On the use of White Rabbit for Precise Time Transfer in 5G URLLC Networks for Factory Automation Applications

Óscar Seijo, Iñaki Val
Communication Systems Group,
IK4-Ikerlan Technology Research Centre,
Pº J.M. Arizmendiarrieta 2, 20500,
Mondragón Spain
oseijo@ikerlan.es, ival@ikerlan.es

Jesús A. López-Fernández
Department of Electrical Engineering
Group of Signal Theory and Communications,
University of Oviedo, Gijón 33203, Spain
jelofer@uniovi.es

Jon Montalban, Eneko Iradier
Department of Communications Engineering
University of the Basque Country, UPV/EHU
Bilbao, Spain
jon.montalban@ehu.eus,
eneko.iradier@ehu.eus

***Abstract*—The implementation of 5G networks has several technological challenges, which are mainly related to the Ultra-Reliable and Low Latency Communications (URLLC). One of these challenges is precise time transfer. Precision Time Protocol (PTP) is the most widely used protocol for precise time transfer. However, PTP performance may not be sufficient in large 5G networks.**

In this paper, we propose the use of white rabbit as the main technology for frequency and time transfer in 5G transport networks. However, white rabbit can only be implemented in wired links, thus the time transfer performance in wireless links may not be enough for the most challenging URLLC use cases. Therefore, we propose in this paper a modification of the white rabbit protocol that can be used in wireless communications. The proposed wireless white rabbit protocol has been tested by simulations over a single carrier point to point communication system, and different channel models. The results show significant improvements of the time transfer accuracy in Line Of Sight channels compared to other common wireless time transfer approaches and a similar performance to white rabbit over wired links.

Keywords—clock synchronization, PTP, white rabbit, industrial wireless communications, 5G, URLLC

I. INTRODUCTION

5G is a combination of networks and advanced technologies that combines efficiency in design, flexibility in operation and potential configurations. Both efficiency and flexibility enable the use of 5G communications to virtually any communication scenario and provide significant advantages with respect to other networks and communication technologies practically in any environment. The performance targets of 5G represent a revolutionary step ahead, with user data rates over 10 Gbps, end-to-end latencies below 1 ms, mobility use cases over 500 km/h and connected device densities that exceed 10^6 per square km. Latency control is one of the major constraints in 5G designs and future deployments and is still a matter of discussion depending upon the final use case. In the case of enhanced mobile broadband (eMB) and massive machine type communications (MMTC) use cases, the target values range from 10 to 100 ms. These values are not low enough for the Ultra-reliable and low latency communications (URLLC) case. This use case is focused on time sensitive applications originating for vertical industries such as industrial automation, smart grids, tactile internet, automotive, etc and the target latency boundaries for these cases are below 1 ms.

The ecosystem of infrastructure and services offered by 5G can be described using a three layered model [1]. The first layer is an Infrastructure Resource Layer that comprises the physical resources of a fixed-mobile converged network. This converged network includes access nodes, cloud infrastructure, 5G devices, networking nodes and associated links. All these physical assets are managed based upon a paradigm of virtualization of network functions controlled by an orchestration entity. The second layer is the Business Enablement Layer that can be considered as the network function library. This library will be used by the orchestration entity to provide the required network services for each specific configuration and use case. Finally, the Business Application Layer will serve the specific applications and services of the operator and verticals using the 5G network. A simplified model of the 5G architecture is shown in Fig. 1. The network is composed of three blocks, wireless terminals (UEs), Radio Access Network (RAN) and Core Network.

Transport networks, i.e. backhaul are a critical part of the infrastructure required for successful deployment of 5G networks. There is currently a wide range of backhaul solutions that together can address the backhaul challenges of 5G networks. In most cases, the 5G-transport network will be based on fiber but a universal deployment of fiber is unrealistic for several reasons [2]. Wireless solutions for front and backhaul are especially convenient in dense urban microcell and picocell areas. The deployment and operation costs provide competitive advantages in this environment if compared with massive fiber deployments. In addition, civil works for fiber deployments are not possible in protected urban and rural areas.

While economically advantageous from a deployment and operation perspective, wireless backhaul faces some technical challenges to fulfill 5G requirements. One of the potential

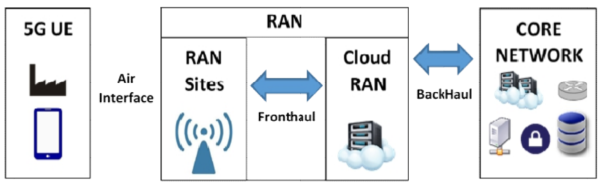

Fig. 1. 5G network architecture and function of front/backhaul.

limitations of a wireless backhaul is related to latency and synchronization. Advanced functions such as inter-site carrier aggregation, shortest frames in 5G as well as positioning functions impose a synchronization accuracy in the order of few hundreds of nanoseconds [3]. If compared to the typical microsecond requirement in 4G and previous generation networks, clock synchronization distribution will be key for successful wireless backhauling. Besides, advanced synchronization capabilities would also benefit URLLC applications deployed through 5G. For example, two possible applications are:

- Financial markets, which need to share a very precise global time to ensure the integrity of the transactions.
- Cyber Physical Systems (CPS), where the nodes of the CPS need to synchronously perform specific tasks, such as periodically reading the value of a sensor, or acting over a physical variable through an actuator.

Focusing on CPS, the time transfer performance required in a CPS strictly depends on the application and technologies involved. For example, a Factory automation facility is usually deployed in an area of hundreds of square meters and requires time transfer accuracies that exceed current practical performance of nowadays cellular networks. However, this synchronization level cannot be delivered by nowadays cellular networks. In consequence, the nodes of an industrial network, such as a power smart grid network, usually synchronize their local clocks to the Coordinated Universal Time (UTC) of a Global Navigation Satellite System (GNSS) constellation, i.e. Global Positioning System (GPS). The data generated by the nodes is then transferred either through the cellular network or dedicated links. Due to the aforementioned reasons, the nodes of a smart grid are expensive and rather complex. Finally, GNSS synchronization is very vulnerable to spoofing, hacking and bad climate conditions [4].

Definitely, precise time transfer in 5G would offer three main advantages: i) a reduction in the deployment cost and complexity of 5G-based industrial networks, ii) an improvement in the fault tolerance and security of the 5G-based industrial networks, and iii) a significant improvement of the 5G network performance.

The most commonly used protocol for time transfer is Precision Time Protocol (PTP). PTP can provide a synchronization level in the range of some nanoseconds [5]. However, PTP time transfer accuracy is deteriorated in multi-hop networks, which would be an issue in 5G networks, as they are composed by multiple interconnected systems.

White Rabbit protocol is an extension to PTP that has two major advantages over PTP: it can provide sub-nanosecond time synchronization accuracy, and it has little loss of accuracy in multi-hop networks. The key of White Rabbit performance is sharing the clock phase and frequency along every element of the network.

White Rabbit was initially designed by researchers from The European Organization for Nuclear Research (CERN) to cope with the strict timing requirements of the Large Hadron Collider (LHC). Now, there are some initiatives to introduce White Rabbit in other scenarios. For example, White Rabbit for Industrial Timing Enhancement (WRITE) project [6] aims at progressively substitute Precision Time Protocol (PTP) with White Rabbit. According to WRITE, the adoption of White Rabbit in could cause a major impact in the performance of industrial systems and scientific communities [6].

White Rabbit will be included in the IEEE 1588 standard as High Accuracy [7] and it can be handily implemented in wired links. However, wireless systems lack a robust and standardized method to perform high-accuracy time transfer. The lack of a reliable method to provide precise time transfer over wireless would be a major issue in the deployment of 5G wireless backhauls.

There are several state-of-the-art solutions based on different technologies to perform precise time transfer over wireless. For example, a relevant research is done in [8], where an IEEE 802.11b standard [9] implementation over FPGA with high precision timestamps is presented. This system offers time transfer accuracy in the order of hundreds of picoseconds under Line-Of-Sight (LOS) conditions. To do so, the authors use the IEEE 802.11b Direct Sequence Spread Spectrum (DSSS) modulation to precisely estimate the Time-of-Arrival (ToA) of the frames. However, IEEE 802.11b is a low-speed standard that has been superseded by faster IEEE 802.11 standards. On the other hand, we presented in [10] an IEEE 802.11g FPGA modem implementation that uses hardware timestamps and PTP to achieve a reasonable precision in the range of tens of nanoseconds over variant channels. The hardware timestamps were based on the detection of the received frames start index. Nevertheless, the clock accuracy of both proposals is still very far from the White Rabbit time transfer accuracy.

To the best of our knowledge, there was only one active project targeting White Rabbit translation into wireless links [11], which was supported by CERN. The result of the project was an implementation of White Rabbit over a wireless link [12]. However, the White Rabbit implementation was not specifically optimized for wireless. In consequence, the synchronization accuracy was in the range of some nanoseconds, still far away from the White Rabbit performance over fiber/Ethernet.

In this work, we have considered the use of White Rabbit as the main clock synchronization protocol for 5G instances. This contribution may be key for the deployment 5G networks with aim at providing high-performance URLLC. However, White Rabbit can only be used in wired links, thus 5G wireless backhaul will not be able to provide sub-nanosecond time transfer accuracy. Then, we have deeply studied White Rabbit protocol principles and how White Rabbit could be used in wireless communications. Based on this analysis, we propose in this work a wireless White Rabbit design, which reaches sub-nanosecond time transfer accuracy, similar to the performance of wired links.

The article is organized as follows. First, White Rabbit and its integration with PTP are described in Section II. Secondly, the modifications of White Rabbit and the wireless system architecture are detailed in Section III. Section IV presents the simulation setup and the numerical results achieved with the proposed solution, compared with other common approaches to wireless time transfer. Finally, section V summarizes some conclusions of the article.

II. White Rabbit protocol

White rabbit protocol is the combination of three technologies: PTP, Synchronous Ethernet (SyncE) and the Digital Dual Mixer Time Difference (DDMTD). PTP is used for time transfer, SyncE is used to share the Master clock phase and frequency, and the DDMTD is used to increase the timestamps precision. Furthermore, White Rabbit specification also includes some calibration techniques to maximize the protocol performance, such as the link asymmetry calibration. In the following subsections the three technologies and their integration are detailed.

A. PTP

IEEE 1588v2 protocol (PTP), is the de facto standard for high-precision time transfer in point to point wired links, such as Ethernet. The protocol follows a Master-Slave structure. The system has a Grand Master Clock (GMC) that shares its local time with the nodes. PTP time correction process uses four timestamps obtained in a four frame exchange (Fig. 2). The idea behind PTP is to separately estimate the channel delay and the time difference between the Master and Slave clocks.

The frame exchange starts when the Master transmits a PTP Sync frame. Two timestamps are taken in this step, one in the Master side (t_1), which is the Time-of-Departure (ToD) of the frame, and one in the Slave side (t_2), which is the Time-of-Arrival (ToA) of the frame. Just after the Sync frame, the Master transmits a PTP Follow up frame to the Slave, which carriers the t_1 timestamp. It should be noted that the Follow up frame is optional as the t_1 timestamp can be delivered inside the PTP Sync frame if the implementation has this feature. The same process is then done reversely: the Slave transmits a PTP Delay request frame to the Master, and two more timestamps are taken (t_3, t_4). The frame exchange ends when the Master sends the (t_4) timestamp to the Slave through a PTP Delay response frame. When the Slave receives the last timestamp, it performs the PTP corrections to synchronize its local time to the Master time using the calculations stated in (1) and (2).

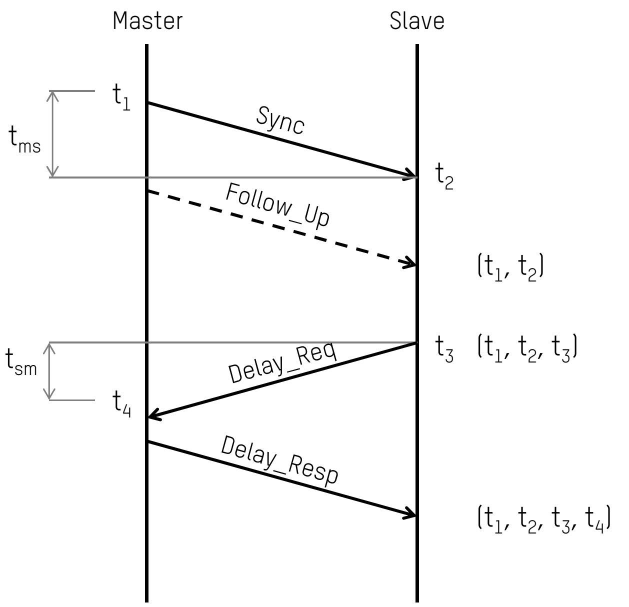

Fig. 2. PTP frame exchange.

$$\tilde{t}_{ms} = \frac{t_2 - t_1 + t_4 - t_3}{2}, \tag{1}$$

$$\tilde{t}_o = t_2 - t_1 - \tilde{t}_{ms}, \tag{2}$$

being $\tilde{t}_{ms}$ the estimated path delay and being $\tilde{t}_o$ the estimated error between the Master and Slave clocks. However, PTP timestamps have a resolution limit, which is related to the timestamping clock frequency. For example, the timestamping module of a common 1 Gbps Ethernet PHY is usually connected to a 125 MHz clock, which has a resolution bound of 8 ns. The two additional technologies introduced in White Rabbit (SyncE and DDMTD) are used to eliminate the resolution bound of PTP and obtain sub-nanosecond accuracy timestamps.

B. Synchronous Ethernet

The Ethernet PHY can be used to transfer the clock phase and frequency from a Master node to a Slave node. This is usually done with the Synchronous Ethernet (SyncE) standard. This technology uses an analog Phase Locked Loop (PLL) to synchronize the Slave clock signal to the Master clock signal. To do so, the Master continuously transmits a 125 MHz signal whenever the Tx PHY is not transmitting a frame and the Slave synchronizes its 125 MHz local clock to that reference signal.

SyncE provides two improvements to PTP: i) the synchronization performance does not depend on the network traffic and, ii) its performance is not significantly degraded in long links.

C. Precise delay calculation and PHY delay calibration using the DDMTD

SyncE eliminates the timestamps variability caused by the frequency drift of the Slave clock. In order to obtain very precise delay calculations, the timestamp resolution bound of eight nanoseconds in common Ethernet PHY layers must be eliminated. To do so, the white rabbit uses the DDMTD. The DDMTD can measure phase differences between two clock signals. This functionality is used to estimate the phase difference between the Master and Slave clocks at the Master side. Then, the value can be used to adjust the PTP timestamps to sub-nanosecond precision. It is worth mentioning that the phase difference in the Slave side is zero, due to the Slave clock is synchronized in phase and frequency to the Master clock.

The DDMTD is also used for precise Ethernet PHY calibration [13]. The Rx and Tx paths of a Ethernet PHY are connected to a DDMTD, as well as the clock signal of the PHY. During the calibration process, a 125 MHz clock signal

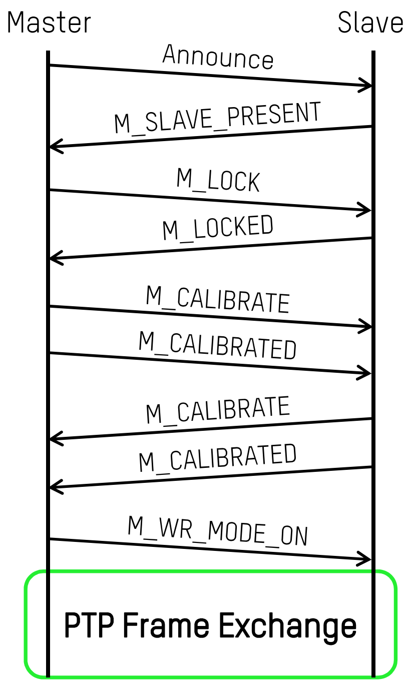

Fig. 3. White Rabbit frame exchange.

is generated at the Tx path of both Master and Slave. The Tx clock is then compared to the PHY clock signal using the DDMTD, achieving a very precise characterization of the Tx delay. This is done at the Rx side as well.

D. White Rabbit as an extension of PTP (WR-PTP)

As mentioned before, White Rabbit uses PTP along with SyncE and three DDMTDs to obtain a very precise time and frequency transfer. Thanks to the customization capabilities of PTP, White Rabbit can be built as a custom extension of PTP by adding White Rabbit mechanisms before PTP starts [13]. Hence, a WR-PTP node will use WR-PTP if their adjacent nodes know White Rabbit, or just legacy PTP if not.

The WR-PTP extension introduces three stages before the start of the PTP operation: i) Slave clock synchronization, ii) Master PHY calibration, and Slave PHY calibration. The message flow is depicted in Fig. 3.

The Master node starts sending a White Rabbit announce message. If the Slave node recognizes the White Rabbit announce message, it sends a M_SLAVE_PRESENT message and the White Rabbit frame exchange starts. First, the Slave node must synchronize its local clock to the Master clock using SyncE. This is done during the M_LOCK / M_LOCKED exchange. Afterwards, the Master sends a M_CALIBRATE frame to start the calibration of its PHY delay. The calibration ends with the M_CALIBRATED frame. The Slave node also performs its PHY calibration after the Master node. Finally, the Master node sends a M_WR_MODE_ON frame and the PTP Frame exchange starts.

III. Wireless White Rabbit

A modification of the White Rabbit architecture for wireless systems is presented in this section. The main challenges of the designed architecture are related to the clock frequency and phase transfer, as SyncE cannot be directly used in wireless communications. The system architecture is depicted in Fig. 4. This wireless white rabbit solution is designed for point to point communications with LOS and low multipath. Hence it may be applicable to communication systems working in any frequency band if the LOS and low multipath conditions are met.

A. From Synchronous Ethernet to Synchronous Wireless

As previously stated, there is not a common standard or implementation to share the frequency and phase of a clock in a wireless network. A first approach could be sending a clock signal in a similar way to how it is done in Ethernet systems: the Master transmits a sequence of zeros and ones, and the Slave locks its local clock with the Master one. However, the effects of the pulse-shaping filter in the clock signal, the absence of a DC component in RF communications, the huge waste of radio resources and the fact that the communication stack would be violated makes this option unfeasible in real wireless implementations. Therefore, neither SyncE nor DDMTD, which depends on SyncE, can be used in wireless White Rabbit.

A suitable option could be the use the incoming frames to recover the Master clock phase and frequency. Indeed, most wireless communication systems use a Symbol Timing Recovery (STR) algorithm for this purpose and sample the incoming data at the exact instant. The precision of the STR depends on the wireless system and its implementation. Regarding the implementation of White Rabbit over wireless, there exist several algorithms to perform symbol timing recovery. Some of them are: the early-late detector, the Mueller-Muller, and the Zero Crossing Detector (ZCD) [14]. The early-late detector is a non-data-aided detector, thus its performance is overcome by decision-directed detectors. The Mueller-Muller detector and the ZCD detector are both decision-directed and have a similar performance. The ZCD detector has been used in this paper mainly because it does not require prior phase recovery and hence it is more robust against errors in the initial estimation. The chosen STR algorithm can be fully implemented on the digital domain, or may be implemented half digital and half analog. A mixed digital/analog implementation could be suitable for this design because the STR algorithm would directly act over the Phased Locked Loop (PLL) of the local oscillator. However, in

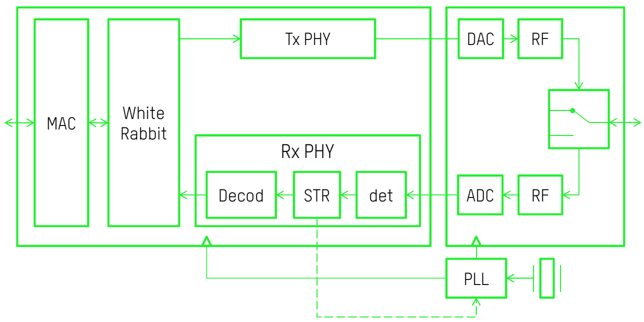

Fig. 4. Wireless system Architecture.

practical implementations, the STR output could be too slow and the feedback error at the STR could cause a mutual interference between the digital circuitry and the local oscillator [15]. On the other hand, the digital implementation uses a free running oscillator, and the samples are regenerated with a digital interpolator. When the frame ends, the STR information can be used to adjust the PLL. Due to the aforementioned reasons, an all-digital receiver seems to be a better option. Hence, the proposed architecture in this paper is an all-digital receiver that uses a digital implementation of the ZCD algorithm.

B. High precision timestamps using the STR

As stated in subsection III.B., the STR algorithm of a wireless receiver is a suitable substitute of the DDMTDs to estimate the transmitter clock phase and obtain very precise timestamps with sub-nanosecond accuracy. Therefore, whenever a frame is received, the frame ToA will be adjusted based on the phase information provided by the STR algorithm.

C. Tx and Rx paths startup calibration

Tx and Rx calibration are needed in most Ethernet PHY implementations in order to correct the uncertainty of the Ethernet DACs and ADC parallel to serial interface start. However, DACs and ADCs of most wireless implementations do not use a parallel to serial interface. Therefore, Tx and Rx paths startup calibration is not required, as the samples are deterministically sent to the RF system.

IV. Simulation Setup and Results

The system used to test the performance of the protocol is a single carrier wireless system. It has a total bandwidth of 28 MHz. The system works in the 2.4 GHz band and the carrier frequency has been set to 2.45 GHz. The frames have two training sequences: a preamble used for frequency offset correction, and a synchronization word used for frame start detection. The preamble comprises 16 symbols and the synchronization word comprises 64 symbols. The synchronization word is used to obtain the initial estimation of the symbol phase. Its shape is a Golay code of 64 symbols [16] with good autocorrelation properties. The amount of data transmitted in each PTP frame was set to 38 bytes modulated in QPSK. The system uses a raised cosine pulse-shaping filter

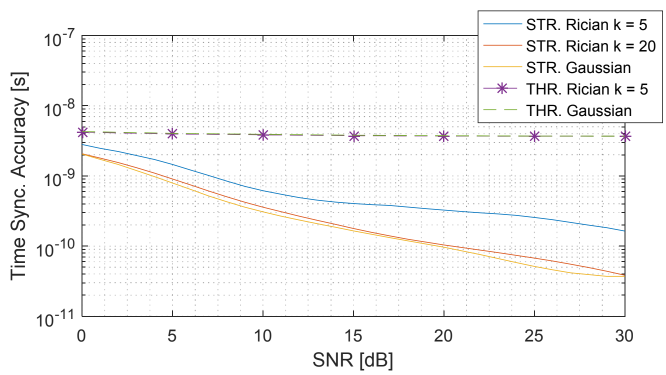

Fig. 5.Time Synchronization accuracy in LOS point to point wireless link.

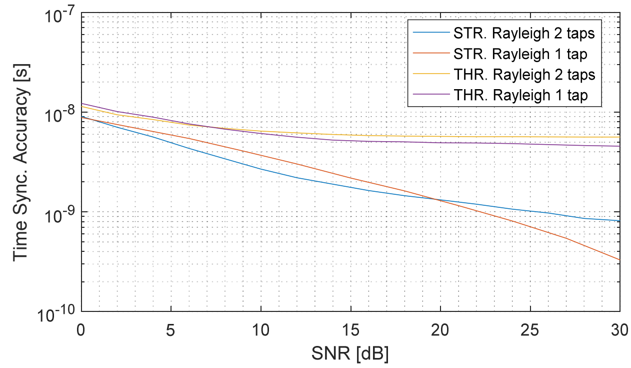

Fig. 6. Time synchronization accuracy in Non-LOS wireless link.

with a Roll-Off Factor of 0.2. The STR algorithm used in the simulation is the ZCD algorithm using 2 samples per symbol and a normalized bandwidth of 0.01.

Regarding the PTP configuration, the frame exchange has been set to 1 s. The PTP delay request are always sent 1 ms after the sync frame. The PTP Proportional-Integral (PI) Loop filter was configured with K_i and K_p constants equal to 0.17 and 0.35 respectively.

The clock drift was randomly set from -5 ppm to 5 ppm to both transmitter and receiver using a uniform distribution. The clock drift has been considered constant for each iteration.

The system has been designed for LOS conditions. However, the simulations were carried out over Non-LOS (NLOS) conditions as well.

The LOS channels configuration was:

- Rician fading channel with $k = 5$.
- Rician fading channel with $k = 20$.
- Gaussian channel.

The NLOS channels configuration was:

- Rayleigh fading channel with 1 tap.
- Rayleigh fading channel with 2 tap, with a gain of -3 dB each and spaced 30 ns.

We have considered a scenario with some elements moving at a speed of 10 km/h close to the communication link. Thus, to simulate the time-varying behavior of the channel in this scenario, we have used a Doppler spectrum with Bell shape and with a Doppler frequency obtained from the speed of the nodes and the carrier frequency. It is worth mentioning that the Doppler frequency does not cause a significant effect in the synchronization result over LOS conditions, because the channel delay is fixed.

The system performance has been compared with an approach only based on the frame start detection (named as THR), such as [10], where the timestamp is taken using only the frame start detection index. In this case, the phase and frequency information of the STR is not used.

The time synchronization accuracy is depicted in Fig. 5 and Fig. 6 for LOS and NLOS channels respectively. The results show significant improvements in the time synchronization accuracy with regard to frame start detector approaches at

LOS conditions at $SNR > 10\ dB$ for all LOS channels. The results also show that the time synchronization accuracy of the THR detector has little improvement at high SNR, because the main source of error in this approach is the error of the start detection.

Regarding the results over NLOS channels, the system performance is highly deteriorated. This is quite significant in the 2 tap channel, mainly because the STR algorithms do not work well under time-dispersive channels. The performance of the Rayleigh channel is still acceptable, because it is a non-dispersive channel, although it is commonly classified as an NLOS channel. Despite these facts, the time synchronization accuracy of the proposed architecture is still better than the THR approach in NLOS conditions.

V. Conclusions

The time synchronization requirements and challenges in 5G networks are presented in this paper. We have stated in this paper that time synchronization in wired backhaul is not a technological challenge, and we propose to use white rabbit protocol to deliver sub-nanosecond time synchronization accuracy. On the other hand, white rabbit cannot be used in wireless communications, thus time synchronization in wireless backhaul is still a technological challenge, especially in 5G networks for URLLC, where strict latency and synchronization requirements are needed. Based on this fact, we have analyzed white rabbit, and we have redesigned the protocol so it can be used in wireless communications as well. The protocol and architecture are not strictly related to any wireless system and can be easily adapted to most point to point wireless systems. It is worth mentioning that the wireless white rabbit solution stated in this paper could be used in other applications where precise time transfer over tens of km are needed, such as in CERN experiments.

The simulation results show that the designed architecture can provide sub-nanosecond time transfer over LOS wireless links, similar to the performance of White Rabbit over Ethernet. The simulation results also showed that the time transfer accuracy is severely deteriorated over NLOS links with time-dispersion, because the STR algorithms does not perform well in this kind of channels.

In future research we will analyze possible enhancements to the architecture to deliver sub-nanosecond time transfer over NLOS wireless links.

Acknowledgment

This work has been partly supported by DETRED (ELKARTEK 2017) project of the Basque Government (Spain) and by ARTEINE (TEC2017-86619-R) project of the Spanish Ministry of Economy and Competitiveness.

References

[1] M. Iwamura, "NGMN View on 5G Architecture," *2015 IEEE 81st Veh. Technol. Conf. (VTC Spring)*, pp. 1–5, 2015.

[2] U. Siddique, H. Tabassum, E. Hossain, and D. In Kim, "Wireless backhauling of 5G small cells: challenges and solution approaches," *IEEE Wirel. Commun.*, vol. 22, no. October, pp. 22–31, 2015.

[3] L. Han, X. Hu, H. Li, L. Wang, and N. Hua, "First Demonstration of Distributed Time Synchronization System over Transport Network towards 5G Requirements," *Opt. Fiber Commun. Conf. Exhib.*, pp. 5–7, 2017.

[4] S. Siu, J. L. Wang, W. H. Tseng, C. S. Liao, and H. F. Hu, "Primary reference time clocks performance monitoring using GNSS common-view technique in telecommunication networks," *18th Asia-Pacific Netw. Oper. Manag. Symp. APNOMS 2016 Manag. Softwarized Infrastruct. - Proc.*, 2016.

[5] H. Kirrmann, C. Honegger, D. Ilie, and I. Sotiropoulos, "Performance of a full-hardware PTP implementation for an IEC 62439-3 redundant IEC 61850 substation automation network," *ISPCS 2012 - 2012 Int. IEEE Symp. Precis. Clock Synchronization Meas. Control. Commun. Proc.*, pp. 43–48, 2012.

[6] "White Rabbit for Industrial Timing Enhancement (WRITE)." [Online]. Available: http://empir.npl.co.uk/write/.

[7] M. Lipinski, "White Rabbit Standardization in IEEE1588-2008," 2018. [Online]. Available: https://www.ohwr.org/project/wr-std/wikis/home.

[8] R. Exel, "Clock Synchronization in IEEE 802.11 Wireless LANs using Physical Layer Timestamps," in *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication, ISPCS*, 2012.

[9] *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, 2012.

[10] Ó. Seijo, I. Val, J. A. López-fernández, and M. Vélez, "IEEE 1588 Clock Synchronization Performance over Time-Varying Wireless Channels," *IEEE Int. Symp. Precis. Clock Synchronization Meas. Control. Commun. ISPCS*, 2018.

[11] "Nanosecond Distribution over a Wireless Link." [Online]. Available: https://esplab.epfl.ch/page-118049-en.html. [Accessed: 20-Aug-2011].

[12] M. Rico, J. P. Aubry, C. Botteron, and P. A. Farine, "Ns-level time transfer over a microwave link using the PTP-WR protocol," *2015 Jt. Conf. IEEE Int. Freq. Control Symp. Eur. Freq. Time Forum, FCS 2015 - Proc.*, pp. 690–695, 2015.

[13] M. Lipiński, T. Włostowski, J. Serrano, and P. Alvarez, "White rabbit: A PTP application for robust sub-nanosecond synchronization," *IEEE Int. Symp. Precis. Clock Synchronization Meas. Control. Commun. ISPCS*, pp. 25–30, 2011.

[14] U. Mengali, *Synchronization Techniques for Digital Receivers*. Springer, 1997.

[15] N. D. Vo and T. Le-Ngoc, "Maximum Likelihood (ML) Symbol Timing Recovery (STR) techniques for reconfigurable PAM and QAM modems," *Wirel. Pers. Commun.*, vol. 41, no. 3, pp. 379–391, 2007.

[16] J. E. M. Golay, "Complementary Series," *IRE Trans. Inf. Theory*, pp. 82–87, 1961.