Exploring the Impact of Wi-Fi on PTP Synchronization: Leveraging Wi-Fi Features to Enhance Clock Synchronization

Ecem Zehra Büyük, Filip Rezabek*, Leander Seidlitz*
*Chair of Network Architectures and Services
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: ecem.bueyuek@tum.de, frezabek@net.in.tum.de, seidlitz@net.in.tum.de

***Abstract*—Integrating Precision Time Protocol (PTP) with WiFi technology holds the potential to significantly enhance clock synchronization accuracy in wireless networks. This paper thus explores the feasibility and advantages of this integration, particularly in light of possible implementations of software and hardware timestamping in wireless networks that could lower the margin of error for timestamping in general. Furthermore, incorporating features of Time Sensitive Networking (TSN) into WiFi, we aim to balance the flexibility of wireless connections with the stability and low latency traditionally associated with Ethernet. Our findings suggest that realizing a synergy between PTP and WiFi can provide Ethernet-like latency, revolutionizing real-time applications and offering unprecedented reliability and performance. This synergy could lead to more efficient and better synchronized network systems, meeting the growing demand for precise time synchronization.**

***Index Terms*—PTP, TSN, wireless networks, clock synchronization, timestamping**

1. Introduction

As technology continues to permeate every aspect of human life, the importance of seamless communication between devices has never been greater. Consequently, the quest for precise time synchronization has emerged as a critical and highly discussed topic. Amongst various methods available, this paper focuses on the Precision Time Protocol (PTP) and how it might be impacted by Wifi. Furthermore, this paper also discusses whether features of Wi-Fi can be leveraged to minimize the impact on clock synchronisation.

The Precision Time Protocol is a message-based time transfer protocol that enables synchronization accuracy and precision in the submicrosecond range for packet-based network systems [1]. Because of its low latency, this protocol finds use in various time sensitive areas, such as telecommunications and the energy sector.

Time Sensitive Networking is another service of networking, which values time synchronization, high availability and bounded low latency through an Ethernet connection [2], utilizing PTP. With the cost of latency, WiFi introduces a more flexible, mobile and less complex networking. Unlike TSN, WiFi utilizes wireless connection instead of having a physical Ethernet connection [3]. Trying to minimize this compromise in latency by combining features of TSN within a WiFi implementation can significantly enhance the performance and availability of network systems whilst also remaining relatively uncluttered. Thus this paper focuses on if and how a synergy between Wifi and PTP be realised in order to achieve Ethernet-like latency in a wireless network.

2. Background

This section explains various methods for clock synchronization in PTP, followed by background information about NTP and the Wifi Standard.

2.1. Precision Time Protocol

The Precision Time Protocol (PTP), Figure 1, is designed to provide highly accurate time synchronization for packet-based network systems, achieving precision down to the submicrosecond level. Introduced in the IEEE 1588 standard [4], it operates by exchanging timing messages between network devices, thereby ensuring that all devices maintain a consistent and precise time reference across the network. A PTP packet is composed of a PTP daemon and a lower part, which timestamps the packets [5]. Synchronization is achieved through syncing the slave clock (secondary clock) to the master clock (primary clock) [5]. The accuracy of the synchronization is measured by calculating the difference between the time held on the master node and slave node. PTP offers two delay calculation modes, Peer to Peer and End to End.

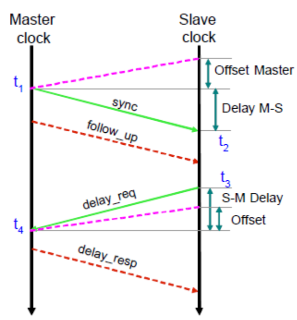


Figure 1: PTP protocol

In Peer to Peer (P2P) delay calculation mode the network operates in a manner where every participant or peer holds capabilities and duties. This means that each peer can both act as a client and server, allowing for direct communication and resource sharing between

 doi: 10.2313/NET-2024-09-1_09

them without the need of a central server. This is achieved through establishing a distributed system with each node running the same software [6].

End to End (E2E) delay calculation mode refers to a method of communication or data transfer where the information is encrypted at the senders' end and can only be decrypted by the intended recipient at the other end. This ensures that the information remains confidential and secure throughout the transmission, and can only be accesssed by the sender and the recipient.

Both P2P and E2E involve direct communication between participants, P2P refers to the network architecture, while E2E refers to the security and privacy measures applied to data transmissions. Through the P2P mode, a more accurate link delay measurement can be achieved, which then also leads to a better clock synchronization. Although for P2P to be effective, it requires all network devices to be PTP capable, since P2P views each participant as a peer with identical abilities and responsibilities. On the other hand, E2E supports non PTP devices, but in return has a worse clock synchronization performance when it comes to larger network scales [4].

In addition to different delay calculation modes, clocks are also utilized in the PTP standard to guarantee a synchronized and precise transmission of data. PTP offers three clock variations to choose from, the Ordinary Clock, the Boundary Clock and the Transparent Clock.

Ordinary Clock is the simplest version of available clocks and only has one port. That port is either used as a slave or master. In comparison to OC, the Boundary Clock is relatively more complex, possessing two or more ports, which are all in the master state with a single exception being in the slave state, which is then used to synchronize the internal clock within the BC. Thus BC is also considered to be a complete PTP node, allowing its synchronized inner clock to be used by other applications in the PTP topology. This is the main difference between BC and the Transparency Clock. Instead of synchronizing its inner clock, TC forwards the PTP message and adjusts a time correction field in the PTP message according to the residence time in the TC [7]. Transfer Clock feature is used by bridges or routers to assist clocks in measuring and adjusting for packet delay. It computes the variable delay as the PTP packets pass through the switch on the router. Any of these Clocks can be the Grandmaster Clock, which is then used by the network as the main source of time and is used as a reference for other clocks to synchronize their times with.

In the Linux ecosystem, there exists an implementation of the Precision Time Protocol *linuxptp*, a design according to the IEEE 1588 standard. This software can be used to configure PTP service on a system. *Linuxptp* consists of *ptp4l* and *phc2sys*. *Ptp4l* is used for the implementation of PTP, specifically for the OC and BC. Meanwhile *phc2sys* is used for synchronizing two clocks, the PTP hardware Clock (PHC) and system clock, as its name suggests [8]. Depending on the timestamping version, the implementations of this software vary. If hardware timestamping is being used, *ptp4l* is utilized to adjust PHC whilst *phc2sys* adjusts the system clock. If the system opts for software timestamping, then *ptp4l* directly adjusts the system clock and *phc2sys* is not needed [8].

2.2. Network Time Protocol

PTP's predecessor, the Network Time Protocol (NTP), was developed and released in 1985 and is used to organize and maintain a set of time servers and transmission paths as a synchronization subset [9]. With NTP, a precision within the millisecond range is possible [9]. Synchronization of the clocks follow a hierarchical structure, in which clocks near the top are considered more accurate than the ones near the bottom. Clients then take these more accurate clocks as reference to synchronize their time [10]. Due to its simpler build, NTP has become a central protocol for many applications requiring time synchronization over the internet and is still used for applications that do not demand a higher level of precision. However its milisecond precision is not precise enough for modern applications demanding higher accuracy. This limitation paved the way to the development of PTP, achieving a higher precision range. As real time applications continue to evolve, the enhanced precision of PTP becomes even more essential.

2.3. the Wi-Fi Standard

Wireless Fidelity, short for Wi-Fi, is a wireless transmission of radio signals and acts essentially as an alternative for Ethernet for network connectivity in modern systems [11]. WiFi was released in the 1990s with IEEE 802.11 standard [12]. Free of the limitations of a cable, WiFi offers an extended reach to places previously unavailable to a cabled connection. Without such need for a cable infrastructure, WiFi offers a lower cost in comparison to Ethernet while simultaniously enhancing mobility. Nevertheless this flexibility of WiFi comes at the expense of latency and inconsistency [13]. In contrast to stable connection of a wired network, mobility, signal strength and neighboring interference render wireless networks unpredictable [5]. As a synchronization option in wireless networks, IEEE 802.11 introduces the Time Synchronization Function (TSF). This mechanism harmonizes clients with the time broadcasted in the AP's beacons [5]. The problem with this method is that it only works well within the range of one AP, but mobile devices might move across larger areas. To able to synchronize wireless clients streching across large areas, PTP is utilized.

WiFi primarily functions within three frequency bands: 2.4 GHz (802.11), 5 GHz (802.11ac), and the more recent 6 GHz (802.11ax). Newer versions offer more channels and higher data rates with higher speed whilst also lowering latency and solving interference problems the 2.4 GHz band had. However, 5 GHz and 6 GHz implementations have a reduced range [14].

3. Analysis

This section explores the distinctions between PTP and NTP through a comparative analysis and explores the potential integration of precision timing technologies with wireless networks.

3.1. Comparison between NTP and PTP

Although both NTP and PTP provide time synchronization over a packet based network, in analysing the qualities NTP and PTP provide, it becomes evident that both protocols differ in terms of accuracy, topology, hardware and thus may benefit from different applications.

NTP achieves millisecond to sub-millisecond accuracy whereas PTP excels with sub-microsecond precision. Furthermore NTP employs a client-server hierarchical topology, in contrast to PTP, which adopts a peer-to-peer architecture, eliminating traditional hierarchies. While NTP operates efficiently with standard Ethernet hardware, PTP requires specialized equipment to reach its superior accuracy levels.

3.2. PTP with WiFi

Numerous applications rely on precise timing for the exchange of sensor data and control signals. Failure to meet these deadlines can lead to operational issues, instability, and safety risks. Due to PTPs significant improvements in terms of latency compared to its predecessor, a fusion of PTP and Wifi became a very prominent research topic. Wifi's adaptability, when combined with the precision PTP provides, presents potential benefits and improvements in terms of speed and accuracy for wireless local area networks (WLAN). The problem with combining PTP with Wireless Fidelity manifests itself within the uncertainties in PTP timestamps [5]. The uncertainties stem mainly from fluctuating delays, in data packets, signal interference and the intrinsic characteristics of communication all of which can impact the accuracy that PTP strives to deliver. As explained in *Section 2*, PTP is a protocol based on wired connections, mainly Ethernet, whilst WiFi aims to achieve a more flexible and lower-cost connection network sacrificing better synchronization a wired connection brings. Despite these hurdles, continuous research and enhancements in WiFi technology in combination with software and hardware based timestamping in PTP, and pushing TSN towards wireless networks strive to mitigating these issues and achieve highly accurate synchronization.

4. Integrating PTP and WiFi

Even though originally designed for wired LANs, there are several implementations of PTP for wireless networks. In contrast to wired networks, wireless channels introduce uncertainties in PTP timestamps. Recent work to overcome these instabilities involve timestamping and developing a wireless TSN variant, e.g. WTSN.

4.1. Timestamping

One workaround to synchronize wireless devices through PTP is to use timestamping (TS).

Figure 2 assumes Boundary Clock as the default clock method, which means synchronization transpires in several steps. The System Clock of the AP is designated as the master clock, which other clocks synchronize themselves to. The system clocks of the clients are regarded as the slave clock and thus sync themselves to the AP through WiFi. The Master Clock is synced with the help of a PTP clock, connected through LAN. There are two possible approaches, hardware timestamping and software timestamping.

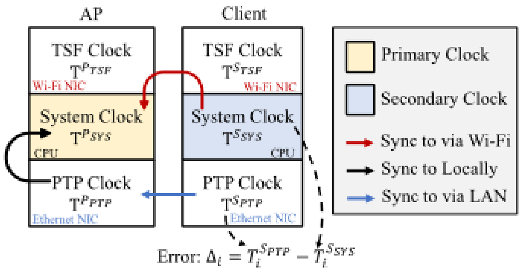
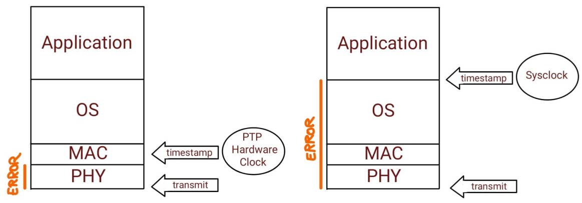


Figure 2: Wireless Clock Synchronization [5]



(a) Hardware Timestamping (b) Software Timestamping

Figure 3: Types of Timestamping

Hardware timestamping involves using dedicated hardware components within network devices, such as network interface cards (NICs), to generate timestamps directly at the physical layer of the network stack. But wireless networks use WNIC (Wireless Network Interface Controller) instead of NIC and thus do not support hardware counters that are needed for hardware timestamping. A solution for this issue presented by [5] is to treat TSF as the hardware clock. Thus, we can emulate the hardware PTP process used for Ethernet NICs on WNICs. The application of hardware timestamping produces sub-microsecond bias error and jitter.

Software timestamping on the other hand, captures precise time information at the software level instead of relying on hardware timestamps. This approach facilitates clock synchronization across networked devices, especially in wireless environments where hardware timestamping may not be feasible due to cost or practical limitations. Software timestamping records the exact time when a PTP event message is processed by the protocol stack. This, however leads to a worse performance and a bigger error margin when it comes to synchronization in comparison to hardware timestamping *(Figure 3)*. Since clock synchronization relies on getting the time from the system clock [8], its synchronization is neither accurate nor stable [5]. To minimize inaccuracy, we can observe the effect Interrupt Mitigation and CPU Power Management has over latency in software timestamping. Interrupt Mitigation aggregates multiple NIC interrupts into one singular interrupt to reduce computational cost and performance impact. When interrupt mitigation is enabled, interrupts of packets that were received at the same time period get delayed and gets a later response from the system, which results in latency [5]. This feature can be disabled to minimize timestamping latency. Moreover, dynamic ticks can also be disabled to boost clock stability [8]. When there are no operations requiring much computational power, modern CPUs turns off several hardware components to be more lightweight and to conserve power [5]. When disabled, software timestamping performs much better with a smaller offset [8].

4.2. Extending TSN towards Wireless Networks

Time Sensitive Networking (TSN) are a set of standards for providing a deterministic and reliable connection in Ethernet networks [15]. But due to the insufficient flexibility of a wired network, newer networks shifted towards wireless connections, sacrificing determinism TSN brings along the way. Wifi 7 (IEEE 802.11be), released in Januay 2024, aims to also integrate TSN extensions for low-latency real time traffic [3]. The central challenge involves adapting TSN mechanisms, initially tailored for Ethernet, to the inherently less predictable wireless environment. This task entails tackling issues like link unreliability, asymmetric path delays, and interference, all while maintaining compatibility with existing WiFi standards. Notably, the wireless network should have less overhead to achieve an accurate clock synchronization [16].

IEEE 802.11be introduces significant enhancements to both the physical (PHY) and medium access control (MAC) layers, specifically tailored to support TSN. On the PHY side, the amendment incorporates the 6 GHz band, allowing for wider channels up to 320 MHz and supporting higher modulation schemes like 4096-QAM [3]. These improvements collectively enhance data rates and reduce latency. Additionally, the expansion to 16 spatial streams optimizes spectrum utilization, benefiting time-sensitive applications by minimizing waiting times in buffers.

At the MAC layer, key advancements include extending multi-user (MU) capabilities such as MU-MIMO and orthogonal frequency-division multiple access (OFDMA) [3]. This extension is included in IEEE 802.11be. These technologies enhance spectral efficiency and decrease channel access latency by enabling simultaneous transmissions between multiple users. According to [17], the introduction of multi-link operation (MLO) is substantial. MLO allows multiple links for a single transmission, improving throughput, reliability, and latency. Opportunistic link selection, link aggregation, and multi-channel full duplex operations further enhance time-sensitive network handling. Their research believes that MLO will allow for a better performance for real-time applications even with the presence of heavy network traffic.

Furthermore, IEEE 802.11be emphasizes multi-AP coordination, bolstering its TSN capabilities [3]. By enabling access points (APs) to coordinate transmissions and share opportunities, the amendment reduces inter-network interference and optimizes overall network performance. This coordination is particularly valuable in operation settings with closely located APs.

The integration of TSN into WiFi 7 via IEEE 802.11be holds promise for various IoT applications, including multimedia, healthcare, industrial automation, and transportation. These applications demand low-latency and high-reliability communication, which WiFi 7 addresses through advanced PHY and MAC enhancements, whilst caving the way for a wireless implementation of the TSN regulations. Future challenges for next WiFi implementations include optimizing the existing PHY layer to reduce computational costs and achieve ultra-low latency, whilst also maintaining efficiency and network management [17]. While challenges persist in adapting TSN to wireless contexts, ongoing research shows a bright future for time-sensitive wireless communications.

5. Conclusion and Future Work

Evident with the advances made in the latest Wifi releases and continuous research refining the accuracy of software and hardware timestamping, merging Precision Time Protocol with WiFi not only seems feasible but also promises significant rewards in terms of precision and clock synchronization. Developing WiFi in the direction of TSN regulations and finding a middle ground between flexibility of wireless connections and the stability TSN brings could revolutionize real-time applications, offering unparalleled reliability and performance. When it comes to optimizing timestamping, disabling Interrupt Mitigation and dynamic ticking would end up reducing power efficiency, which is also an important aspect to consider in mobile devices. Conducting research on how to configure the operating system in a way to allow logical duty cycling between active and idle modes inbetween transmitting can optimize power management and enhance overall network efficiency. Combined with the evolution of TSN regulations, WiFi can offer Ethernet-like latency, setting a new standard for real-time communication in diverse applications.

References

[1] S. T. Watt, S. Achanta, H. Abubakari, E. Sagen, Z. Korkmaz, and H. Ahmed, "Understanding and applying precision time protocol," pp. 1–7, 2015.

[2] N. Finn, "Introduction to Time-Sensitive Networking," vol. 2, no. 2, pp. 22–28, 2018.

[3] T. Adame, M. Carrascosa-Zamacois, and B. Bellalta, "Time-sensitive networking in IEEE 802.11 be: On the way to low-latency WiFi 7," *Sensors*, vol. 21, no. 15, p. 4954, 2021.

[4] 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," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3808–3819, 2022.

[5] P. Chen and Z. Yang, "Understanding PTP Performance in Today's Wi-Fi Networks," *IEEE/ACM Transactions on Networking*, vol. 31, no. 6, pp. 3037–3050, 2023.

[6] S.-Y. Hu, J.-F. Chen, and T.-H. Chen, "VON: a scalable peer-to-peer network for virtual environments," vol. 20, no. 4, pp. 22–31, 2006.

[7] F. Rezabek, M. Helm, T. Leonhardt, and G. Carle, "PTP Security Measures and their Impact on Synchronization Accuracy," pp. 109–117, 2022.

[8] K. Ichikawa, "Precision time protocol on linux introduction to linuxptp," in *Proc. Linux Conf.*, 2014.

[9] D. Mills, "Internet time synchronization: the network time protocol," vol. 39, no. 10, pp. 1482–1493, 1991.

[10] D. L. Mills, "RFC0958: Network Time Protocol (NTP)," 1985.

[11] *An Introduction to Wi-Fi*, Digi International Inc., 2007-2008.

[12] A. A. O. Wafa S. M. Elbasher, Amin B. A. Mustafa, "A Comparison between Li-Fi, Wi-Fi, and Ethernet Standards," vol. 4, no. 12, 2015.

[13] C. Pei, Y. Zhao, G. Chen, R. Tang, Y. Meng, M. Ma, K. Ling, and D. Pei, "WiFi can be the weakest link of round trip network latency in the wild," pp. 1–9, 2016.

[14] S. Fan, Y. Ge, and X. Yu, "Comparison Analysis and Prediction of Modern Wi-Fi Standards," pp. 581–585, 2022.

[15] M. K. Atiq, R. Muzaffar, Ó. Seijo, I. Val, and H.-P. Bernhard, "When IEEE 802.11 and 5G Meet Time-Sensitive Networking," *IEEE Open Journal of the Industrial Electronics Society*, vol. 3, pp. 14–36, 2022.

[16] J. Haxhibeqiri, X. Jiao, M. Aslam, I. Moerman, and J. Hoebeke, "Enabling TSN over IEEE 802.11: Low-overhead Time Synchronization for Wi-Fi Clients," vol. 1, pp. 1068–1073, 2021.

[17] D. Cavalcanti, C. Cordeiro, M. Smith, and A. Regev, "WiFi TSN: Enabling Deterministic Wireless Connectivity over 802.11," *IEEE Communications Standards Magazine*, vol. 6, no. 4, pp. 22–29, 2022.