

Enhancing White Rabbit Synchronization Stability and Scalability Using P2P Transparent and Hybrid Clocks

José Luis Gutiérrez-Rivas , Felipe Torres-González , Eduardo Ros , and Javier Díaz 

Abstract—Time synchronization faces an increasing performance demand nowadays. This is particularly common in different segments, such as new scientific infrastructures, power grid, telecommunications or fifth-generation (5G) wireless networks. All of them share the need for a time synchronization technology offering a very low synchronization error along large networks. The new version of the IEEE-1588-2019 protocol offers a considerable performance improvement thanks to the high accuracy profile based on white rabbit (WR). The original WR implementation guarantees a synchronization accuracy below 1 ns for a small number of cascaded nodes. This contribution evaluates the performance degradation when a considerable number of devices are deployed in cascaded configurations. In order to improve the performance, different delay measurement implementations have been evaluated in a series of experiments. The results prove the considerable benefits of our implementation against the current WR implementation.

Index Terms—Accelerators, fifth generation (5G), precision time protocol (PTP), scalability, stability, synchronous ethernet (SyncE), telecom, white rabbit (WR).

I. INTRODUCTION AND RELATED WORK

TIME synchronization has become key (first-order concept) in many scientific and industrial infrastructures. Distributed measurements, data analysis, and service provision require accurate time references in order to ensure precise results in the framework of distributed facilities. In addition, the evolution of the technologies related to accelerators, power grid, and telecommunications has pushed the enhancement of timing protocols to meet the demanding requirements of these

segments. This motivates the use of different protocols such as inter-range instrumentation group B (IRIG-B), network time protocol (NTP), or precision time protocol (PTP) (with different performance profiles) depending on the requirements imposed by each domain. PTP includes specific profiles to meet the needs of each infrastructure and ease the deployment of its timing network. For the most demanding applications, the description of a new high accuracy profile for IEEE 1588-2019 [1], [2] has been recently approved in 2019, taking, as a basis, the white rabbit (WR) technology.

Concerning smart grid networks, they are composed of multiple interconnected nodes in cascade and parallel configurations [3], [4]. For this reason, synchronization accuracy must be evaluated accordingly to determine the maximum number of hops ensuring the time accuracy requirement at the last network elements. In this field, according to IEEE standards, the time error for phasor measurement units (PMUs) need to be below 1 μ s, but recent studies point out a desirable accuracy below 10 ns [5]. The timing solution most globally used in smart grid is the global navigation satellite system (GNSS) because of its high availability, but it is vulnerable to accidental or malicious interferences (spoofing or jamming satellite signals), thus representing a thread for this critical infrastructure. IEEE recommendations and other works [6], [7] focus on providing an alternative method to GNSS by using terrestrial systems [8]. In this regard, the integration of the WR (wired technology) in smart grid communication networks resolves the vulnerability of GNSS and covers the forthcoming strict synchronization needs introduced by the utilization of PMUs and long-cascade configurations [5], [9], [10].

In the automation industry, hard real-time systems like control applications, where data transmission is crucial for the proper functioning of the system, require of isochronous real-time (IRT) communications. These IRT communications coordinate data exchange between nodes to achieve a deterministic real-time behavior. IRT is typically characterized by cycle times of less than 1 ms and jitter less than 1 μ s [11]. In protocols such as Profinet IRT and time-sensitive networks (TSN), this is improved by the integration of an extremely accurate, shared clock, using PTP [12]. In this regard, it should be noted that the utilization of more accurate synchronization protocols like WR may improve the cycle times and jitter in IRT communications, thus improving data transmission determinism in industrial real-time applications.

Manuscript received August 3, 2020; revised October 24, 2020 and December 28, 2020; accepted January 17, 2021. Date of publication January 25, 2021; date of current version July 26, 2021. This work was supported in part by EMPIR programme cofinanced by the Participating States and in part by European Union's Horizon 2020 Research and Innovation Programme (WRITE) under Project 17IND14. Paper no. TII-20-3782. (Corresponding author: José Luis Gutiérrez-Rivas.)

José Luis Gutiérrez-Rivas is with the R&D Department, Seven Solutions S.L, 18014 Granada, Spain, and also with the Department of Computer Architecture and Technology, University of Granada, 18071 Granada, Spain (e-mail: jlgutierrez@sevensoles.com).

Felipe Torres-González, Eduardo Ros, and Javier Díaz are with the Department of Computer Architecture and Technology, University of Granada, 18071 Granada, Spain (e-mail: torresfelipe1@gmail.com; eros@ugr.es; jda@ugr.es).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TII.2021.3054365>.

Digital Object Identifier 10.1109/TII.2021.3054365

In the framework of cellular networks, time and phase synchronization is key. For example, frequency division duplexing for call initiation, time division duplexing for time slots alignment, long-term evolution advanced (LTE-A), enhanced intercell interference coordination for interference coordination, coordinated multipoint, or multiuser multiple-input and multiple-output [13]. Timing needs are moving from frequency synchronization to time and phase synchronization, evolving from the old requirement of $\pm 1.5 \mu\text{s}$ for the fourth generation of broadband cellular network technology (4G/LTE), similar to fifth generation (5G) Phase 1, aiming to ns for Phase 2 and beyond 5G developments [14]. In this regard, IEEE P802.1CM describes A+ networks or common public radio interface protocols that demand an accuracy better than 12.5 ns [13], [15], [16]. In terms of scalability, the most restrictive time error (class B) requirement in cascade configurations formed by 21 hops [17] is proposed to be 420 ns constant time error, where each of them cannot exceed an offset of 20 ns [18]. Making use of an ultra-accurate time transfer approach facilitates scalability across the network.

Following the timing requirements and recommendations from the industrial and telecom domains previously discussed, this article has two main scientific goals. First, we will evaluate the impact of the computational delay model on the timing accuracy. It is known that the IEEE-1588 protocol may use different computational schemes to adapt the distribution of time to the characteristics of the network. To this end, we have adapted WR to the peer-to-peer (P2P) and end-to-end (E2E) computational delay models and evaluated their impact on the accuracy in order to achieve the best synchronization accuracy and interoperability with other IEEE-1588 profiles. Second, as suggested by ITU-T G.8271.1, telecom networks need to be able to deploy cascade chains with more than 21 nodes, with a synchronization accuracy better than 420 ns for the whole chain on the most demanding case, class B [17]. We will evaluate the impact of large cascade chains using these computational delay models with WR, included on the high accuracy profile extension, in order to evaluate the impact on the accuracy depending on the length of the cascade. In summary, we will extend the interoperability, cascade capabilities, and accuracy of the standard WR implementation. Hence, providing useful insights to extend the features of this protocol for the previously described domains.

The rest of this article is organized as follows. Section II summarizes the default WR implementation and its main features and characteristics. Section III describes the developments carried out to improve WR scalability and stability. Section IV shows the results of these developments. Finally, Section V concludes this article.

II. WHITE RABBIT INTRODUCTION

The basis of this work relies on the WR technology [19], taking as a starting point its default implementation. WR, distributed under an open license, was born at the European Organization for Nuclear Research (CERN) as an Ethernet-based technology to synchronize devices with an accuracy better than

1 ns in scientific facilities such as accelerators and colliders. It is based on three elements: an extension of IEEE 1588 PTPv2, the distribution of frequency using a Layer 1 (L1) syntonization mechanism similar to synchronous Ethernet (SyncE), and the measurement of the phase offset using dual digital mixer time difference (DDMTD) components so as to improve the timestamps accuracy. WR main features are as follows:

- 1) sub-ns synchronization;
- 2) connecting thousands of nodes;
- 3) typical distances of 10 km between nodes but extensible beyond 100 km;
- 4) Ethernet-based Gigabit rate reliable data transfer;
- 5) open hardware, firmware, and software.

WR devices are nowadays implemented as ordinary clocks (OC) and boundary clocks (BC), which perform the estimation of the link delay and the synchronization hop-by-hop using a master-slave hierarchical architecture using a two-step E2E delay model to propagate the clock. For this, *Delay-Request* messages are used to estimate the delay between them. Each device recovers the clock from its proceeding master frequency reference using an L1 frequency distribution approach, and after estimating the delay to the master, it computes the offset to the master using PTP frames. Regarding scalability, E2E studies have stated that this mechanism increases both jitter and skew of the synchronization signals with the number of hops [20].

WR basis for syntonization, phase difference measurement, and synchronization are presented below.

A. WR Layer 1 Frequency Distribution

SyncE uses the physical layer to transmit timing similarly to synchronous optical network (SONET)/SDH. It provides a mechanism to transfer frequency over Ethernet networks that can be traceable to global positioning system references. In contrast to standard SyncE, WR devices do not propagate the received clock immediately; they use their local oscillator to transmit the frequency reference. In addition to this, frequency/phase of the local oscillator is influenced by WR-PTP since it controls the L1 clock in contrast to standard SyncE+PTP implementations, where PTP only transmits the time and does not influence the phase of the local oscillator. Therefore, time of a sending node is only propagated to the directly linked node.

B. WR Phase Recovery: Improving Hardware Timestamps

In PTPv2 (with hardware timestamps), timestamps are bound to one period of the reference clock (8 ns). In order to improve this resolution, WR uses DDMTDs to achieve a synchronization accuracy below one period of the reference clock. Initially, the slave device has recovered the master's clock reference from the link and has disciplined its main local clock to follow the master's clock (L1 syntonization). Despite both clocks working with the same frequency, the slave's clock is delayed an unknown amount of time due to the propagation of the master's clock to the slave over the link. Then, the DDMTD method is used to improve the resolution of the measurements of the delay.

The delay measurement is performed using field-programmable gate array (FPGA) logic elements. These are limited by the clock used to drive them, which is the reference clock. The DDMTD modules use a third clock signal derived from the main local clock reference (helper clock). A known offset is applied to the helper clock. Then, it is used to sample the recovered clock (master's reference) and the local clock (slave's reference). The resulting signals from the sampling are running at a much lower frequency than the input clocks. However, the phase relationship between them is equivalent, i.e., the delay between the input clocks is the same as the delay of the output signals. In particular, for WR, the resulting signal frequencies are in the kHz range. Time delay between signals in the kHz range can be measured in a straightforward way using regular FPGA logic elements. This measurement is later used by WR to increase the accuracy of the timestamps, thus achieving the sub-ns synchronization accuracy. A detailed explanation about the theory behind this technique is included in [21].

The high accuracy measurements from the DDMTD modules are used in both ends. On the slave, the measurement is locally used for the $\text{offset}_{\text{ms}}$ computation. On the master side, this phase difference is also measured and included in WR PTP frames' correction field, which will be later retrieved by the slave and introduced in the previously mentioned computation.

C. White Rabbit Precision Time Protocol (WR-PTP)

IEEE-1588 describes two different ways to estimate the offset and delay errors between two different clock devices: E2E and P2P. In E2E, the latency of the network is computed directly between the master and the slave without taking into account the types of network devices that are deployed along the link path. On the other hand, P2P computes the delay between the egress of the upstream node and the ingress of the downstream node instead of computing the delay of the whole network at once. PTP messages are sent down the network and the residence delay is accounted for in each node and transmitted to downstream devices to compensate the delay of the whole link path taking into account all residence delays of each device of the link path.

In industrial networks, the accuracy achieved through a network implementing E2E will not be as good as the same topology implementing P2P with transparent clocks. E2E generates a larger volume of network traffic when multiple slaves being present on the network which can be a significant processing load to the master device. Alternatively, P2P is more efficient and is less affected by asymmetry within the network. When multiple slaves are present in a network, peer delay packets are only sent to the nearest node, not all the way upstream to the master. Finally, if the network is suddenly changed due to a fault or similar, recovery time is reduced as not all delay calculations will be affected [22].

In view of these considerations and taking into account that WR implements the E2E approach, the evaluation of a WR P2P implementation may show better synchronization results for WR. In the following lines, we explain the WR-PTP implementation and determine if the computation model currently used

has any impact on the performance achieved or, alternatively, the accuracy can be improved by implementing a P2P approach.

WR-PTP is implemented as an Open Source PTP daemon called PPSi. By default, PPSi uses two-step E2E clocks and measures the delay between two clocks using the *Delay-Request* mechanism. The offset between the master and the slave is modeled by (1). All equations described in this manuscript are based on IEEE 1588 and have been adapted for WR

$$\text{offset}_{\text{ms}} = t_2 - t_1 + \text{delay}_{\text{ms}}. \quad (1)$$

Values t_2 and t_1 are the timestamps for the *Sync* message in master's and slave's local time references, respectively. The $\text{offset}_{\text{ms}}$ is affected mainly by two components: hardware delays and media propagation delays (delay_{ms}). Two different wavelengths are used in WR to establish a full-duplex connection over an optic fiber cable. In addition to that, we need to consider the difference between the internal transmission (Tx) and reception (Rx) paths in the hardware. The global existing asymmetry is modeled by the the following equation:

$$\text{asymmetry} = \Delta_{\text{tx}_m} + \Delta_{\text{rx}_s} - \frac{\Delta - \alpha\mu + \alpha\Delta}{2 + \alpha} \quad (2)$$

where Δ indicates the summation of all the fixed hardware delays, Δ_{tx_m} is the Tx fixed delay for the master, Δ_{rx_s} is the Rx fixed delay for the slave, and α indicates the relation between the Tx and Rx wavelengths, which is a constant value, experimentally calculated following the procedure included in [23]. Finally, delay_{ms} is given by

$$\text{delay}_{\text{ms}} = \mu + \text{asymmetry} \quad (3)$$

$$\mu = ((t_4 - t_1) - (t_3 - t_2))/2. \quad (4)$$

Fig. 1 depicts a complete WR-PTP message exchange between a master and a slave node. The first step is to perform the syntonization of the reference clock using the L1 WR link initialization. Due to the asymmetry of hardware components, this process includes the exchange of the different asymmetries on both sides, master and slave (3). Once the syntonization process is over, PTP starts computing the asymmetry of the link using the round-trip delay and continues measuring the phase difference thanks to the utilization of the previously described DDMTDs. PTP synchronization is carried out regularly, adjusting the slave oscillator by tracking the changes on the phase ($\text{offset}_{\text{ms}}$).

After introducing the default E2E WR implementation, this article presents the development and results of the P2P delay model for the WR protocol in order to improve the synchronization performance in terms of scalability and stability. This involves the development of two new types of P2P WR clocks: transparent (TC) and hybrid (HY) clocks based on the current software stack implementation of PPSi using a portable operating system interface (POSIX) implementation, in contrast to previous ones (no-POSIX) [24], thus facilitating the integration of these features in current WR devices. Moreover, this article includes a longer deployment experiment and results in comparison to [24] and [25], deploying daisy-chains cascades composed of up to 19 nodes. Section III focuses on these developments and the results obtained, being significantly better than the results of the default E2E WR implementation. These results clearly help

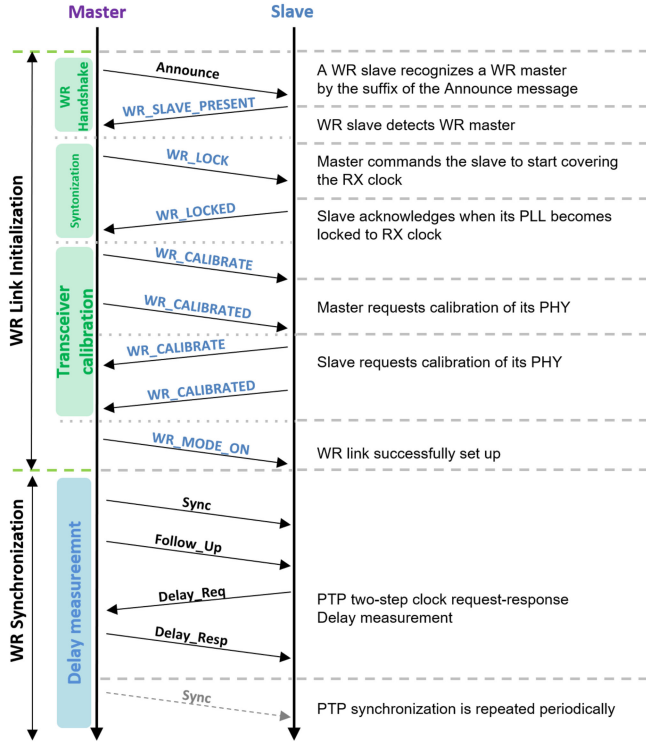


Fig. 1. Complete PTP message flow during WR synchronization. The WR link initialization process is performed only once as soon as the link has been established. The WR synchronization process is repeated every second after the link has been initialized. Announce messages are sent every 2 s, whilst Sync, Follow_up and Delay_Req messages are sent every second. Figure inspired from [21].

to differentiate which computation delay model is a better fit for a high-accuracy timing protocol.

III. WR TRANSPARENT AND HYBRID CLOCKS

Whilst E2E is used in scientific and telecom networks (ITU-T G.8265 [26] and ITU-T G.8275.1 [27]), P2P is mainly used in engineering (power profile [28]) networks where all nodes are known to be IEEE 1588 compatible. In this type of networks, PTP frames are sent from the master to the slave node, being forwarded by intermediate nodes like switches and routers, considering the entire network as a simple fiber link. Fig. 2 depicts both E2E and P2P delay models.

The development of WR TC/HY involves the implementation of mainly two mechanisms: a P2P mechanism to send *Announce*, *Sync*, and *Follow_Up* frames from the master to the slave, and a *Peer-Delay* mechanism to estimate the delay between neighbor devices. Furthermore, WR requires the dissemination of the frequency over L1 too. This makes a significant difference compared with industrial IEEE-1588 protocol implementations, generating new problems and challenges that must be overcome to integrate such mechanism into the WR-PTP stack.

Sections III-A and III-B present the differences and similarities compared to the default WR implementation. Fig. 3 describes the diagram blocks of our implementation of WR BCs, TCs, and HYs.

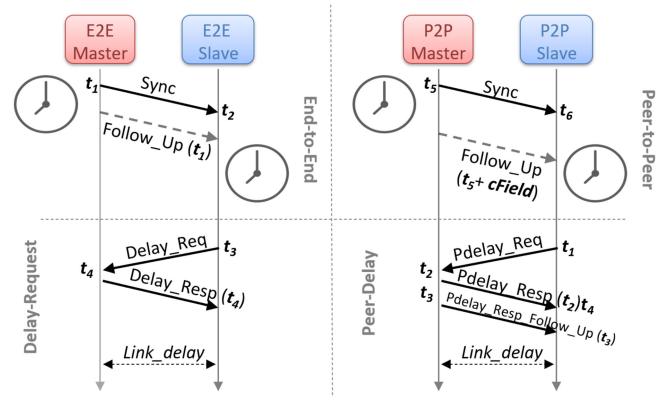


Fig. 2. E2E delay model uses four time-stamps and the *Delay-Request* mechanism to compute the offset between the master and the slave node (left image). The P2P delay model computes this delay using six timestamps and the *Peer-Delay* mechanism (right image).

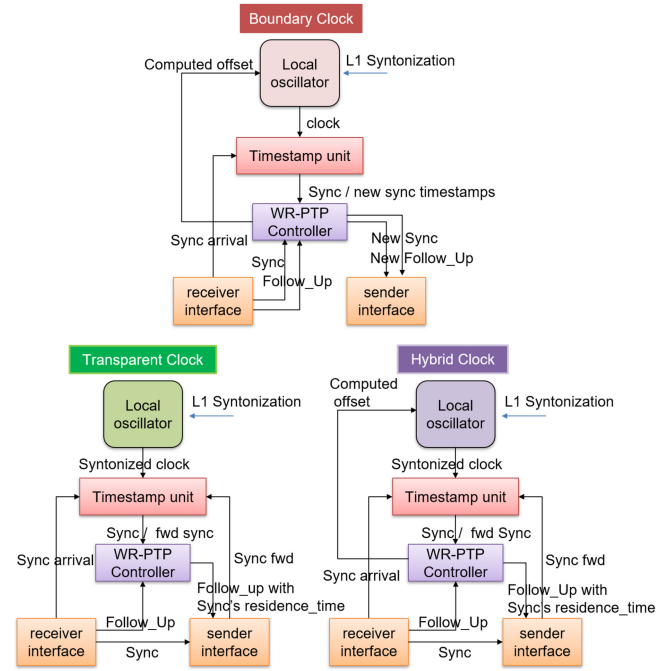


Fig. 3. Block diagrams for BCs, TCs, and HYs. BCs compute the $offset_{ms}$, adjust the local oscillator, and create new PTP messages that are sent to the next node. TCs forward the incoming PTP frames to the next node without adjusting the local oscillator. HYs, after adjusting their local oscillators, forward the PTP frames to the next node (HYs do not generate new PTP frames as BCs do).

A. Syntonization on WR P2P Clocks

In spite of the utilization of TC or HY in P2P networks, the distribution of the frequency from the master to the slave must also be forwarded through intermediate nodes. This is due to the fact that final P2P slave nodes (HYs) need master's frequency to reach the WR sub-ns synchronization. For this reason, the WR syntonization for TC/HY has been realized in the same way it is carried out for BCs.

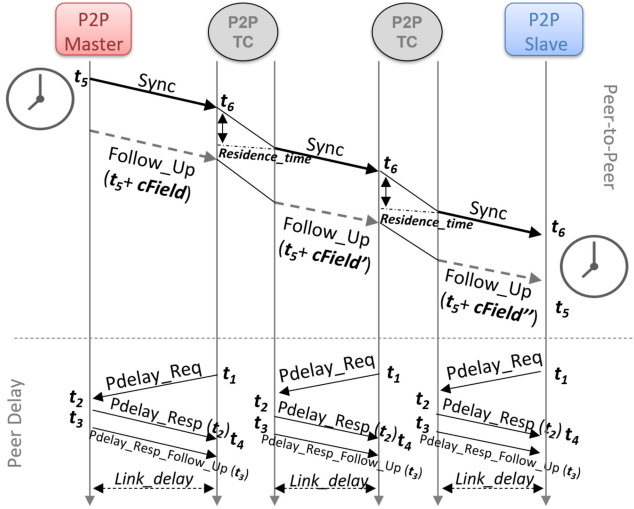


Fig. 4. Clock offset measurement using WR TC/HYs. Master sends *Announce*, *Sync*, and *Follow_Up* frames to the slave through TC/HYs. Each time a *Sync* goes through a TC/HY, its residence time inside the device is measured and, together with the link delay of the incoming port, it is added to the *Correction_Field* (cField) of the next *Follow_Up*.

B. WR Synchronization on WR P2P Clocks

In contrast to E2E WR clocks, P2P devices send PTP frames from the master to the final slave through TC or HY intermediate nodes. These nodes forward *Announce*, *Sync*, and *Follow_Up* frames instead of generating them locally. In addition, they compute the link delay using the *Peer-Delay* mechanism instead of the one used by BCs (*Delay-Request*). These concepts are detailed below.

Peer-Delay measures the delay of the link between two adjacent nodes using four timestamps (t_1 , t_2 , t_3 , and t_4) as Fig. 4 shows. It uses three types of messages: *Pdelay_Req*, *Pdelay_Resp*, and *Pdelay_Resp_Follow_Up*. First, t_1 corresponds to the moment a node sends a *Pdelay_Req* to its adjacent node, and t_2 is generated in the other node as soon as *Pdelay_Req* is received. This second node responds with a *Pdelay_Resp* message and generated t_3 , which is received in the requester as t_4 . Previous t_3 is received immediately after in a *Pdelay_Resp_Follow_Up* message. The receiver uses the timestamps to calculate the delay as follows:

$$\text{delay}_{\text{node,node}} = (t_2 - t_1 + t_4 - t_3)/2. \quad (5)$$

In P2P, in order to measure the clock offset, *Announce*, *Sync*, and *Follow_up* frames are sent from the WR master node to the slave, considering all intermediate nodes of the network as TCs or HYs, where these frames are just forwarded as indicated in Fig. 4. The utilization of this clock offset measurement method involves a clear improvement with respect to the E2E method since it mitigates the error propagation on each hop of the network. Due to the experimental errors that occurred in the WR calibration process together with an inherent addition of jitter per hop in a E2E WR network, each slave measures its clock offset with respect to a less precise copy of the original clock. This represents an addition of a significant noise component per hop.

This effect degrades quickly the accuracy in timing networks composed of many hops in cascade configurations [29].

A slave node is not aware of the delay from the master since its delay is estimated with the current neighbor nodes using *Peer-Delay*. For this reason, it is necessary to keep track of the delay accumulated by all the links/nodes of the network path. This is performed using the *Correction_Field* (cField) of *Follow_Up* frames.

When the master sends a *Sync* message, t_5 is generated and sent in the next *Follow_Up*. When a *Sync* message is received in a TC, a timestamp $t_{\text{sync_ingress}}$ is generated and the message is immediately forwarded to the other active ports generating a $t_{\text{sync_egress}}$ timestamp. These two timestamps are used to calculate the *Residence_Time* (6) of each *Sync* message on each of the outgoing ports. In addition, cField must also add the link delay computed in the incoming port.

$$\text{Residence_Time} = \text{sync_egress} - \text{sync_ingress} \quad (6)$$

$$\text{cField} = \text{Residence_Time} + \text{link_delay}. \quad (7)$$

When the end node receives a *Sync* message, it generates t_6 and waits for the *Follow_Up*, which contains t_5 and the total link delay in the cField. By applying the following equation, the slave adjusts its local clock to the master reference:

$$\text{offset}_{\text{ms}} = (t_6 - t_5 + \text{cField} + \text{link_delay})/2. \quad (8)$$

We have implemented WR P2P TCs and P2P HYs, making use of these mechanisms. The main difference between WR P2P TCs and P2P HYs is that a TC does not compute the $\text{offset}_{\text{ms}}$ to adjust its local time (time counter and phase) to the master reference, while a HY does. An HY computes the $\text{offset}_{\text{ms}}$ after forwarding the received PTP frames to the next node and applies the changes to its local oscillator. In this way, HYs synchronize and synchronize to the master reference while TCs only implement synchronization.

Regarding precision problems caused by timestamps generation, note that WR uses hardware timestamps that are generated immediately before/after a PTP frame is sent/received so that the uncertainty that could be introduced by the utilization of these timestamps at higher levels of the open systems interconnection (OSI) model are reduced to despicable values.

Section IV presents a performance evaluation of both delay measurement mechanisms for a WR network. In this contribution, we have stated the benefits of using P2P instead of E2E in terms of performance. The next section addresses the experimental demonstration of the following hypotheses: 1) Offset accuracy suffers less degradation in P2P networks because it measures directly the offset difference with the master of the network; 2) the jitter that is propagated over the network (directly related to scalability) is improved in TC configurations since these devices only synchronize their local oscillators instead of synchronizing to the master reference.

IV. RESULTS

A setup composed of a cascade of 20 WR lite embedded nodes (WR-LEN) provided by Seven Solutions SL has been deployed in order to evaluate these hypotheses.

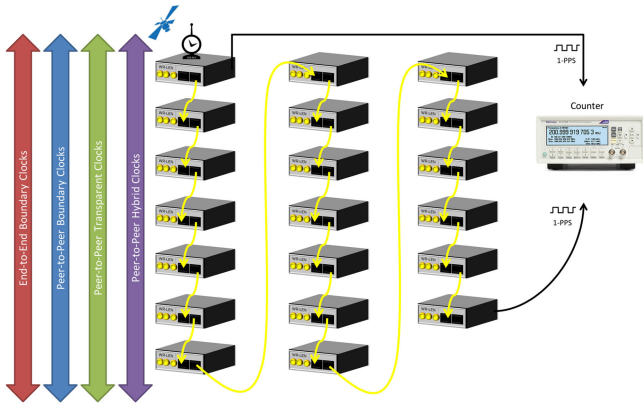


Fig. 5. WR-LEN daisy-chain design composed of 20 WR-LENs used to evaluate the scalability and stability of different WR implementations. The first node is configured as master and the rest as follows: E2E BCs (red), P2P BCs (blue), P2P TCs (green), and P2P HYs (purple). This color coding is used in subsequent result plots.

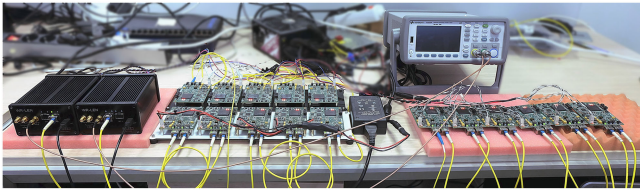


Fig. 6. WR-LEN daisy-chain setup in lab.

These devices are the most simple WR nodes with two ports that allow daisy-chain configurations. In addition, they share clock components with the WR switch (WRS). Although a WR-LEN includes an Artix-7 low-end FPGA and a WR PTP Core (dual port) architecture, the results obtained are comparable to using other WR devices such as the WRS.

The laboratory setup presents a daisy-chain configuration over 0.5 m fiber links, in which N_{00} is the master of the entire network; the rest are E2E BC slaves, P2P BC slaves, P2P TCs, or P2P HYs. Fig. 5 depicts this setup, while Fig. 6 shows the real deployment. The experiments have been performed in lab, under temperature conditions of 25°C using short fiber links so that thermal effects on propagation asymmetry are negligible.

Four scenarios have been deployed to evaluate the scalability and the performance of the approaches developed: E2E BCs with *Delay-Request*, P2P BCs with *Peer-Delay*, P2P TCs with *Peer-Delay*, and P2P HYs with *Peer-Delay*. Measurements have been carried out at the following slaves nodes: N_{01} , N_{03} , N_{07} , N_{11} , N_{15} , N_{17} , N_{18} , and N_{19} . Not all the evaluated approaches were able to fully synchronize the 19 slave nodes. On each scenario, the measurements are presented until the last fully synchronized slave. The device used to compare 1-pulse per second (PPS) outputs ($\text{offset}_{\text{ms}}$) is a Keysight 53230 A universal frequency counter/timer, which presents a resolution of 20 ps.

The accuracy at each spot is evaluated measuring the time interval between the rising edge of the PPS signal of the master and the rising edge of the the PPS signal of the n th slave node. These signals include a relevant component of random noise; hence, we decided to average 180 consecutive samples of the

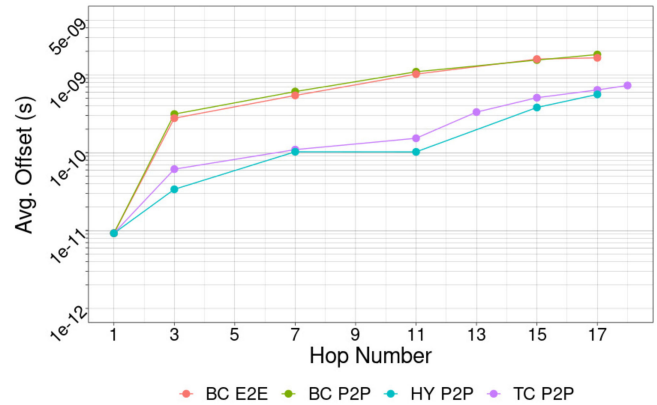


Fig. 7. Averaged PPS offset measurements for E2E BCs (orange line), P2P BCs (green line), P2P TCs (purple line), and P2P HYs (blue line).

TABLE I

$\text{offset}_{\text{ms}}$ RESULTS FOR ALL EVALUATED SCENARIOS. NODES PRESENTING AN ACCURACY ABOVE 1 NS (IN AVERAGE) ARE DEPICTED IN RED; GREEN FOR ONES BELOW 1 NS. ALL VALUES ARE DESCRIBED IN PS (10^{-12} s)

Hop	E2E BC		P2P BC		P2P HY		P2P TC	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ
N_{01}	9	16	9	16	9	16	9	16
N_{03}	277	45	313	43	34	52	61	53
N_{07}	543	25	609	25	103	69	109	43
N_{11}	1020	17	1093	22	102	29	153	43
N_{15}	1590	56	1546	81	380	62	508	94
N_{17}	1651	76	1819	38	560	115	639	87
N_{18}	Not synchronized						728	180
N_{19}	Not synchronized							

time interval between the PPSs. In addition to the averaged value of the time interval, the standard deviation of the measurement is included as well. To prove that our results are repeatable, we made five repetitions of every single measurement, i.e., we completely power cycled the whole chain of devices and waited until all nodes got synchronized without changing any parameter. Then, we averaged the five time interval values and the five associated standard deviation values to produce each one of the single values included in the following tables and graphs.

A. WR E2E BC Using Delay-Request

This is the WR default configuration, in which nodes are slave from their upstream node and master of the downstream. Each node generates PTP frames for the downstream node. The measurement of the delay is carried out using the WR default *Delay-Request* approach. Nodes compute the $\text{offset}_{\text{ms}}$ using four timestamps.

In this scenario, all nodes of the cascade are synchronized and synchronized to the master reference.

Offset accuracy results are summarized in Fig. 7 and Table I. This approach was capable of achieving 17 fully synchronized slave nodes. Nodes as of the 18th are not even able to synchronize to the retrieved frequency, and, thus, synchronization process cannot be started. This is so because of the degradation of the distributed frequency over the physical layer after 17 hops. After

TABLE II

JITTER RESULTS FOR ALL EVALUATED SCENARIOS. VALUES ABOVE SUB-NS RANGE (SEE TABLE I) ARE SHOWN IN RED; GREEN FOR THE ONES BELOW 1 NS. VALUES ARE DEPICTED IN PS (10^{-12} s)

Hop	E2E BC		P2P BC		P2P HY		P2P TC	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ
N_{01}	17.9	1.5	17.9	1.5	17.9	1.5	17.9	1.5
N_{03}	20.5	4.6	23.3	5.2	22.5	3.0	19.0	1.9
N_{07}	34.3	5.0	32.1	4.8	37.5	3.2	24.0	2.0
N_{11}	89.4	5.0	93.3	3.6	95.3	14.4	75.4	9.7
N_{15}	308.7	53.7	334.5	28.2	337.7	44.3	239.7	15.1
N_{17}	642.6	90.0	699.3	92.2	736.2	51.6	441.8	48.5
N_{18}	Not synchronized						711.3	35.8
N_{19}	Not synchronized							

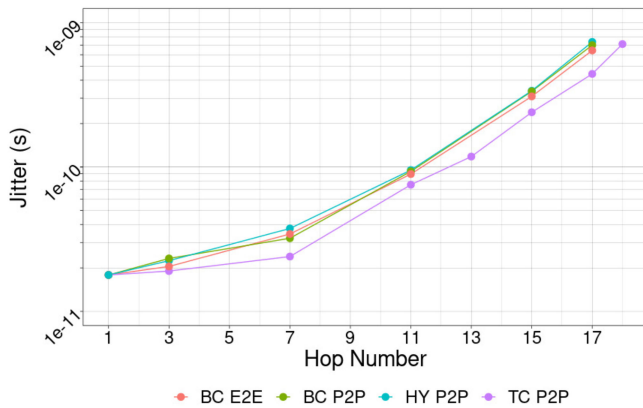


Fig. 8. Averaged PPS jitter measurements for E2E BCs (orange line), P2P BCs (green line), P2P TCs (purple line), and P2P HYs (blue line).

17 slave devices, the quality of the received frequency does not meet the software phase-locked loop quality constraints to perform the syntonization process.

Regarding 1-PPS $\text{offset}_{\text{ms}}$, the results evidence that, in average, the sub-ns accuracy is preserved until hop N_{10} . From N_{11} to N_{17} , in spite of being synchronized using WR, the average of the $\text{offset}_{\text{ms}}$ measured reaches a maximum mean value of 1651 ± 76 ps (σ).

The PPS jitter (Fig. 8) starts to be considerable as of hop N_{11} . This significant increment of the jitter matches the position in the chain where sub-ns accuracy is lost. The highest value is reached at N_{17} , 642 ± 90 ps (σ).

B. WR P2P BC With Peer-Delay

All nodes are slave from their upstream node and master of the downstream ones. Each node generates PTP frames for the downstream node. The measurement of the delay is carried out using the *Peer-Delay* approach. All nodes compute the $\text{offset}_{\text{ms}}$ using six timestamps.

In this scenario, all nodes of the cascade are syntonized and synchronized to the master reference.

Offset accuracy results are summarized in Fig. 7 and Table I. As in the previous case, there are no results for nodes N_{18} and N_{19} since they are not able to recover the L1 frequency from the previous reference.

P2P BCs show similar accuracy results compared to E2E BCs. Averaged sub-ns accuracy is also preserved until hop N_{10} . From N_{11} to N_{17} , the averaged $\text{offset}_{\text{ms}}$ measured reaches a maximum value of 1819 ± 38 ps (σ).

C. WR P2P TC With Peer-Delay

The first node of the cascade is a P2P master clock and the last one is a P2P slave. Intermediate nodes were set as TCs so that they only forward PTP frames from their upstream to the downstream port. The last node computes the $\text{offset}_{\text{ms}}$ using six timestamps.

In this scenario, only the last node of the cascade is syntonized and synchronized to the master reference. Intermediate nodes are only syntonized.

Synchronization accuracy results are summarized in Fig. 7. There are no results for N_{19} since it was not possible to syntonize this node to the retrieved reference either, as occurred for E2E and P2P BCs scenarios. P2P TCs present very satisfactory results, achieving an accuracy below 1 ns for the entire cascade, reaching a maximum averaged $\text{offset}_{\text{ms}}$ of 728 ± 180 ps (σ) for N_{18} . In Fig. 8, a considerable reduction of the jitter for the TC scenario is observed. Jitter is nearly constant through the seven first nodes. Then it starts to rise as occurred with the rest of the experiments but presenting significantly better results.

D. WR P2P HY With Peer-Delay

The first node of the cascade is a P2P master clock and the last one is a P2P slave. Intermediate nodes are set as HYs so that they forward PTP frames from their upstream to the downstream port and, in addition, compute the $\text{offset}_{\text{ms}}$ using the timing information of these forwarded PTP frames. The last node computes the $\text{offset}_{\text{ms}}$ using six timestamps.

In this scenario, all nodes are syntonized to the retrieved L1 frequency and synchronized using the received PTP frames (that are also forwarded downstream) from the master node N_{00} .

Offset accuracy results are summarized in Fig. 7 and Table I. There are no results for N_{18} and N_{19} since these nodes were not able to syntonize, as occurred with E2E and P2P BCs scenarios.

P2P HYs $\text{offset}_{\text{ms}}$ measurements present also an important improvement with respect to a BC configuration, achieving a maximum averaged $\text{offset}_{\text{ms}}$ value of 560 ± 115 ps (σ) for N_{17} . In terms of scalability, it is possible to synchronize 17 nodes, instead of the 18 ones achieved by P2P TCs. Although accuracy results are close in TCs and HYs, Fig. 8 shows similar jitter values in HYs and BCs (P2P and E2E). This behavior is produced by the phase tracking process on each HY node, which degrades the propagated clock reference the same way a BCs chain does. 736 ± 51 ps (σ) is the maximum jitter value, achieved at N_{17} . This value is slightly better than the one obtained in BCs scenarios. However, there is a significant increment in the standard deviation for the N_{18} for the TCs setup. Considering the four evaluated scenarios, we need to remark the main difference between TCs and the rest ones: In TCs, only the last node of the chain performs a full synchronization (frequency and phase), whereas, in the others, all nodes perform a full synchronization. This might explain why the given standard deviation values for

TC in Table I are slightly greater for all the evaluated nodes, specially, for the last node, presenting a considerable increment. Nevertheless, the TC implementation stands out as the best choice.

Jitter results are summarized in Fig. 8 and Table II. Comparing the four scenarios, it can be stated that jitter is not influenced by the delay estimation mechanism method used, E2E or P2P. For BCs scenarios, as well as for the HY ones, jitter results are quite similar. The observed differences could be produced by the variability between each link up. Conversely, the TC scenario presents a significant jitter reduction concerning the other evaluated scenarios. This could be potentially the reason TCs chains reach 18 nodes.

Another interesting result obtained is the jitter upper bound limit, which directly affects the recovery process of the clock system from the incoming port. In the four scenarios, slave nodes presenting 700 ps of jitter could not recover the clock from the master.

In terms of synchronization accuracy, results are very promising. It can be assumed that all clock implementations are capable of synchronizing to the master reference with an accuracy below 1 ns until hop N_{10} . However, only P2P TCs and P2P HYs are capable of guaranteeing sub-ns accuracy up to 17 and 18 slave nodes, respectively. This demonstrates that WR offers better accuracy using P2P TC/HYs clocks instead of the originally developed E2E BC implementation.

Finally, it is also worth indicating that new developments as the one presented in [30] can play a significant role on the distribution of low jitter and accurate frequencies, forming even larger chains maintaining the synchronization accuracy below 1 ns.

V. CONCLUSION

This article presented three main clear scientific contributions. First, it demonstrated that very large cascade of timing devices can be synchronized below 1 ns, improving results from previous work from 11 to 19 hops. Second, it demonstrated that the E2E delay model is not the best approach for accurate time transfer, demonstrating that a P2P delay model guarantees better accuracy. Third, the article provided a reference implementation for WR that can be extended to other protocols that use PTP + L1 syntonization, such as UIT-T G.8275.1. Furthermore, the experiments showed the impact of TCs and HYs on the synchronization, together with their pros and cons.

As a consequence, these developments have enhanced the performance of the WR protocol in terms of scalability and stability in long daisy-chain deployments compared to the default E2E BC WR implementation. Furthermore, since P2P is the most commonly used delay measurement mechanism in industrial and telecom timing networks, the utilization of P2P opens the doors to the integration of the WR protocol in these domains. For this reason, the implementation of P2P WR clocks in IRT communications in different protocols such as Profinet or TSN may help to improve the determinism and scalability of Ethernet networks for hard real-time industrial applications.

The results presented in Section IV demonstrated that the utilization of P2P TCs and P2P HYs improves significantly the scalability and accuracy of the disseminated timing signal (1-PPS) over long-cascade configurations. In this regard, HYs and TCs guarantee a sub-ns synchronization accuracy for chains composed of 17 HYs and 18 TCs, whilst E2E and P2P BC scenarios are only able to guarantee this accuracy until the tenth slave node. Furthermore, measured 1-PPS offsets for P2P HYs and TCs scenarios show a 3–5 times better accuracy than using BC configurations. Jitter results for HY deployments unveil that jitter increases in a similar way to BC implementations. This is caused by the WR phase correction process. Since P2P TCs avoid this correction process, noise is reduced by 20–30%, and, consequently, the scalability is improved.

It is clearly evidenced that the scalability of WR does not rely on the synchronization process using WR-PTP, the major contribution comes from the degradation of the transmitted frequency over L1. This degradation is caused by the clock recovery system: frequency syntonization and the phase tracking process. All the analyzed scenarios use the same circuitry and share the same parameterization of the servo system in charge of the clock recovery system.

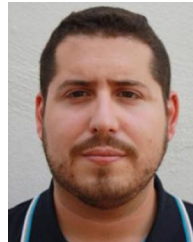
These results demonstrated that the utilization of P2P TCs and HYs instead of the default E2E BC implementation improves significantly the WR performance, being now suitable for wide area networks composed of many devices forming cascade configurations such as scientific infrastructures, telecom networks, smart grid, and 5G, increasingly demanding better timing accuracy [3], [5].

After demonstrating that the mechanism to measure the offset is not relevant for the increase of jitter in WR devices, a deeper study of the servo system together with the characterization of its parameters must be performed in order to reduce jitter and, thus, improve scalability beyond 19 nodes. The integration of low jitter devices, such as [30], is also part of this evaluation to increase the number of hops in the chain. In addition to that, new control algorithms may be evaluated in order to reduce the noise generated within the phase tracking synchronization process.

REFERENCES

- [1] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Standard 1588-2019, Jul. 2020. [Online]. Available: <https://standards.ieee.org/standard/1588-2019.html>
- [2] F. Girela-López, J. López-Jiménez, M. Jiménez-López, R. Rodríguez, E. Ros, and J. Díaz, "IEEE 1588 high accuracy default profile: Applications challenges," *IEEE Access*, vol. 8, pp. 45211–45220, 2020.
- [3] M. I. Ridwan, N. S. Miswan, M. S. M. Shokri, M. N. Noran, R. M. Lajim, and H. N. Awang, "Interoperability in smart grid using IEC 61850 standard: A power utility prospect," in *Proc. IEEE Innov. Smart Grid Technol. - Asia (ISGT ASIA)*, Kuala Lumpur, Malaysia, 2014, pp. 261–266.
- [4] J. von Hoyningen-Huene *et al.*, "Timing evaluation of cascaded industrial communication networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5497–5504, Oct. 2019.
- [5] A. Derviskadic, R. Razzaghi, Q. Walger, and M. Paolone, "The white rabbit time synchronization protocol for synchrophasor networks," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 726–738, Jan. 2020.
- [6] B. Moussa, M. Kassouf, R. Hadjidj, M. Debbabi, and C. Assi, "An extension to the precision time protocol (PTP) to enable the detection of cyber attacks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 18–27, Jan. 2020.

- [7] T. Qiu, Y. Zhang, D. Qiao, X. Zhang, M. L. Wymore, and A. K. Sangaiah, "A robust time synchronization scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3570–3580, Aug. 2018.
- [8] *IEEE Guide for Synchronization, Calibration, Testing and Installation of PMUs for Power System Protection and Control*, IEEE Standard C37.242–2013, Mar. 6, 2013.
- [9] G. Frigo, D. Colangelo, A. Derviskadic, M. Pignati, C. Narduzzi, and M. Paolone, "Definition of accurate reference synchrophasors for static and dynamic characterization of PMUs," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 9, pp. 2233–2246, Sep. 2017.
- [10] U. Annakage *et al.* *Application of Phasor Measurement Units for Monitoring Power System Dynamic Performance*, EPFL Standard 232958, 2017.
- [11] P. Danielis *et al.*, "Survey on real-time communication via ethernet in industrial automation environments," in *Proc. IEEE Emerg. Technol. Factory Autom.*, Barcelona, Spain, 2014, pp. 1–8.
- [12] R. Schlesinger, A. Springer, and T. Sauter, "Automatic packing mechanism for simplification of the scheduling in Profinet IRT," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1822–1831, Oct. 2016.
- [13] D. P. Venmani, O. Le Mout, F. Deletre, Y. Lagadec, and Y. Morlon, "On the role of network synchronization for future cellular networks: An operator's perspective," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 58–64, Sep. 2016.
- [14] S. Ruffini, S. Rodrigues, M. Lipinski, and J. Lin, "Synchronization standards toward 5G," *IEEE Commun. Standards Mag.*, vol. 1, no. 1, pp. 50–51, Mar. 2017.
- [15] S. Bhattacharjee, R. Schmidt, K. Katsalis, C. Chang, T. Bauschert, and N. Nikaein, "Time-sensitive networking for 5G fronthaul networks," in *Proc. IEEE Int. Conf. Commun.*, Dublin, Ireland, 2020, pp. 1–7.
- [16] "Industry leaders release the new CPRI specification for 5G," Jul. 2020. [Online]. Available: <http://www.cpri.info/press.html>
- [17] *ITU-T Recommendations: Network Limits for Time Synchronization in Packet Networks*, ITU-T Standard G.8271.1/Y.1366.1, 2017.
- [18] H. Li, L. Han, R. Duan, and G. M. Garner "Analysis of the synchronization requirements of 5G and corresponding solutions," *IEEE Commun. Standards Mag.*, vol. 1, no. 1, pp. 52–58, Mar. 2017.
- [19] P. Moreira, J. Serrano, T. Wlostowski, P. Loschmidt, and G. Gaderer, "White rabbit: Sub-nanosecond timing distribution over ethernet," in *Proc. Int. Symp. Precis. Clock Synchronization Meas., Control Commun.*, Brescia, Italy, 2009, pp. 1–5.
- [20] O. Seijo, J. A. López-Fernández, H. Bernhard, and I. Val, "Enhanced timestamping method for subnanosecond time synchronization in IEEE 802.11 over WLAN standard conditions," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5792–5805, Sep. 2020.
- [21] T. Wlostowski, "Precise time and frequency transfer in a white rabbit network," Diss. Instytut Radioelektroniki, 2011. Accessed: Oct. 2020. [Online]. Available: https://white-rabbit.web.cern.ch/documents/Precise_time_and_frequency_transfer_in_a_White_Rabbit_network.pdf
- [22] Z. Idrees *et al.*, "IEEE 1588 for clock synchronization in industrial IoT and related applications: A review on contributing technologies, protocols and enhancement methodologies," *IEEE Access*, vol. 8, pp. 155660–155678, 2020.
- [23] E. F. Dierix *et al.*, "White rabbit precision time protocol on long-distance fiber links," *IEEE Trans. Ultrason. Ferroelect., Freq. Control*, vol. 63, no. 7, pp. 945–952, Jul. 2016.
- [24] C. Prados, "White rabbit transparent clock," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control Commun.*, Lemgo, Germany, 2013, pp. 13–17.
- [25] J. L. Gutiérrez-Rivas, C. Prados, and J. Díaz, "Sub-nanosecond synchronization accuracy for time-sensitive applications on industrial networks," in *Proc. Eur. Freq. Time Forum*, York, U.K., 2016, pp. 1–4.
- [26] International Telecommunication Union, "G.8265: Architecture and requirements for packet-based frequency delivery," Accessed: Jul. 2020. [Online]. Available: <https://www.itu.int/rec/T-REC-G.8265/en>
- [27] International Telecommunication Union, "G.8275.1: Precision time protocol telecom profile for phase/time synchronization with full timing support from the network," Accessed: Jul. 2020. [Online]. Available: <https://www.itu.int/rec/T-REC-G.8275.1/en>
- [28] *IEEE Standard for Synchrophasor Measurements for Power Systems*, IEEE Standard C37.118.1-2011 (Revision of IEEE Standard C37.118-2005), Dec. 28, 2011.
- [29] F. Torres-González, J. Díaz, E. Marín-López, and R. Rodríguez-Gómez, "Scalability analysis of the white-rabbit technology for cascade-chain networks," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control, Commun.*, Stockholm, Sweden, 2016, pp. 1–6.
- [30] M. Rizzi, M. Lipinski, P. Ferrari, S. Rinaldi, and A. Flammini, "White rabbit clock synchronization: Ultimate limits on close-in phase noise and short-term stability due to FPGA implementation," *IEEE Trans. Ultrason., Ferroelect. Freq. Control*, vol. 65, no. 9, pp. 1726–1737, Sep. 2018.



José Luis Gutiérrez-Rivas received the M.Sc. degree in computer science and the Ph.D. degree in information and communication technologies from the University of Granada, Granada, Spain, in 2009 and 2018, respectively. His main research interests include safety-critical systems, time and frequency distribution and network dependability, fault tolerance, and reliability.



Felipe Torres-González received the M.Sc. degree in data science and computer engineering from the University of Granada, Granada, Spain, in 2017.

He is currently a FPGA Engineer with Integrated Control Division, European Spallation Source, Lund, Sweden. His main research interests include embedded systems based in field-programmable gate arrays, time distribution based on White Rabbit or micro-research Finland (MRF), and data acquisition for particle accelerators.



Eduardo Ros received the Ph.D. degree from the University of Granada, Granada, Spain, in 1997.

He is currently a Full Professor with the Department of Computer Architecture and Technology, University of Granada, Granada, Spain. He is also the Co-Founder and intensively involved in Seven Solutions enterprise focused on ultra-accurate timing systems. He is a very active Researcher at an international level, and he has participated as IP in eight European Grants (of the 5th, 6th, 7th, and H2020 EU frameworks). He leads an interdisciplinary lab, with interest in computational neuroscience, neuromorphic engineering, real-time processing, etc. He is also actively researching ultra-accurate timing systems in different industrial applications. He has authored or coauthored more than 85 papers on international journals. He has three patents under exploitation. His main research interests include ultra-accurate time transfer and synchronization of distributed instrumentation systems, simulation of spiking neural networks, high performance computer processing, hardware implementation of digital circuits for real-time processing in embedded systems, etc.

Dr. Ros was the recipient of the Andalucía Award for Best Young Researchers in 2002.



Javier Díaz received the M.S. degree in electronics engineering and the Ph.D. degree in electronics from the University of Granada, Granada, Spain, in 2002 and 2006, respectively.

He is currently a University Professor and Collaborates with research facilities such as CERN, IFMIF-EVEDA, CTA, or SKA working on sub-nanosecond time transfer solutions based on White Rabbit technology. He has authored or coauthored more than 40 peer reviewed journal papers and supervised more than eight Ph.D.

thesis. His main research interests include high performance image processing architectures, safety-critical systems, highly accurate time synchronization, and frequency distribution techniques.