has had a significant influence on the both public and private domains. 802.11 has developed into a common option for a constantly growing application field due to its cost-effective chipsets and support for high data rates [2]. In modern times, WLANs are utilized to provide a communication infrastructure for a wide range of applications, spanning from small-scale in-home networks to large-scale deployments in office buildings, as well as mobile networks in airports and other public spaces.

1.1. Infrastructure Mode in IEEE 802.11

In IEEE 802.11, there are several communication modes defined for the transmission of data between devices. These communication modes control the interaction and information exchange among devices within the network. The basic service set (BSS) is a fundamental component of IEEE 802.11 WLAN. It is a group of wireless stations (STAs) that communicate at the physical layer (PHY). Depending on the communication mode, BSS can be classified into three categories: independent BSS (IBSS), infrastructure BSS and Mesh BSS (MBSS). This paper will focus on investigating time synchronization techniques designed for the infrastructure mode in IEEE 802.11-based networks.

The infrastructure mode of WLAN enables communication between STAs through a centralized entity called an access point (AP). The AP manages the network and coordinates communication between devices. This mode is particularly suitable for use in centralized architectures utilized in various applications, such as smart grids [3] and industrial automation [4].

1.2. Time Synchronization Problem

In an ideal clock, the rate remains constant over time, while an ordinary clock develops an offset. So, if an ordinary clock C_{ord} has a rate of a and an offset b to the ideal clock t, then C_{ord} can be calculated using (1).

$$C_{ord}(t) = a \cdot t + b \qquad (1)$$

The goal of time synchronization is to minimize the error ϵ (2) between two clocks.

$$\epsilon = C_{ord}(t) - t \qquad (2)$$

Time synchronization among wireless nodes is one of the key functions for controlling and monitoring activities within wireless networks. It is essential not only for network management but also for MAC layer protocols such as Time Division Multiple Access (TDMA), which relies on synchronized timing for achieving collision-free channel access in shared medium networks. Moreover, time synchronization is essential for power management in IEEE 802.11 networks [5], real-time applications [6], and Internet of Things (IoT) applications [7]. In IEEE 802.11, proper time synchronization enables power-saving features, allowing devices to coordinate wake-sleep schedules efficiently. For real-time applications, accurate time synchronization is crucial to minimize jitter and latency. In IoT applications, time synchronization is fundamental for synchronizing data from multiple sensors and devices.

Despite the fact that synchronizing wireless nodes is essential, there has not been much support for providing synchronized clocks until the introduction of the IEEE 802.11-2012 amendment. Prior to this amendment, the

 doi: 10.2313/NET-2023-11-1_12

preferred approach to achieve synchronized clocks in IEEE 802.11 networks was the utilization of synchronization protocols such as the Network Time Protocol (NTP) and the IEEE 1588 Precision Time Protocol (PTP) over WLAN. The 802.11-2012 standard has expanded the methods available for clock synchronization in wireless LANs with two mechanisms: Timing Advertisement (TA) and Timing Measurement (TM).

The remaining sections of the paper are structured as follows: Section 2 examines alternative clock synchronization protocols that are not part of the IEEE 802.11 standard; Section 3 provides an overview of the clock synchronization mechanisms included within the IEEE 802.11-2012 standard; and Section 4 analyzes the performance aspects of wireless time synchronization.

2. Non-IEEE 802.11 Protocols for Time Synchronization

Due to the lack of time synchronization support in IEEE 802.11, other protocols such as IEEE 1588 Precision Time Protocol (PTP) and Network Time Protocol (NTP) have been used for synchronization. These protocols were originally designed for synchronization purposes in wired networks and later adapted for wireless networks.

Both of these protocols can be utilized for relative and absolute time synchronization. The goal of absolute synchronization, also known as external synchronization, is to align devices within a network to a universal reference such as International Atomic Time (TAI) or Universal Coordinated Time (UTC). This ensures that all devices maintain a consistent time with respect to the specified reference. Relative synchronization (or internal synchronization) is employed solely to establish a shared timebase among synchronized devices within a network.

2.1. PTP over WLAN

PTP, introduced in the IEEE 1588 standard, is commonly used in LAN networks. It is based on a master-slave approach, where the master node is responsible for synchronizing slaves in the network. In PTP, the default Best Master Clock Algorithm (BMCA) is used. In infrastructure mode WLAN, the AP can be considered as the master and other STAs as slaves. Therefore, a custom BMCA can be implemented, which will choose AP as the master clock [8]. Several ways have been researched in order to employ PTP over WLAN. The prototypes using software and hardware-based timestamping were first presented in [9].

PTP uses a two-way packet exchange for synchronization. The master clock periodically sends synchronization messages (SYNC) every two seconds (by default), which contain an estimated timestamp of the message transmission time t_1. Upon receiving the SYNC message, the slave stores a timestamp of the reception time t_2. The master clock can also send a follow-up message with a more precise value of the transmission time timestamp. The difference $t_2 - t_1$ could be used for calculating the offset; however, it includes not only the offset from slave to master o_{sm}, but also the propagation delay d_{ms} from master to slave (3).

$$t_2 = t_1 + d_{ms} + o_{sm} \tag{3}$$

For this reason, the slave clock periodically sends a delay request (DELAY_REQ) message to the master clock and records the transmission time timestamp t_3. In response to the request, the master clock sends a delay response (DELAY_RESP) message containing the reception time of the received request message t_4 [9]. The difference t_4 - t_3 includes the propagation delay d_{sm} from slave to master (4).

$$t_4 = t_3 + d_{sm} - o_{sm} \tag{4}$$

With this information, the slave can calculate the offset to the master o_{sm} using (5), and adjust its clock accordingly [8].

$$o_{sm} = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} - \frac{d_{ms} - d_{sm}}{2} \tag{5}$$

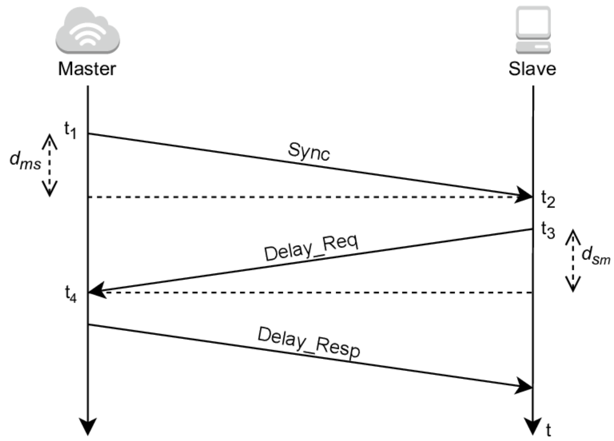

Figure 1: Message exchange in PTP [10];

If the propagation delays d_{ms} and d_{sm} are equal, then it is possible to calculate the offset precisely. PTP assumes that the propagation delay is symmetric in both directions, but this assumption is not always true and can result in synchronization bias. As shown by Mahmood et al. in [11], in wireless networks, asymmetry comes from the multicasting of packets from STA to AP, and the propagation delay in the direction from slave to master is commonly greater than from master to slave.

One of the main problems that arises when using PTP over WLAN is the handover of STAs from one AP to another, which is one of the requirements for use in large industrial environments where mobility is important. As PTP is designed for wired networks, it does not provide a fast handover of slaves from one master to another. A broader discussion of this problem has been presented in [11].

2.2. NTP over WLAN

NTP is a client-server protocol which can be used for time synchronization over WLAN. In client-server synchronization clients should request synchronization from the server. In a WLAN setup, the AP can be seen as a server, while the STAs function as clients. NTP uses a two-way packet exchange and can calculate the offset using (5). Similar to PTP, NTP also makes the assumption of symmetric propagation delays.

 doi: 10.2313/NET-2023-11-1_12

Timestamping in NTP can be implemented using both software-based and hardware-based approaches. In the case of software timestamping, the performance depends on where the timestamping is done. When timestamping is done at the application level, the timestamping jitter can increase due to random channel access delays. Timestamping can also be done in the device driver to avoid channel access delays. Using hardware timestamping can provide more accurate and reliable timestamps [8].

3. Synchronization over IEEE 802.11

Various synchronization methods included in IEEE 802.11 are shown in Figure 2.

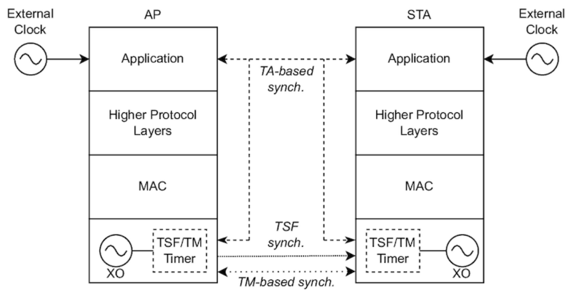

Figure 2: Synchronization Schemes in IEEE 802.11 [7];

3.1. Relative Synchronization Methods in IEEE 802.11

As all STAs communicate through the AP in infrastructure mode, it is essential that their clocks are synchronized with the AP. For internal synchronization, IEEE 802.11 uses the Timing Synchronization Function (TSF) timer with modulus 2^{64}, counting in increments of microseconds, which is present within every STA. In order to synchronize other STAs in a BSS, AP sends special frames called Beacon frames. These frames contain the TSF timer of the AP. STAs receive Beacon Frames at a regular rate, with the interval determined by their dot11BeaconPeriod parameter. The value of this parameter is included in Beacon frames and is defined by the AP. When a STA joins the BSS, it should adjust its beacon period to match the one specified in the Beacon frame [12]. As soon as a STA receives the Beacon frame, it should accept the timing information contained within this frame. If the STA's TSF timer differs from the received timestamp, it should update its local timer by adding the delay introduced by the STA's local PHY components and the time elapsed since the first bit of the timestamp was received at the MAC/PHY interface to the received timestamp.

It should be noted that the TSF method applies only offset correction and does not include rate correction. Additionally, it does not estimate propagation delay when calculating the offset. Both of these can lead to synchronization bias. However, the accuracy requirement for the TSF timer in IEEE 802.11 is achievable even without performing propagation delay compensation [8]. As the TSF timer's accuracy shall be within a tolerance of ±0.01%, it can still be met using hardware timestamps from TSF timers.

3.2. Absolute Synchronization Methods in IEEE 802.11

Alongside relative synchronization, IEEE 802.11-2012 has introduced two external synchronization mechanisms.

3.2.1. Timing Measurement (TM) Method. The TM method employs a two-way packet exchange between STA and AP for end-to-end synchronization. To initiate the Timing Measurement Procedure, the STA sends a request to the AP in the form of an Action frame, with the trigger value set to 1. Then, the AP starts transmitting action frames. The initial frame is transmitted by the AP at time t_1 and received by the STA at time t_2. Upon receiving this frame, the STA promptly sends an acknowledgement frame at time t_3. The AP receives this frame at time t_4 and responds with an action frame containing the values t_1 and t_4.

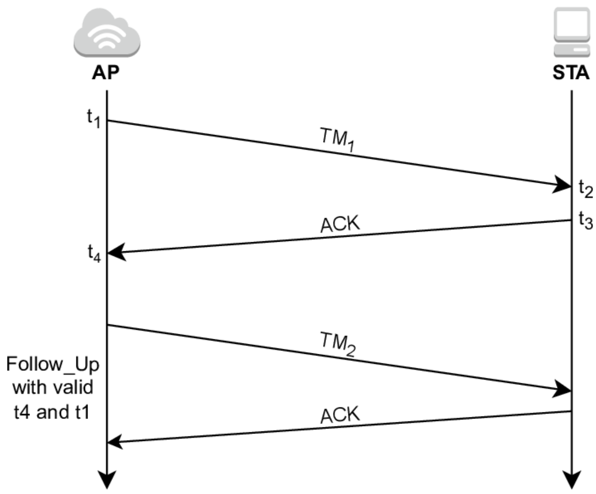

Figure 3: Message exchange in TM method [13];

With these values, the STA can calculate its time offset O from the AP using (5). Now, using the computed offset and the time reference provided by the AP, the STA can synchronize its own clock with the reference clock. To stop the timing measurement procedure, the STA should send a new action frame with the trigger value set to 0 [12].

In this method, the timestamping timer has a resolution of 10 ns, which is different from the TSF timer used in IEEE 802.11 networks, with a resolution of 1 µs. The IEEE 802.11 standard does not provide specific details regarding the timer used for timestamping in the TM method, so it's assumed that this timer is implemented as a vendor-specific timer and is capable of carrying TAI or UTC time [14].

3.2.2. Timing Advertisement (TA) Method. The TA method uses the TSF timer of the AP along with external timing standards such as UTC or TAI to achieve synchronized time across the STAs within a BSS. The AP shares a timestamp from its TSF timer and the offset between the TSF and the local clock (system clock). When a STA receives a frame containing TA information, it generates a timestamp using its TSF timer. Then, it passes this timestamp and the received TA information to higher layers in order to synchronize its local clock with the AP's local clock.

Since the local timer and the TSF timer of the AP operate on different oscillators, the offset between them can change due to the skew between the two oscillators, which can affect synchronization. Also similarly to internal synchronization with TSF, the TA method can also lead to synchronization bias as it does not include the propagation delay between the sender and receiver. In [15] Mahmood et al. performed a performance analysis of the TA method.

4. Factors Influencing Synchronization Performance

Time synchronization performance is affected by several factors, such as the oscillator, quality of timestamps, clock adjustments, and synchronization rate. In this section, the effects of the mentioned factors will be discussed.

4.1. Oscillator Impact

A clock consists of an oscillator, which serves as the fundamental source of clock ticks. The main time source in modern-day devices is a quartz crystal oscillator (XO). In an ideal clock, the frequency of the oscillator remains constant over time. However, XOs frequency stability and accuracy are affected by various physical and electrical factors such as temperature, voltage, noise, and other conditions [16]. In order to achieve high accuracy and precision, it is important to minimize the error originating from the oscillator.

4.2. Timestamping Accuracy

In packet-based networks like IEEE 802.11, timestamps can be included in packets to distribute time information throughout the network. An important factor is when and how the timestamps are drawn. To ensure accurate offset calculation and avoid asymmetry, the timestamps should be drawn based on a common reference point, for example, upon detecting the start of the frame delimiter or packet preamble [8]. There are two methods for drawing timestamps: hardware-based and software-based. Hardware-based timestamps are generated by the hardware at the physical (PHY) layer and are known for their high accuracy. The TSF and TM methods utilize hardware timestamps. In contrast, software timestamps, typically generated within the device's operating system (OS) at higher protocol layers, are generally less accurate as they do not provide the exact departure and arrival times of packets. When using software timestamping, it is crucial to employ a stable clock in order to minimize the variation in access time when deriving timestamps. One important aspect is determining the location for software timestamping. As Mahmood et al. mentioned in [8], the device driver is often the initial point where software timestamping can be implemented. The timestamp can be captured within the interrupt service routine (ISR) in the driver, as this is the earliest point in time when the operating system (OS) is notified of an incoming or outgoing packet. The TA method is one of the software-based approaches for time synchronization. Table I in [7] presents the timestamping type and achievable accuracy for each of the mentioned methods.

4.3. Clock Adjustment

The clock adjustment is performed when a STA synchronizes its local clock with the reference clock, which is the AP in infrastructure mode. This adjustment aims to minimize the offset between the STA's clock and the reference clock. Various approaches can be employed for this process, including linear least-squares regression, statistical methods, and control theory-based approaches. Depending on the chosen method, an adaptive methodology should be implemented to account for the variations in error sources [8].

4.4. Synchronization Rate

The synchronization rate determines the frequency at which synchronization packets are exchanged between the AP and the STAs. The goal of selecting an appropriate synchronization rate is to minimize the combined effect of errors introduced by the oscillator, propagation delays, and other factors. Hardware timestamps, when used with a higher synchronization rate, can lead to higher accuracy in time synchronization as they rely on hardware components that provide precise and consistent timing measurements. On the contrary, software timestamps may not effectively improve accuracy with higher synchronization rates, as the software will frequently collect timestamps, including those with potential noise and jitter. Bringing noisy timestamps into the controller more frequently can result in inaccurate time measurements and introduce instability in the synchronization process [13]. However, it's important to note that lower synchronization rates can result in larger deviations from the reference clock, so finding the right balance is essential.

5. Conclusion and Future Outlook

This paper analyzed different time synchronization methods for the IEEE 802.11 infrastructure mode. This work presented non-IEEE protocols such as PTP and NTP and examined their applications in wireless networks. However, it is important to note that these are not the only non-IEEE solutions available. There are some other custom designed protocols. For example, in [15], a new protocol called SyncTSF was introduced, which provides relative time synchronization. Additionally, this paper discussed the new methods introduced in the IEEE 802.11-2012 amendment. The utilization of the TSF scheme in ad-hoc and mesh BSS modes can be found in [7]. Furthermore, different factors that have an impact on time synchronization have been presented in this work. It is essential to consider all of the factors introduced in order to achieve high performance.

Future work may include the development of protocols that can handle propagation delays. Also, it should be noted that fault tolerance and robustness should be considered in future development.

References

[1] B. Crow, I. Widjaja, J. Kim, and P. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997.

[2] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, 2010.

[3] P. P. Parikh, M. G. Kanabar, and T. S. Sidhu, "Opportunities and challenges of wireless communication technologies for smart grid applications," in *IEEE PES General Meeting*. IEEE, 2010, pp. 1–7.

[4] S. Ivanov, E. Nett, and S. Schemmer, "Automatic WLAN localization for industrial automation," in *2008 IEEE International Workshop on Factory Communication Systems*. IEEE, 2008, pp. 93–96.

[5] J. Singh, "PERFORMING CLOCK SYNCHRONIZATION FOR POWER MANAGEMENT IN MULTI-HOP AD HOC NETWORKS," *International Journal on Intelligent Electronic Systems*, vol. 3, 01 2009.

[6] M. Mock, R. Frings, E. Nett, and S. Trikaliotis, "Continuous clock synchronization in wireless real-time applications," in *Proceedings 19th IEEE Symposium on Reliable Distributed Systems SRDS-2000*, 2000, pp. 125–132.

[7] A. Mahmood, T. Sauter, H. Trsek, and R. Exel, "Methods and performance aspects for wireless clock synchronization in IEEE 802.11 for the IoT," in *2016 IEEE World Conference on Factory Communication Systems (WFCS)*. IEEE, 2016, pp. 1–4.

[8] A. Mahmood, R. Exel, H. Trsek, and T. Sauter, "Clock Synchronization Over IEEE 802.11—A Survey of Methodologies and Protocols," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 907–922, 2017.

[9] J. Kannisto, T. Vanhatupa, M. Hannikainen, and T. Hamalainen, "Software and hardware prototypes of the IEEE 1588 precision time protocol on wireless LAN," in *2005 14th IEEE Workshop on Local & Metropolitan Area Networks*. IEEE, 2005, pp. 6 pp.–6.

[10] A. Mahmood, R. Exel, and T. Sauter, "Delay and Jitter Characterization for Software-Based Clock Synchronization Over WLAN Using PTP," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1198–1206, 2014.

[11] A. Mahmood, G. Gaderer, H. Trsek, S. Schwalowsky, and N. Kerö, "Towards high accuracy in IEEE 802.11 based clock synchronization using PTP," in *2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE, 2011, pp. 13–18.

[12] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999*, pp. 1–1076, 2007.

[13] A. Mahmood, R. Exel, and T. Sauter, "Impact of hard-and software timestamping on clock synchronization performance over IEEE 802.11," in *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*. IEEE, 2014, pp. 1–8.

[14] J. Henry, "Indoor Location : study on the IEEE 802.11 Fine Timing Measurement standard," Ph.D. dissertation, Ecole nationale supérieure Mines-Télécom Atlantique, 2021.

[15] A. Mahmood, R. Exel, and T. Sauter, "Performance of IEEE 802.11's Timing Advertisement Against SyncTSF for Wireless Clock Synchronization," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 370–379, 2017.

[16] H. Zhou, C. Nicholls, T. Kunz, and H. Schwartz, "Frequency Accuracy & Stability Dependencies of Crystal Oscillators," 2023.