IEEE 802.1 TSN Time Synchronization over Wi-Fi and 5G Mobile Networks

Minh-Thuyen Thi*, Sosthène Guédon†, Siwar Ben Hadj Said*, Michael Boc*, David Miras‡
Jean-Baptiste Dore‡, Marc Laugeois‡, Xavier Popon‡, Benoit Miscopein‡
* Institute LIST, CEA, Paris-Saclay University, France
† École Polytechnique, Polytechnic Institute of Paris, France
‡ CEA-Leti, Grenoble Alpes University, France
Email: minh-thuyen.thi@cea.fr

***Abstract*—IEEE 802.1 Time-Sensitive Networking (TSN) is a set of standards that target deterministic communications with low latency. Among those standards, IEEE 802.1AS Time Synchronization is highly important as many other standards rely on it. Similar to wired TSN, wireless TSN currently also sees growing demand, especially from the real-time applications in smart factory. The document *IEEE Std 802.1AS-2020* has specified how we can use this standard in non-Ethernet networks such as Wi-Fi. In this paper, we present our experiments and performance analyses of IEEE 802.1AS over Wi-Fi and 5G mobile systems. We provide qualitative, quantitative and experimental analyses on the setup, benchmark, and system performance. These analyses provide insights into (i) whether IEEE 802.1AS can be used straightforwardly in wireless networks, (ii) what and how factors impact its performance, and (iii) what hardware/software should be developed to improve this performance.**

I. Introduction

IEEE 802.1 Time-Sensitive Networking (TSN) consists of many standards that are extended from the IEEE Ethernet and IEEE 802.1Q [1]. TSN aims to support low delay and deterministic communications such as the real-time communications in industrial smart manufacturing or Advanced Driver-assistance Systems (ADAS). These communications have to guarantee the performance of critical traffic when it coexists with best-effort traffics. For example, in automated production lines of a smart factory, the controllers/servers, robots, actuators and sensors need to have rapid and deterministic exchanges between each other, in order to guarantee synchronized and timely operations for the robots. The objective of TSN is to support deterministic communications in these industrial devices. Similar requirements exist in ADAS, where the controller needs to collect data and react rapidly according to road traffic events. To address these requirements, some proprietary or non-Ethernet standards have been developed, however currently TSN is expected to be widely adopted, providing an inter-compatible Ethernet-based solution.

Among four main groups of TSN standards (i.e., time/clock synchronization, low latency, high reliability and resource management), time synchronization is an essential group. This group and its main standard IEEE 802.1AS provide network-wide highly precise time synchronization, which is used by several other standards.

Since TSN initially targets Ethernet, most of TSN developments are on wired network. However, wireless communications have become increasingly important in current and future industrial communication systems. For instance, in the above example of smart factory, the controllers/servers might locate in a remote site and connect to other devices via a wireless link, e.g., 5G mobile link. Then the communications among the devices are over both wired and wireless connections.

Achieving accurate time synchronization is the first step towards making TSN available on wireless networks. In this paper, we focus on the implementation and experiment of IEEE 802.1AS over IEEE 802.11 and 5G mobile networks. Differ with wired connections, there exist several challenges when running time synchronization over wireless connections. The two main challenges are the high delay variation and the imprecise timestamping in wireless networks. These two factors can cause major negative impacts to the performance of IEEE 802.1AS due to the following reason. IEEE 802.1AS is based on Precision Time Protocol (PTP) that estimates the transmission delay (called *peer delay*) and calculates the difference between devices' clocks (called *offset*) by exchanging messages containing transmission and reception timestamps. Therefore if the peer delay estimation and the timestamps are inaccurate, the offset will be inaccurately calculated.

We provide analytical evaluations (i.e., qualitative, quantitative) and experimental analyses on the setup, benchmark, and performance of IEEE 802.1AS over Wi-Fi and 5G. To highlight the issues when running IEEE 802.1AS over wireless networks, we especially analyze the setup challenges and the factors that negatively impact the performance. We show that 5G generally achieves higher performance than IEEE 802.11 but the two systems have some similar challenges, which come from the characteristics of wireless links. For simplicity, hereafter we use *Wi-Fi* interchangeably with IEEE 802.11 since Wi-Fi is our default implementation of IEEE 802.11.

The contributions of this paper are twofold. First, we conduct several experiments of IEEE 802.1AS on Wi-Fi and 5G mobile networks, then show the benchmarks and compare the performances on various aspects of time synchronization. Second, we provide qualitative and quantitative analyses on these experiments to help answer the following questions: whether wireless networks can adopt IEEE 802.1AS straightforwardly, how performance can be degraded by the wireless links, and what hardware/software development can help to improve this performance.

II. Related Works

There exist several research works about running PTP over Wireless Local Area Network (WLAN) and Wi-Fi. The studies of Kannisto et al. [2] [3] and Cooklev et al. [4] are among the earliest works. The software-timestamping approach in [2] is developed further in [3]. The authors also develop a hardware prototype in [3] to show that nanosecond accuracy is possible with hardware-based timestamping.

Aneeq et al. [5] [6] [7] [8] [9] proposed some other implementations of PTP over WLAN, using software timestamping. In [6] and [7], a protocol similar to IEEE 1588 is utilized, however the synchronization information is transmitted inside the WLAN beacons, instead of inside dedicated synchronization messages. Reinhard et al. [10] [11], who belong to the same group with Aneeq, studied Physical Layer Timestamps in WLAN using an FPGA-based hardware platform called SMiLE. They buid an inherited version of IEEE 1588 to evaluate their Physical Layer Timestamps approach.

Recently, several new approaches have been proposed, e.g. FTM for Wi-Fi [12], TSN translators for 5G [13]. The authors in [14] and [15] implement FTM for indoor positioning. The integration of FTM into IEEE 802.1AS requires supports from hardware [12], such as adding PTP information into *Vendor-Specific* field of FTM messages, or accessing FTM timestamps. To the best of our knowledge, there is not yet any published work on the implementation of FTM-PTP integration.

In cellular systems, there are several implementations and proposals for high accurate time synchronization in 5G. For example, one of the proposals for time synchronization in 5G is an enhanced SIB16 [16]. However, the most noticeable effort is to integrate TSN into 5G. For example, Farkas et al. [17] proposed TSN translators that allow the synchronization of Ethernet TSN to cross 5G wireless domain. The translators record the ingress and egress timestamps of the PTP messages when these messages cross the 5G domain. After having two timestamps, the translators can calculate the residence time of PTP messages crossing the 5G domain, then deduct this residence time from the overall delay. Several other works have been also developed based on this TSN translator model [18] [19]. Recently the TSN translators were prepared for RFC in IETF [13] and adopted in 3GPP Release 16 [20]. In this paper, we show that the approaches such as TSN translators are necessary for integrating TSN into 5G. Otherwise, the performance of PTP will be degraded significantly when we run it over wireless links straightforwardly.

III. PTP and IEEE 802.1AS

Precision Time Protocol (PTP) was defined in IEEE 1588 [21], and was extended in IEEE 802.1AS standard [12] to target highly precise network-wide synchronization in TSN. For example, the standard specifies that end-to-end precision is below 1 microsecond over 7-hops network. In the network, one node is assigned to be Grand Master (GM) to play the role of master. Other nodes play the role of slave by synchronizing their clocks to the clock of GM.

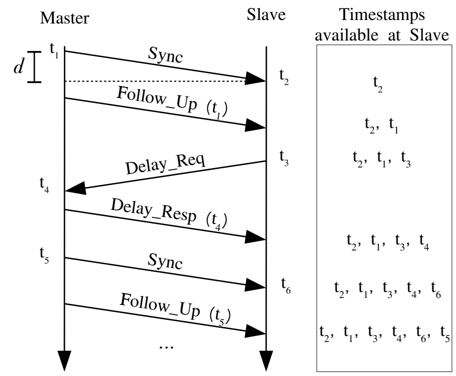


Fig. 1: PTP protocol

Fig. 1 depicts the PTP protocol, in which a slave gathers timestamps to estimate the clock offset between its clocks and the GM's clock. The GM disperses `Sync` messages in order to advertise its clock. Note that in this two-step PTP mechanism, the GM sends `Follow_Up` message to convey the transmission timestamp of the `Sync` message. Similarly, the slaves send `Delay_Req`, then receive `Delay_Resp` to collect other timestamps.

After collected timestamps, the slave estimates peer delay d by averaging the two-way transmission's delay, assuming that the link is symmetric. That estimation is given as

$$d = \frac{(t_4 - t_3) + (t_2 - t_1)}{2}, \tag{1}$$

where t_1, t_2, t_3 and t_4 are transmission timestamps and reception timestamps of `Sync` and `Delay_Req` message, respectively. The time offset δ is calculated as

$$\delta = t_6 - t_5 - d, \tag{2}$$

with t_5 and t_6 are the transmission and reception timestamp of the next `Sync` message, respectively.

Eq. 1 is possibly inaccurate due to link asymmetry and clock drift, i.e., clocks advancing at different rates. In real systems, peer delay variation might highly impact the precision of the synchronization.

IV. PTP over Wireless Link

Wi-Fi has its time synchronization mechanism called the Time Synchronization Function (TSF). This mechanism was reported to have an error around $10\mu s$ [22]. However, TSF is not highly scalable in multi-hop networks [23] [24]. To allow running PTP over non-Ethernet networks, the revision IEEE 802.1AS-2020 introduces Media-dependent and Media-independent interfaces. These interfaces act as abstraction layers to facilitate the integration of PTP into non-Ethernet networks. These layers allow the networks to have their own means of measuring round trip time (RTT), for example Fine Time Measurement (FTM) in IEEE 802.11.

In cellular systems, several approaches have been developed or proposed for time synchronization, such as PTP- or non-PTP-based over-the-air (OTA) synchronization, Global Navigation Satellite System (GNSS)-based synchronization, dedicated Radio Resource Control (RRC) signaling [16]. Among these approaches, OTA sychronization is mainly used in Long-Term Evolution (LTE) small cells base stations [16] [25], while GNSS-based synchronization requires costly equipments and generally targets outdoor environment [26]. As 5G requires high precision and low cost network-wide synchronization, especially for indoor industrial systems, TSN and IEEE 802.1AS were proposed to be integrated into 5G. However, in the following subsections, we show that IEEE 802.1AS encounters several challenges when it is used in wireless networks.

1) Challenges in Delay and Jitter: The precision of PTP relies strongly on delay estimation (Eq. 2), but delay in wireless networks is less deterministic than in wired networks. On one hand, various factors of wireless links, e.g., noise, interference, shadowing and multipath, can cause retransmission, making delay increase and become less deterministic. On the other hand, unlike Ethernet, wireless links are not full duplex and the medium is shared among multiple devices. Due to this non-full duplex, the uplink and downlink are asymmetric, then the peer delay d is not equal to the average as in Eq. 1. Moreover, due to the medium sharing, wireless networks need mechanisms for multiple access, e.g., Carrier Sense Multiple Access (CSMA). This mechanism potentially introduces more delay, making the overall delay becomes higher and less deterministic.

2) Challenges in Timestamping: PTP also relies heavily on the accuracy of timestamps, in order to estimate the peer delay d precisely (Eq. 1). As a result, several Ethernet cards implement hardware-timestamping which has an accuracy lower than a nanosecond. With this capacity, the timestamping accuracy is limited mainly by the quantization of the timestamps only, rather than the hardware itself [27].

To the best of our knowledge, hardware-timestamping is not currently available on commercial off-the-shelf (COTS) wireless cards. Therefore on these cards, we can only use system-clock to take software-timestamps. This process possibly produces delay and jitter, which come from the operating system's scheduling and interrupt handling. The variation of these timestamps might reach $600\mu s$ [5].

V. Experiment Setup

A. Hardware

Fig. 2 depicts the whole network of the Wi-Fi experiment. However, some experiment scenarios might use only a part of the network. Specifically, in *one-hop experiments*, only the Access Point and the Wi-Fi Station are used, and they connect to each other by both Ethernet and Wi-Fi. In *two-hop experiments*, all the stations and some links are used. The Wi-Fi Stations are computers with Linux kernel 4.19, connected with the Wi-Fi card *Intel* Dual Band Wireless-AC 8260.

For the 5G setup, we have one 5G User Equipment (UE), one 5G base station (BS), two Wired stations, two TSN bridges (Fig. 3), and one channel emulator. These 5G hardwares were described in our previous work [28] as a 5G-like testbed. The UE and BS, each of them is built from a radio frequency (RF) transceiver and a custom digital board. The RF transceiver is based on the AD9361 Agile Transceiver from *Analog Device*. This transceiver's band ranges from 70 MHz to 6 GHz, and supports bandwidths from less than 200kHz to 56 MHz. The custom digital board is built on the Zynq-045 *Xilinx* FPGA with a dual Cortex-A9 *ARM* processor. The two Wired Stations are Linux computers, while the TSN Bridges are the *NXP* switches with LS1028A processor. The Channel Emulator is used for emulating the radio channel such as setting the distance between the 5G BS and the UE.

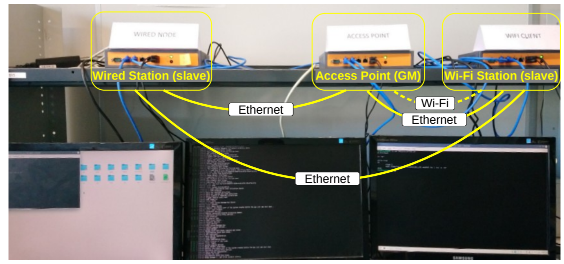


Fig. 2: Wi-Fi setup

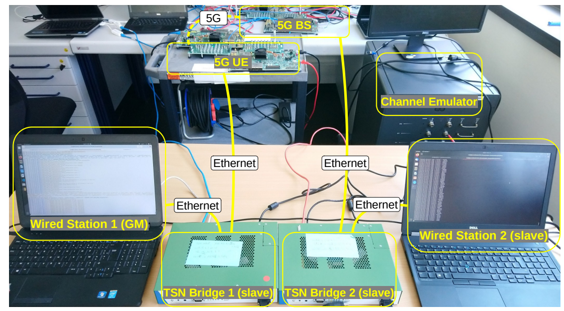


Fig. 3: 5G setup

B. Implementation

To have IEEE 802.1AS on Linux, we use *Linuxptp*, which has open source PTP implementation. We also use Linuxptp for all TSN bridges, wired and wireless stations in the experiments of both Wi-Fi and 5G. To run IEEE 802.1AS on the *Intel* Wi-Fi card, some modifications inside Linuxptp and the card's driver are needed. For example, as only reception timestamping is available, we need to implement transmission timestamping.

By default, we set the values of important Linuxptp parameters as in Table I. Note that the values are in logarithm, which means, for example, if a value is -3 then the time interval is 0.125 second.

Parameter	Description	Value
logSyncInterval	Time interval between two `Sync` messages	-3
logPdelayReqInterval	Time interval between two `Delay_Req` messages	-1
syncReceiptTimeout	Number of missed `Sync`/`Follow_Up` messages that indicates a missed transmission	1

TABLE I: Values of important parameters in Linuxptp

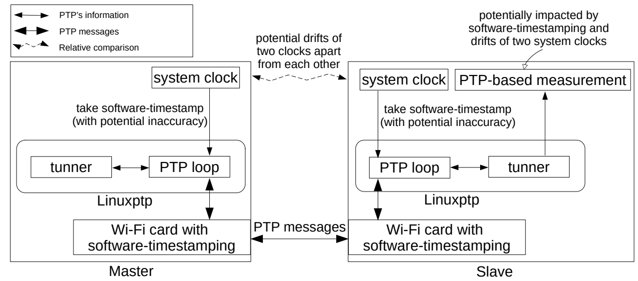


Fig. 4: Potential factors causing inaccuracy to PTP-based

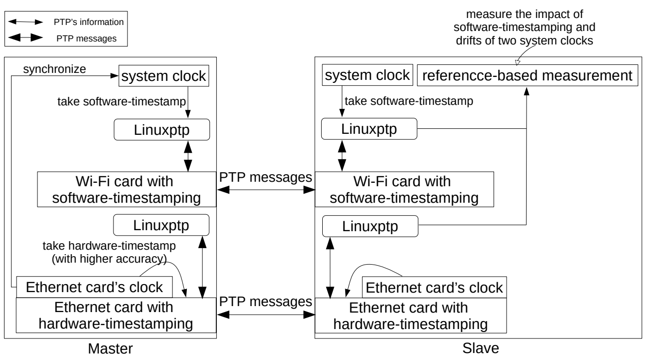


Fig. 5: Setup of reference-based measurements

C. PTP-based and Reference-based Setup

With the Wi-Fi experiment, we have two setups corresponding to two types of measurement, namely *PTP-based* and *reference-based* measurement. In general, the goal of PTP-based setup is to compare the synchronization over Wi-Fi and over Ethernet. The goal of reference-based setup is to benchmark more thoroughly the synchronization over Wi-Fi itself. PTP-based setup measures the time offset between clocks of GM and slave. This offset is computed based on the PTP protocol, and can be found at the output of Linuxptp.

In reality, there are factors that potentially make this PTP-based measurement less reliable. As showed in Fig. 4, those factors are the inaccuracy of software-timestamping and the clock drift between two system-clocks. In the figure, the master and slave exchange PTP messages that contain timestamps inside. Linuxptp takes these timestamps from the software-system-clock, instead of the hardware-clock inside the network card. Based on these timestamps, the slave measures the *offset* using the program *tunner*, and prints the results to the users. This measurement is subject to a potential clock drift between the two system-clocks, as well as a potential inaccuracy of software-timestamping. To make the measurement more reliable, we propose reference-based setup.

In the reference-based setup, we use a more accurate synchronization as reference for Wi-Fi. Given that hardware-timestamping has precision to a few nanoseconds, the synchronization over Ethernet with hardware-timestamping is used as reference (Fig. 5). Similar to Fig. 4, the master and slave in Fig. 5 exchange PTP messages and the slave measures the *offset*. However, in this setup, there are two separate PTP exchanges running in parallel, one via Wi-Fi link and one via Ethernet link. The PTP exchange via Wi-Fi is the same as in Fig. 4, while the PTP exchange via Ethernet is done with hardware-timestamping. At the master side, we synchronize the system-clock to the Ethernet card's clock. In this way, at the slave side, we can measure the difference between the system-clock and the Ethernet card's clock. This difference is named *reference-based offset*.

VI. Experiment Results

A. Wi-Fi

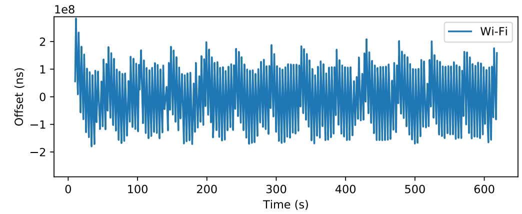
1) PTP-based Experiment: This is a *one-hop* experiment, so we use a network as in Fig. 2 but with the Wired Station disabled. The Access Point connects to the Wi-Fi Station via both Ethernet and Wi-Fi. We run time synchronization over Wi-Fi link first, then over Ethernet link. For each scenario (i.e., Wi-Fi, Ethernet with hardware-timestamping, and Ethernet with software-timestamping) we run an experiment for 3 periods of 10 minutes. The performances of 3 scenarios are showed in Fig. 6. The synchronization takes around 10 to 20 seconds to startup, so this period shows no result in the figure.

In general, the synchronization over Ethernet hardware-timestamping is around 1000 times more accurate than Ethernet software-timestamping, and the Ethernet software-timestamping is 1000 times more accurate than Wi-Fi (Fig. 6). Note that TSN for future industrial communications typically requires an accuracy in the order of microsecond. In terms of offset standard deviation, we have around $10ns$ with Ethernet hardware-timestamping, $10\mu s$ with Ethernet software-timestamping, and $10ms$ with Wi-Fi.

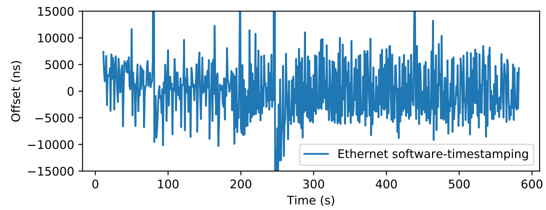
Compared to hardware-timestamping, software-timestamping achieves lower accuracy; this can be caused by less precise timestamps and higher jitter. In case of Wi-Fi, there are even more factors that cause the inaccuracy, e.g., the asymmetry of wireless links, the latency introduced by media access or retransmission.

2) Reference-based Experiment:

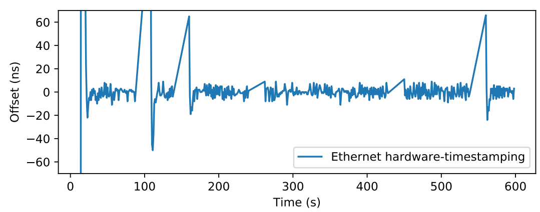
a) Synchronization over only Wireless Link: This is also a *one-hop experiment*, so we keep using the previous network topology, i.e., the Access Point and Wi-Fi Station connect with each other by both Ethernet and Wi-Fi. However, the synchronization over Ethernet is only for reference. Fig. 7 shows that the reference-based offset has generally lower variation than the PTP-based offset. Moreover, the PTP-based offset is centered around zero, whereas the reference-based offset is centered around $118ms$. This gap can be explained by the asymmetric role of the Wi-Fi devices, i.e., Wi-Fi client or access point. This asymmetry can be seen in Fig. 8, where we run the same experiment but with the Wi-Fi Access Point acts as a PTP slave. Unlike Fig. 7, the reference-based offset in Fig. 8 is centered around $-80ms$. This gap can be explained as follows. On one hand, the average amount of time between the reception of a packet and the moment when it is timestamped can be different on two devices. For example, there can be differences in scheduling and interrupt latency due to CPU power states [29]. On the other hand, since the roles of the devices lead to asymmetry, the synchronization offset is reversed when the roles are reversed.



(a) Wi-Fi



(b) Ethernet software-timestamping



(c) Ethernet hardware-timestamping

Fig. 6: One-hop: performance on offset

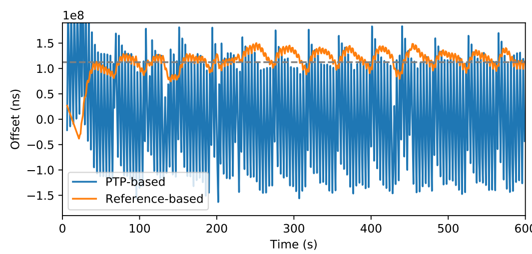


Fig. 7: One-hop: PTP- and reference-based offset

b) Synchronization across Wired and Wireless Links: This is a *two-hop experiment*, so the whole network in Fig. 2 is used. We evaluate the two-hop synchronization from the Access Point to the Wired Station (i.e., via the Wi-Fi Station). This synchronization passes across both wireless and wired links. The Ethernet link from the Access Point to the Wi-Fi Station is not used. The link between the Wired Station and the Wi-Fi Station uses software-timestamping, while the link between the Wired Station and the Access Point has hardware-timestamping. In this way, we use Ethernet

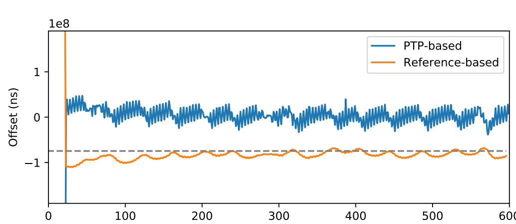


Fig. 8: One-hop: PTP- and reference-based offset (AP is slave)

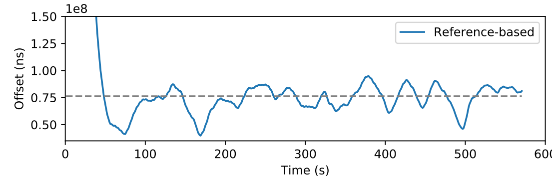
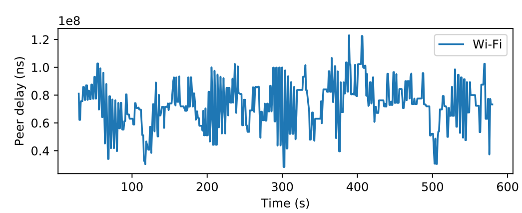


Fig. 9: Two-hop: reference-based offset between Wired Station and GM

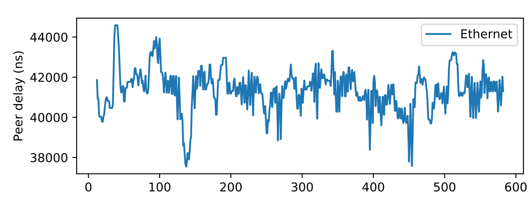
hardware-timestamping as reference in order to benchmark the two-hop synchronization between the Access Point and the Wired Station. The PTP messages are exchanged across both wired and wireless link. The objective is to see how the wireless link affects the overall two-hop synchronization.

Fig. 9 shows the reference-based offset of the two-hop synchronization between the Wired Station and the GM. We found that when the synchronization of the intermediate Wi-Fi Station has low accuracy, the performance of the Wired Station is also degraded. The offset between the Wired Station and GM has standard deviation of $95ms$ (Fig. 9), which is significantly higher than Ethernet software-timestamping whose standard deviation is generally within a few microseconds. In accordance with the results when there is only Wi-Fi, the Wired Station has a constant error of around $72ms$, which is caused by the synchronization over Wi-Fi.

Fig. 10 illustrates the peer delay of Wi-Fi and Ethernet in two-hop setup. Note that we use peer-to-peer instead of end-to-end PTP delay mechanism, so that Fig. 10 can show separately the delays of Wi-Fi and Ethernet links. We can see that the Wi-Fi link will be the main cause of the degradation in the overall performance because the delay and jitter of this Wi-Fi link are much higher than those of the Ethernet link. The jitter of Wi-Fi link is $1ms$, comapared to $1\mu s$ of Ethernet link. Jitter is an important factor in wireless TSN due to the following. On one hand, it is impacted significantly by the unreliability of wireless link. On the other hand, this jitter impacts negatively the accuracy of PTP, as PTP assumes that the link is symmetric and the peer delay is equal to the average of uplink and downlink delay (Eq. 1).



(a) Peer delay between Wi-Fi Station and Access Point

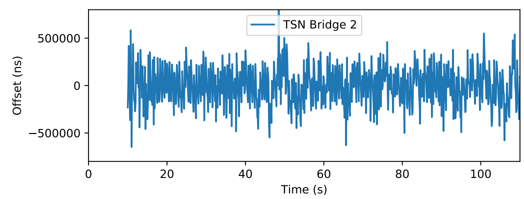


(b) Peer delay between Wi-Fi Station and Wired Station

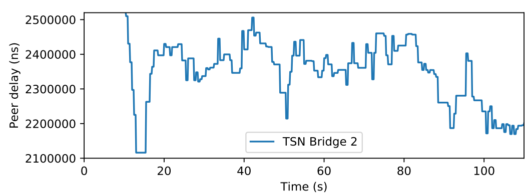
Fig. 10: Two-hop: performance on peer delay

B. 5G

1) Synchronization over only Wireless Link: We setup the network as in Fig. 3, then evaluate the synchronization from the GM to the TSN Bridge 2. This synchronization crosses the 5G mobile link. Fig. 11 shows that the synchronization over 5G has higher performance than over Wi-Fi, in both offset and peer delay. The main reason is that the delay and its variation in 5G (Fig. 11b) are lower than in Wi-Fi (Fig. 10a). An important cause of this difference is the multiple access mechanism of Wi-Fi and 5G. While Wi-Fi uses CSMA for multiple access, 5G multiplexes users in time and frequency without collision in connected mode.



(a) Offset

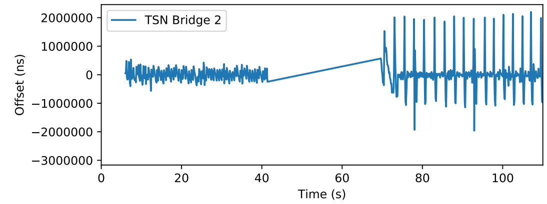


(b) Peer delay

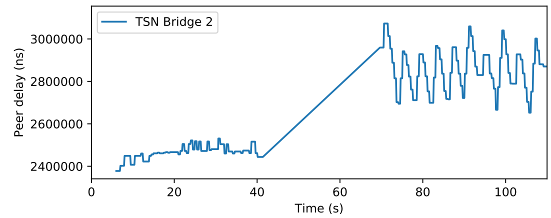
Fig. 11: One-hop: performance of TSN Bridge 2

To evaluate the impact of link capacity on the synchronization, we try varying the number of allocated time slots and changing the link distance. In a real system, the number of allocated time slots can be reduced if there are more user equipments communicate with a same BS. In Fig. 12, after 40 seconds, the 5G BS and the UE communicate with each other using only one time slot instead of four. The synchronization is disrupted and then recovered but keeps having high offset and high variation. The offset in Fig. 12a corresponds to the peer delay in Fig. 12b. In the first 40 seconds, the jitter is $33073ns$, after that it increases to $101087ns$. For both offset and peer delay, when reducing the number of time slots, the link is disrupted, then recovered but has lower performance.

To vary the distance between the 5G BS and the UE, we use the Channel Emulator. In Fig. 13, during the first 40 second, the distance is 1 kilometer, after that this distance changes to 1 meter. In the figure, we see that even the distance increases significantly, the offset is only slightly higher. This can be explained by two reasons. First, the overall delay in the first 40 seconds is only slightly higher than the overall delay in the period after that, $2461661ns$ compared to $2348179ns$. This is because the overall delay includes several component delays, e.g., propagation delay, processing delay; and the propagation delay of electromagnetic wave is only changed slightly when the distance changes from 1 meters to 1000 meter. Note that when increasing distance, the link is kept line-of-sight,



(a) Offset



(b) Peer delay

Fig. 12: One-hop: performance with reduced time slots

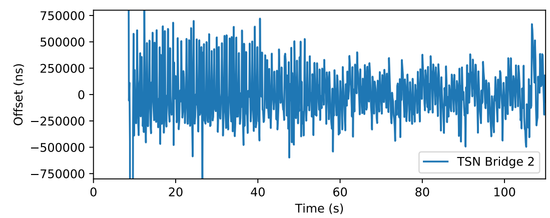
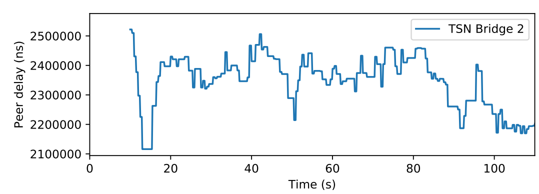
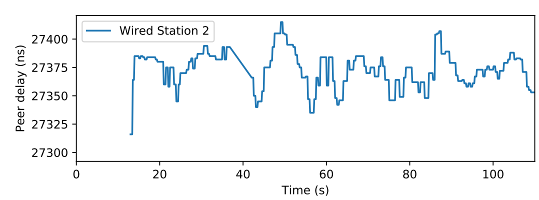


Fig. 13: One-hop: offset with increased distance

without introducing more shadowing or multi-path. Second, when changing distance, the performance degrades but after that it gradually recovers because even the peer delay is high, this delay becomes gradually stable. Since the PTP protocol is significantly impacted by delay variation, if the delay becomes stable, the performance recovers.



(a) Peer delay of TSN Bridge 2



(b) Peer delay of Wired Station 2

Fig. 14: Two-hop: performance on peer delay

2) Synchronization across Wired and Wireless Links: Similar to the Wi-Fi two-hop setup, to observe the synchronization across both wired and wireless links, we collect the results of both TSN Bridge 2 and Wired Station 2 in a same experiment. Peer delays of the two devices are showed in Fig. 14. Similar to the Wi-Fi experiment, the peer delay of Wired Station 2 is significantly lower than TSN Bridge 2 because the link between Wired Station 2 and TSN Bridge 2 has Ethernet hardware-timestamping.

VII. Conclusion

This paper presents the implementation, experiments and analyses of IEEE 802.1AS over Wi-Fi and 5G mobile networks. We propose a reference-based method to benchmark the time synchronization over wireless links. We show that there exist several challenges when running IEEE 802.1AS over wireless networks, and there are still developments needed in order to meet the requirements of future TSN applications. The main problems are the lack of hardware-timestamping and the characteristics of wireless links, e.g., the high delay and jitter introduced by retransmission, media access. To deal with these problems, one promising solution in Wi-fi is to integrate FTM into 802.1AS. However, we have tested and verified that currently full integration cannot be realized on COTS devices due to the lack of hardware support. For example, the hardware must allow writing *VendorSpecific* field and reading the timestamps inside FTM messages. In 5G, TSN translator is also a promising solution. One of our future works is to develop solutions for these problems based on FTM and TSN translator.

Acknowledgement

CPS4EU project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement N 826276.

References

[1] "IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks–Corrigendum 2: Technical and editorial corrections," *IEEE Std 802.1Q-2011/Cor 2-2012 (Corrigendum to IEEE Std 802.1Q-2011)*, 2012.

[2] J. Kannisto, T. Vanhatupa, M. Hännikäinen, and T. D. Hämäläinen, "Precision Time Protocol Prototype on Wireless LAN," in *Telecommunications and Networking - ICT 2004*, J. N. de Souza, P. Dini, and P. Lorenz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1236–1245.

[3] 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.

[4] T. Cooklev, J. C. Eidson, and A. Pakdaman, "An implementation of IEEE 1588 over IEEE 802.11 b for synchronization of wireless local area network nodes," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 1632–1639, 2007.

[5] A. Mahmood and G. Gaderer, "Timestamping for IEEE 1588 based clock synchronization in wireless LAN," in *2009 International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE, 2009, pp. 1–6.

[6] A. Mahmood, G. Gaderer, and P. Loschmidt, "Software support for clock synchronization over IEEE 802.11 wireless LAN with open source drivers," in *2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE, 2010, pp. 61–66.

[7] ——, "Clock synchronization in wireless LANs without hardware support," in *2010 IEEE International Workshop on Factory Communication Systems Proceedings*. IEEE, 2010, pp. 75–78.

[8] 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.

[9] A. Mahmood and F. Ring, "Clock synchronization for IEEE 802.11 based wired-wireless hybrid networks using PTP," in *2012 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication Proceedings*. IEEE, 2012, pp. 1–6.

[10] R. Exel, J. Mad, G. Gaderer, and P. Loschmidt, "A novel, high-precision timestamping platform for wireless networks," in *2009 IEEE Conference on Emerging Technologies & Factory Automation*. IEEE, 2009, pp. 1–8.

[11] R. Exel, "Clock synchronization in IEEE 802.11 wireless LANs using physical layer timestamps," in *2012 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication Proceedings*. IEEE, 2012, pp. 1–6.

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

[13] A. S. J. Farkas Ed., T. Dudda and S. Sandberg, "5G - Ultra-Reliable Wireless Technology with Low Latency," Internet Requests for Comments, RFC Editor, RFC.

[14] M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai, "Verification: Accuracy Evaluation of WiFi Fine Time Measurements on an Open Platform," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 417–427.

[15] B. Horn, "Doubling the Accuracy of Indoor Positioning: Frequency Diversity," *Sensors*, vol. 20, p. 1489, 03 2020.

[16] A. Mahmood, M. I. Ashraf, M. Gidlund, and J. Torsner, "Over-the-air time synchronization for URLLC: Requirements, challenges and possible enablers," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2018, pp. 1–6.

[17] J. Farkas, B. Varga, G. Miklós, and J. Sachs, "5G-TSN integration meets networking requirements for industrial automation," *Ericsson: Stockholm, Sweden*, pp. 0014–0171, 2019.

[18] A. Larrañaga, M. C. Lucas-Estañ, I. Martinez, I. Val, and J. Gozalvez, "Analysis of 5G-TSN integration to support industry 4.0," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2020, pp. 1111–1114.

[19] T. Striffler and H. D. Schotten, "The 5G Transparent Clock: Synchronization Errors in Integrated 5G-TSN Industrial Networks," in *IEEE 19th International Conference on Industrial Informatics*. IEEE, 2021.

[20] "3GPP TS 24.535, Device-Side Time Sensitive Networking (TSN) Translator (DS-TT) to Network-Side TSN Translator (NW-TT) protocol aspects," 3GPP, Tech. Rep., 2020.

[21] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2019 (Revision ofIEEE Std 1588-2008)*, pp. 1–499, 2020.

[22] D. Zhou, L. Huang, and T.-H. Lai, "On the scalability of ieee 802.11 ad-hoc-mode timing synchronization function," *Wireless Networks*, vol. 14, pp. 479–499, 08 2008.

[23] J. So and N. Vaidya, "MTSF: A timing synchronization protocol to support synchronous operations in multihop wireless networks," *University of Illinois at Urbana-Champaign, Tech. Rep*, 2004.

[24] L. Huang and T.-H. Lai, "On the scalability of IEEE 802.11 ad hoc networks," *EPFL MOBIHOC'02, Lausanne*, 05 2002.

[25] D. Bladsjö, M. Hogan, and S. Ruffini, "Synchronization aspects in LTE small cells," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 70–77, 2013.

[26] A. Mahmood, M. I. Ashraf, M. Gidlund, J. Torsner, and J. Sachs, "Time synchronization in 5G wireless edge: Requirements and solutions for critical-MTC," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 45–51, 2019.

[27] 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)*, 2014, pp. 1–8.

[28] R. Gerzaguet, S. Bicais, P. Rosson, J. Estavoyer, X. Popon, D. Dassonville, J.-B. Dore, B. Miscopein, M. Pezzin, D. Miras, and D. Ktenas, "5G Multi-Service Field Trials with BF-OFDM," in *2017 IEEE Globecom Workshops (GC Wkshps)*, 2017, pp. 1–5.

[29] P. Chen and Z. Yang, "Understanding Precision Time Protocol in Today's Wi-Fi Networks: A Measurement Study," in *2021 USENIX Annual Technical Conference*. USENIX, 2021, pp. 597–610.