IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Precise Network Time Monitoring: Picosecond-level packet timestamping for Fintech networks

Francisco Girela-López [1], Eduardo Ros [1], Javier Díaz [1]
[1]University of Granada, Granada, Spain

Corresponding author: Francisco Girela-López (e-mail: girlop@ugr.es).

ABSTRACT Network visibility and monitoring are critical in modern networks due to the increased density, additional complexity, higher bandwidth, and lower latency requirements. Precise packet timestamping and synchronization are essential to temporally correlate captured information in different datacenter locations. This is key for visibility, event ordering and latency measurements in segments as telecom, power grids and electronic trading in finance, where order execution and reduced latency are critical for successful business outcomes.

This contribution presents Precise Network Time Monitoring (PNTM), a novel mechanism for asynchronous Ethernet packet timestamping which adapts a Digital Dual Mixer Time Difference (DDMTD) implemented in an FPGA. Picosecond-precision packet timestamping is outlined for 1 Gigabit Ethernet. Furthermore, this approach is combined with the White Rabbit (WR) synchronization protocol, used as reference for the IEEE 1588-2019 High Accuracy Profile to provide unprecedented packet capturing correlation accuracy in distributed network scenarios thanks to its sub-nanosecond time transfer. The paper presents different application examples, describes the method of implementation, integration of WR with PNTM and subsequently describes experiments to demonstrate that PNTM is a suitable picosecond-level distributed packet timestamping solution.

INDEX TERMS Packet timestamping, visibility networks, monitoring, White Rabbit, IEEE 1588, FinTech

I. INTRODUCTION

Smart cities and grids, Distributed Ledger Technologies (DLT), telecom, wireless networks, electronic trading, vehicular communications or Internet of Things (IoT) are examples of the rise in distributed synchronized networks with demanding requirements in terms of bandwidth, latency or monitoring [1], [2], [3], [4], [5], [6].

In those segments, visibility networks are critical for monitoring the performance of time sensitive applications specifically in distributed scenarios such as an integrated facility. For instance, in the telecom segment, low latency and time synchronization requirements associated to Fifth Generation (5G) are highly demanding and can be even more exigent for future Sixth Generation (6G) [7], [8]. For that reason, data capture is used to detect poor performance and guarantee proper operation [9].

Due to the distributed architecture of communication networks, a selected number of remote measurement points are chosen for monitoring. The nodes in those points must guarantee proper measurement, so they typically offer a performance figure one order of magnitude better than the services being monitored. Therefore, the capabilities of the visibility network are strongly related to the timestamping precision and the synchronization accuracy, imposing highly demanding specifications from many industrial applications and complex visibility networks architectures.

Nowadays, large scale visibility networks use flow-based visibility techniques which consist of software tools integrated in network devices, e.g., routers or switches, that are used to gain insights into the data stream. This approach requires per device support and its monitoring scope is limited to the traffic being monitored on the device. Due to its nature, this approach can affect the performance, adding latency, reducing the throughput, or increasing the computation needs [10]; but relevant information is accessible without requiring extra devices. Alternatively, hardware capturing capabilities or tapping units, that transparently sniff the data stream are commercially available [11], [12], [13], [14], [15]. These devices append the timestamping information to the packets before processing them or export the information to a centralized monitoring platform with advanced functionalities reaching nanosecond or even sub-100 picoseconds precision.

This last option is the norm in finance sector networks due to the criticality of latency. In the electronic trading market, order executions are driven by price and order of arrival, so the reaction time of the participants to market data is crucial to sell or buy equities, shares, or other financial products with better margins. Since the beginning of electronic trading, a "low-latency" race between market participants, especially High Frequency Traders (HFT), has promoted a continuous upgrade of the technology in use to improve this reaction time. In order to reduce the physical propagation time, traders co-locate their servers and network devices in the datacenters where stock exchanges or financial institutions trading engines are located. Additionally, trading firms link the datacenters in financial hubs, like London or New York, using microwave links or hollow core fibers and invest in Ultra Low Latency (ULL) equipment. Consequently, Field-Programmable Gate Array (FPGA) based devices have been developed for networking purposes and High-Performance Computing (HPC) [16], [17]. This fact, combined with other solutions based on layer 1 (L1) switching or Application-Specific Integrated Circuits (ASICs), have led the low latency competition to the single-digit nanosecond level [11], [13], [14], [15]. Logically, non-intrusive packet latency measurement needs in the order of few picoseconds in distributed scenarios is a trend in this segment. That information is additionally used to improve the network performance, to train the Artificial Intelligence (AI) trading algorithms and is used to adhere to various regulations. This forces these trading participants to perform timestamping across distributed trading networks in order to capture packets relating to trading activities [18]. However, the timestamping methods must avoid introducing latency while requiring high accuracy synchronization for event correlation and network-based management, data visualization and security tools. Although there are available commercial timestamping solutions in the sub-nanosecond level, the competence level and the adoption of higher Ethernet bandwidths, thus faster bit transmission, advise that better distributed timestamping precision in non-intrusive visibility networks will be required in the near future.

Another key application where this timestamp precision is required is the metrology field. National Metrology Institutes (NMIs) are continually researching mechanisms for optical pattern comparison. Optical fiber time distribution is one of the key technologies and this has motivated the existence of projects such as CLONETS or CLONETS-DS [19] [20]. In this field, the target performance requires sub-picosecond accuracy but the sub-10 ps approach can be used to deploy general purpose and cost-effective solutions for distributing time references from NMIs without dedicated optical fiber links and highly expensive metrology equipment.

The motivation of this paper is to create a novel, cost-effective FPGA-based packet timestamping mechanism with picosecond-level precision for Ethernet links that can fulfill the requirements from above segments without imposing operational constraints on the production networks. Nevertheless, this work is particularly focused on the FinTech domain. For that purpose, it must be able to maintain the precision when timestamping packets originated by asynchronous and non-syntonized network devices. It needs to be easy to integrate with low hardware and computing resource usage and simple calibration methods. The proposed approach must work well in distributed synchronized networks without affecting the link latency. For that purpose, the contribution in this manuscript focuses on performing phase measurements using a modified Digital Dual Mixer Time Difference (DDMTD) in order to achieve single-digit picosecond timestamping precision of incoming packets generated by asynchronous devices. Later, the timestamper is integrated in White Rabbit (WR) devices to validate its applicability in distributed scenarios with high accuracy time synchronization protocols.

II. CURRENT SYNCHRONIZATION AND TIMESTAMPING METHODS

Correlation and event ordering in distributed asynchronous networks with very accurate temporal resolution is recurrent in modern telecommunications. While single-point correlation only depends on the device timestamping resolution, distributed capturing is a challenge where accurate timing and precise timestamping need to be integrated in remote heterogeneous deployments along different network locations.

A. CLOCK SYNCHRONIZATION

Clock synchronization is considered the ability to transfer a shared notion of time between different devices that can be placed in remote locations. To disseminate the time information several time dissemination mechanisms are available, including analog signals i.e., Pulse Per Second (PPS) synchronization, metrological methods [21], [22] or satellite-based systems [23], [24]. For the sake of clarity and due to its widespread use in the FinTech segment, this article will focus on network synchronization protocols. These protocols use a periodic packet exchange between devices where the packets are timestamped at transmission and reception so the time offset between the clocks of the devices can be calculated and removed.

The timing accuracy of these network synchronization protocols is affected by different factors:

- Latency uncertainty: The variability of the value between different timestamps, i.e., latency between the moment when the packet arrives to the interface and when it is timestamped. This is relevant in software timestamping as the latency of the timestamp can vary in the range of microseconds depending on the system load, the kernel or the number of operating system interruptions.
- Timestamp resolution: The uncertainty related to the capture period of the timestamp. For example,

if digital clock edges are used to trigger the timestamp, the resolution is equal to the period between these digital clock edges.

- Time base resolution: Analogously to the timestamp resolution, this is the time base update period that keeps track of time. In this case, if the time base is updated using an internal counter increased each clock cycle, the uncertainty will be equal to the clock period.
- Clock drift between corrections: This is produced by the frequency difference between the local oscillators in different devices during the elapsed time between corrections. This affects all the previous factors.
- Clock phase offset: An additional inaccuracy caused by the phase relationship between the clocks from different devices. This is caused by the propagation delay which is affected by environmental conditions.

From the previous factors, the latency and the resolution conform the timestamping indeterminism. The impact of this indeterminism can be minimized using hardware-based implementations that ensure better static latencies (not depending on software layers) and high-resolution timestamping mechanisms. After solving the inaccuracies caused by the indeterminism, the clock drift between corrections can be removed using syntonization. This is a mechanism that distributes the frequency from a reference device to the rest of the network. This reference can be used as the internal clock reference for the synchronized devices instead of the local oscillators, removing the temporal drift. A remarkable example of syntonization mechanism is Synchronous Ethernet (SyncE) that recovers the reference frequency from the incoming data stream. Lately, clock phase misalignments can be corrected performing phase measurements and clock phase tuning in the devices.

Reference standards in FinTech are Network Time Protocol (NTP) and IEEE 1588 Precise Time Protocol (PTP). NTP is mostly used in servers and on virtual machines because of its software implementation and its ability to achieve microsecond level accuracy. Meanwhile, PTP can reach tens of nanoseconds accuracy using hardware timestamping in controlled scenarios [25], [26] and is the typical standard for switches and Network Interface Cards (NIC). Some PTP profiles support syntonization in their definition. This is the case with WR which achieves sub-nanosecond accuracy, as a pre-standard implementation of the High Accuracy (HA) profile of the last IEEE 1588-2019 standard. Hereto, WR generates a syntonized internal copy of the recovered clock and exchanges time packets which are timestamped to determine the time offset from the reference. The resolution of the timestamps is improved based on a double picosecond-level phase measurement for the received packets in the reference and synchronized devices thanks to the DDMTD module. After the time offset calculation, the local oscillator is tuned to shift the local clock phase until it matches the reference by forcing faster or slower frequencies for a specific time period, displacing the phase [27], [28]. Thanks to its sub-nanosecond synchronization accuracy, WR is used to replace analog distribution and provide last hop synchronization to FPGA-based devices as L1 switches and NIC cards and will be used as the reference technique for integrating with the PNTM method.

B. TIMESTAMPING

Timestamping is important as part of the time synchronization protocols and for the asynchronous packet timestamping generated in the visibility devices by non-synchronized equipment. This is the case of FinTech where the production devices use Ethernet networks but do not need to be synchronized nor syntonized to a common time or frequency source. In order to properly correlate the captured data between different devices, a proper resolution in the timestamping is mandatory. Although there are specific ASIC solutions for timestamping, most of the techniques can be implemented on generic FPGAs, so these devices will be taken as reference for the following review of the existing techniques. The timestamping solutions offer different performance and cost characteristics:

- Digital clock: They can be used both in software and hardware implementations, but the precision is constrained by the period of the maximum frequency supported by the electronics [29].
- Input/Output (I/O): In FPGAs, these I/O resources can be used to work at higher frequencies than the general FPGA logic. Serializer/de-serializer (SERDES) modules can achieve single nanosecond resolution thanks to their optimized design [30].
- Carry-chain or DSP delay lines: Logic propagation chains can be used to perform timestamping with customized Time-to-Digital Converter (TDC) chips, carry-chains in the FPGA or interpolators [31], [32]. Although they can achieve up to picosecond-level resolutions on FPGA devices [33], they require complex calibrations, are affected by temperature changes, and require dedicated hardware or extensive logic resources.
- Multi-clock: These designs generate several copies of the local clock with known phase delays to reach picosecond-level resolution [34]. They are typically combined with previously exposed methods.
- Phase comparison: These mechanisms rely on recovering the event generator clock and comparing it against the internal clock. Although this mechanism can achieve picosecond-level resolution [35], clock drifts can affect the resolution when asynchronous and non-periodic events need to be timestamped.

It is noteworthy that the technical implementations in FinTech are protected by trade secrecy, but typically share an FPGA-based design and precision levels above the tens of picoseconds i.e., nanoseconds or sub-nanosecond domain, [11], [13], [14], [15], [29]. Our assumption due to these characteristics is that high-frequency clocks and multi-clock solutions are predominant in this segment.

Therefore, the novel approach presented in this paper relies on phase comparisons to timestamp incoming packets in Ethernet links, using a Clock Data Recovery (CDR) module to retrieve the transmission clock encoded into the data stream and compare it with an internal reference. The implemented DDMTD phase comparison module is adapted to overcome its limitations when working with asynchronous sources to reach single-digit picosecond precision, outperforming the resolution of previous FPGA techniques, and equalizing the most accurate non-FPGA based techniques found in the literature [33], [35].

The goal is to avoid using complex designs or external chips that complicate the integrability, require calibration or add latency so the proposed timestamping can be integrated in generic FPGAs while maintaining picosecond-level resolution. Furthermore, the authors have enabled the different timestamping devices to be synchronized using the WR protocol, obtaining a coherent internal clock reference in all the timestampers that allows high precision distributed timestamping in different points of the network. In summary, the solution will be fully compatible with the existing FinTech networks based on Ethernet technologies, allowing tapping techniques in the visibility devices to take full advantage of WR synchronization protocol and highly accurate timestamping capabilities.

III. A NOVEL TIMESTAMPING MECHANISM FOR ASYNCHRONOUS ETHERNET LINKS

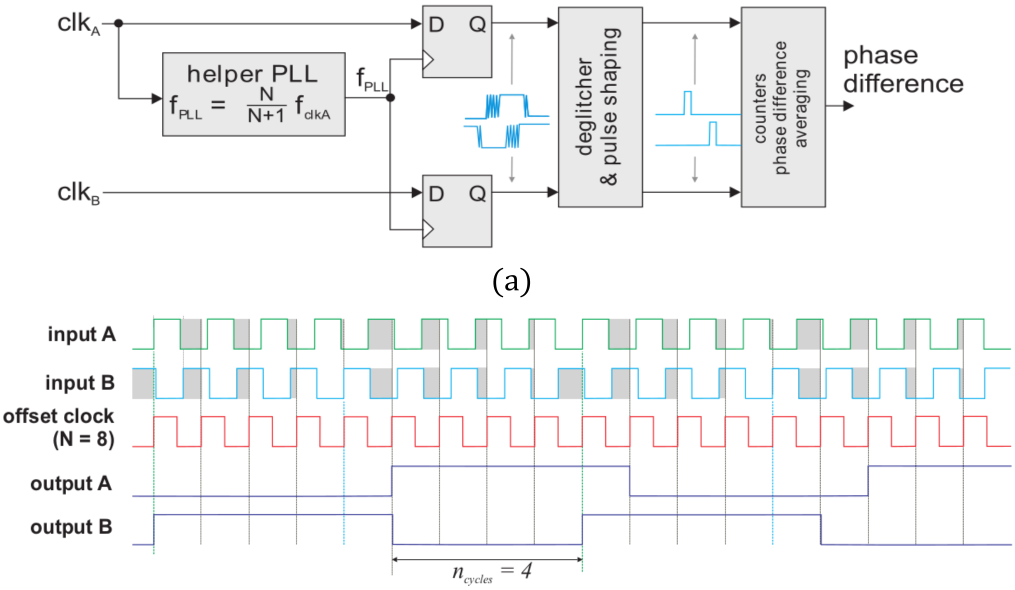
The hypothesis of the proposed distributed timestamping mechanism is that Ethernet devices can use their synchronized local time and frequency references with improved resolution based on phase measurements to correlate the reception of asynchronous packets sent by non-synchronized sources. An adapted DDMTD module and digital counters in general purpose FPGA logic are integrated for this purpose without impacting the data stream latency. The proposed implementation focuses on 1 Gigabit (Gbps) Ethernet connections that define a reference frequency equal to 125 MHz and a frequency deviation limited to ±100 parts per million (ppm). However, the timestamping module principles are applicable to any other link speeds.

A. THEORETICAL BACKGROUND

The first two steps in performing the timestamping are the data sniffing to detect the Start of Frame (SoF) signal of each packet without adding latency; and the recovery of the transmission clock by the CDR mechanism in the transceiver. The SoF signal is used as the trigger for the timestamp and the CDR will be the key to find the phase relationship between clocks.

The design has two counters which keep track of the seconds and the number of elapsed clock cycles from the last second update. For example, the digital clock period in this design is 16 ns, equivalent to a 62.5 MHz clock, which defines the resolution and determinism of the counters.

As an addition to the counters, a DDMTD has been used to perform the sub-cycle phase measurements based on the one implemented on the WR technology. An introduction to this mechanism is presented below, but additional details



(a)

(b)

Figure 1 (a) Block diagram of the DDMTD. (b) Clock signals received and generated by the DDMTD. Source: [37]. As both intpus (clk_A and clk_B) are syntonized, the phase offset between the outputs (clk_{ASlow} and clk_{BSlow}) remains constant and can be sampled because of its lower frequency.

regarding the DDMTD theory and its implementation can be found in [36] and [37]. In this module, a reference clock (clk$_A$) is sampled by a slower frequency copy (clk$_{PLL}$) generated by a Phase Locked Loop (PLL). clk$_{PLL}$ has a frequency difference determined by a fixed value N as showed in (1), which is defined to be 16384 in the default design. clk$_{PLL}$ is used as a common clock to sample clk$_A$ and the measured clock (clk$_B$).

$$f_{ClkPLL} = \frac{N}{N+1} \times f_{ClkA}\ (Hz) \qquad (1)$$

As demonstrated in Fig. 1, after sampling clk$_A$ and clk$_B$ with clk$_{PLL}$, two low frequency clocks (clk$_{ASlow}$ and clk$_{BSlow}$), whose periods are defined by the clk$_A$ and clk$_B$ phase alignments (0º phase relationship) with clk$_{PLL}$, are generated. In its default version, the DDMTD assumes that clk$_A$ and clk$_B$ are syntonized, generating two phase-coherent low frequency clocks with a theoretical period amplified by the N number [37]. The zoomed phase difference is measured as the difference (n$_{cycles}$) between two internal counters with the number of clk$_{PLL}$ cycles since clk$_A$ and clk$_B$ is directly available. Both counters generate an interruption each period, providing the actual number of clk$_{PLL}$ cycles in clk$_{ASlow}$ and clk$_{BSlow}$ (N$_A$ and N$_B$). Assuming a known frequency for clk$_A$, the sub-cycle phase difference can be calculated in general FPGA logic following (2).

$$\Phi = \frac{n_{cycles}}{N+1} \times \frac{1}{f_{ClkA}}\ (s) \qquad (2)$$

Due to the 16 ns clock period in this design, the N times zoom means a theoretical resolution lower than 1 ps per measured clock. This result on a total resolution better than 2 ps in the DDMTD comparison (1 ps for clk$_A$ and another for clk$_B$) but the resolution can be increased by just increasing the N value. The measurement is performed each clk$_{BSlow}$ cycle, so the periodicity of the calculation (DDMTD integration time) is defined by multiplying the clk$_A$ period (16 ns) and N (16384) and equal to 262.144 µs.

B. PRECISE NETWORK TIME MONITORING

Unfortunately, clk$_A$ and clk$_B$ syntonization is mandatory for a stable sub-cycle phase difference calculation using the DDMTD technique. When both clocks are not syntonized, a drift occurs during the elapsed time from the SoF assertion to the next DDMTD phase measurement resulting on erroneous values as can be seen in Fig. 2.

The PNTM hypothesis is that the frequency drift between clk$_A$ and clk$_B$ can be measured and anticipated using a modified (non- syntonized) DDMTD without major changes to its block design, especially in Ethernet links where the nominal frequency is well defined [38]. In contrast to the default DDMTD design, the clk$_B$ recovered from the data stream is sent by a device with the same nominal frequency but a slightly different actual frequency, leading to different clk$_{ASlow}$ and clk$_{BSlow}$ periods. As shown in (3), the frequency difference is measured thanks to the sampling and the internal DDMTD counter that keeps N_B.

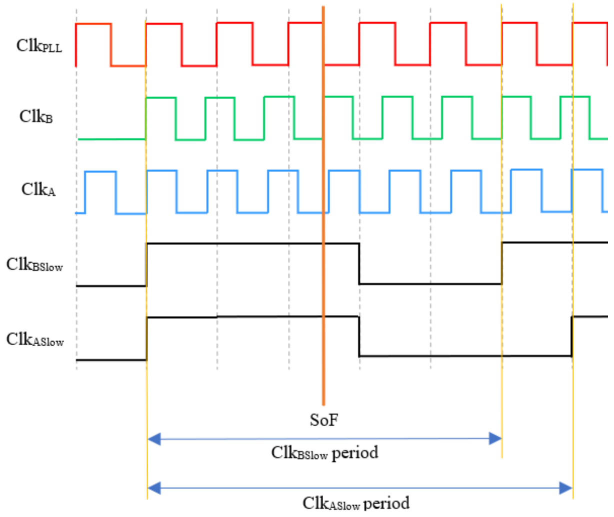


Figure 2 Modified DDMTD clock signals under no syntonization circumstances with N_A = 6 and N_B = 5. The frequency difference between clk$_A$ and clk$_B$ leads to a phase drift between the SoF instant and the next DDMTD measurement resulting in an erroneous phase measurement (shown by the different clk$_{ASlow}$ and clk$_{BSlow}$ periods) that needs to be anticipated and compensated.

$$f_{ClkB} = \frac{N_B+1}{N_B} \times f_{ClkPLL}\ (Hz) \qquad (3)$$

N_B values are no longer constants and depend on clk$_{BSlow}$ period and the stability of clk$_B$ and clk$_{PLL}$. Theorically, N_A should be fixed, but, due to the jitter of clk$_A$ and clk$_{PLL}$, its value can slightly change each clk$_{ASlow}$ cycle. These values are the reference to calculate the actual frequencies in (1) and (3). To minimize the jitter influence, the counted number of clk$_{PLL}$ cycles in N_A and N_B can be averaged to obtain mean values after some slow clock periods i.e., 100. Otherwise, the instantaneous value after each slow clock cycle is used (measured from one single period of the slow clocks).

Although the drift evolution on non-disciplined oscillators can follow different patterns and can be affected by vibrations or temperature changes, this method requires assuming that the frequency difference between non-syntonized oscillators is constant during the 262.144 µs time window of the DDMTD integration time. Consequently, the non-syntonized DDMTD counter values at the SoF instant can be linearly interpolated with the values at the DDMTD integration time. This assumption is evaluated and validated in Section IV.

In the 1 Gbps case, the frequency deviation is limited to ±100 ppm resulting on a worst-case drift between both clocks of 25 kHz. This means that the phase drift between both clocks during the integration time can exceed the 16 ns cycle period. However, the DDMTD provides data about the reference 0º phase relationship between clk$_{PLL}$ and clk$_A$ at the measurement instant and the clk$_A$ and clk$_B$ frequency difference, so full cycle drifts can be removed.

In the PNTM implementation, the SoF pulse is used to capture the most significative time information from the seconds and clock cycles counters and the current DDMTD counters values. Later, it waits until the next DDMTD phase measurement to capture the counters at the reference 0º phase relationship between clk_{PLL} and clk_A. This way, two additional values are obtained in comparison to the syntonized DDMTD system:

- Tag_A: clk_{ASlow} rising edge counter value.
- Tag_B: clk_{BSlow} rising edge counter value.
- Tag_{ASoF}: clk_{ASlow} SoF counter value.
- Tag_{BSoF}: clk_{BSlow} SoF counter value.

The resulting phase difference caused by the frequency drift between clk_A and clk_{PLL} is linearly interpolated. For that purpose, Tag_{ASoF} is subtracted from Tag_A, and the result multiplied by the period difference per cycle between clk_{PLL} and clk_A, as seen in (4).

$$\Delta\Phi_{Main} = (Tag_A - Tag_{ASoF}) \times (T_{ClkPLL} - T_{ClkA})\ (s) \quad (4)$$

Likewise, the phase difference due to the frequency drift between clk_B (3) and clk_{PLL} is calculated subtracting Tag_{TsSoF} from Tag_{Ts}. Later, the result is multiplied by the period difference per cycle between clk_{PLL} and clk_B following (5).

$$\Delta\Phi_{Ts} = (Tag_B - Tag_{BSoF}) \times (T_{ClkPLL} - T_{ClkB})\ (s) \quad (5)$$

After both values are calculated, they are added to remove the drift during the time window between the SoF and the DDMTD measurement, obtaining the absolute phase difference between clk_A and clk_B at the SoF instant. This calculation provides the sub-cycle time information with picosecond-level resolution, equivalent to the fully syntonized DDMTD.

To ensure the timestamp determinism, the reception delay ($\varDelta_{RX}$) present on the timestamper must be removed to get the actual time when the packet arrived from the fiber. $\varDelta_{RX}$ is composed by semi-static delays, i.e., transceiver latency (δ_{PHY}); fixed delays, i.e., Small-Form Pluggable (SFP) and circuit delays (δ_{SFP} and δ_{CIR}); and the clock drift associated to the delay.

Finally, the sub-cycle information and the reception delays are added to the captured values from the seconds and cycles counters in (6).

$$Timestamp = Counter_S + (Counter_{Cycles} \times T_{ClkA}) + \Delta\Phi_{MAIN} - \Delta\Phi_{TS} - \varDelta_{RX}\ (s) \quad (6)$$

High accuracy synchronization can be exploited to obtain a shared notion of time that aligns the time counters and derivate a reference clk_A for the DDMTD. This is used to perform relative timestamping comparisons, allowing monitoring differences across PNTM timestampers in distributed visibility networks. WR has been chosen as the synchronization protocol because of its sub-nanosecond accuracy and its frequency distribution jitter in the picosecond level. Alternatively, other synchronization mechanisms can also be used, but their accuracy will impact

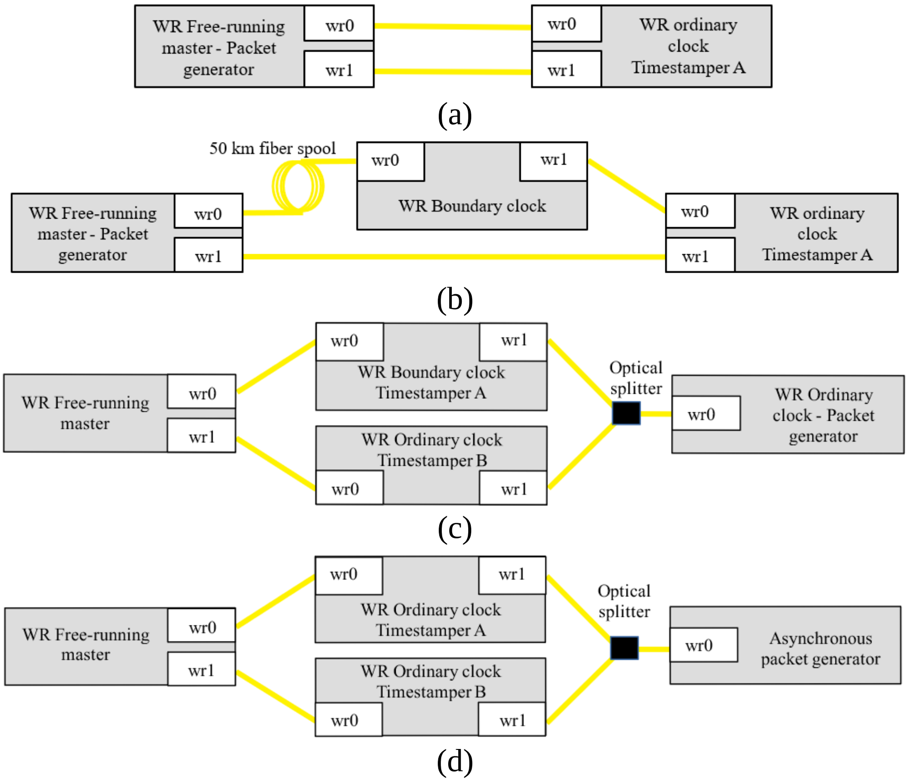

WR Free-running master - Packet generator
wr0
wr1
wr0
wr1
WR ordinary clock Timestamper A
(a)
50 km fiber spool
wr0
wr1
WR Boundary clock
WR Free-running master - Packet generator
wr0
wr1
wr0
wr1
WR ordinary clock Timestamper A
(b)
WR Free-running master
wr0
wr1
wr0
wr1
WR Boundary clock Timestamper A
WR Ordinary clock Timestamper B
wr0
wr1
Optical splitter
wr0
WR Ordinary clock - Packet generator
(c)
WR Free-running master
wr0
wr1
wr0
wr1
WR Ordinary clock Timestamper A
WR Ordinary clock Timestamper B
wr0
wr1
Optical splitter
wr0
Asynchronous packet generator
(d)

Figure 3 **(a) Point-to-point short distance data link between synchronized nodes. (b) Point-to-point short distance data link between syntonized but not synchronized nodes connected via a long-distance link. (c) Point-to-point parallel short distance data links to a synchronized node. (d) Point-to-point parallel short distance data links to an asynchronous node.**

the corresponding timestamping precision. Due to the minimal modification of the default hardware from the DDMTD design, the resource consumption is comparable to other implementations [39], [40] which represents another valuable benefit compared to other methods.

IV. EXPERIMENTS AND RESULTS

For the experimental setups, a variety of WR-compliant devices have been used as auxiliary devices for synchronization or packet generation purposes, including the DOWR, the WR-LEN and the WR-switch [41]. The PNTM timestamper has been developed in the WR-ZEN TP [42] which is a dual-port node (wr0 and wr1) that integrates a Xilinx Zynq-family FPGA. The timestamping implementation is integrated in the optical interface wr1 and only requires general FPGA logic.

There are four different roles in terms of synchronization:

- WR Free-running master: It is the device that uses its internal oscillator as the time reference for the WR-network.
- WR Boundary clock: A device which is synchronized to a master device using the wr0 optical interface and that distributes this time information to another device using its wr1 optical interface.
- WR ordinary clock: The node which is synchronized to a free-running or a boundary clock device and does not distribute this reference to any other node.
- Asynchronous: A device which is not synchronized to the WR network but generates packets to timestamp.

The different setups are indicated in Fig. 3. The timestampers are always synchronized to a reference through the optical interface wr0. The wr1 interface is always connected to the synchronous/asynchronous packet generator that generates the timestamped packets. Our experiments will carefully demonstrate that the presented implementation is performing well with repetitive and accurate results, first with synchronous traffic and lately with fully asynchronous traffic scenarios.

A. THEORETICAL BACKGROUND

The chosen measurement methodology for the validation of the proposed technique is the relative comparison of timestamps extracted from the FPGA using the Vivado Chipscope between synchronized devices. The motivation for this measurement approach is its similarity to the actual results that can be obtained in real-life distributed visibility networks. This comparison provides the combined standard uncertainty (σ_{com}). In this case, the synchronization accuracy, the measurement precision and the timestamping precision affect the outcome, so it is necessary to discern the noise contribution of each one. The measurement noise variance (σ_{meas}) is not included on our evaluation thus we are over-estimating the jitter from the other sources (synchronization and timestamp). These variables have approximately Gaussian jitter distributions, but they are strongly correlated to each other because they use the same clock circuitry and very similar FGPA modules (PNTM is based on the DDMTD blocks). This means that to estimate the timestamp precision (σ_{ts}), the synchronization (σ_{sync}) influence is removed from σ_{com}. Due to the strong correlation between both variables, a purely linear noise aggregation as shown in (7) should be used instead of a root-sum-of-squares (RSS) as typically used for independent noise variables.

$$\sigma_{com} = \left(\sigma_{sync} + \sigma_{ts} + \sigma_{meas}\right), \sigma_{meas} \approx 0\ (s) \qquad (7)$$

For the sake of clarity, the integration is performed in known devices [43] using the PPS outputs to measure a reference σ_{sync} of 14.314 ps in a single-hop link. These devices use specific clocking circuitry to clean the PPS outputs so the actual internal σ_{sync} value inside the FPGA (where the σ_{com} values is extracted) is unknown, although is expected to be slightly higher than the measured results. This measurement is also used to remove the synchronization offset, but other sources of indeterminism are foreseeable, i.e., FPGA transceivers and SFP latency, counter precision, and circuitry delays that could be corrected by providing absolute calibration [17]. A Keysight 53210A [44] with 10 ps single-shot precision is used for the measurement.

Finally, approximations using the nominal clk_A frequency instead of the experimental one and ignoring the fixed reception delay influence are applied. This impacts the accuracy of the absolute timestamping in each device, but minimally affects the relative comparisons as the reference is shared between them and the fixed reception delays are equal. In real scenarios, high precision time sources are used, removing the clk_A frequency error, and the fixed delays can be calibrated. Consistently to the platform and SFPs, sub-nanosecond accuracy is expected.

All experiments have been performed three times to ensure their repeatability and reflect a minimum capture period of 5 minutes, depending on the packet generator data rate. The number of samples are shown in the graphs for each experiment.

B. DETERMINISM AND PRECISION TEST

In Fig. 3.a, a setup with two point-to-point links between a WR free-running master and a timestamper working as an ordinary clock; one for synchronization and other for packet timestamping, is shown. The goal is to evaluate the PNTM precision and determinism. After the synchronization, a packet exchange is timestamped at the transmission (T1) by the WR free-running master and at the reception in the synchronized PNTM timestamper (T2). Because both devices are synchronized with WR, the traffic is generated in a synchronous way so the WR mechanism could be also used on this scenario. The goal is to verify that the PNTM implementation works similarly than the original methods to validate that it does not introduce additional errors.

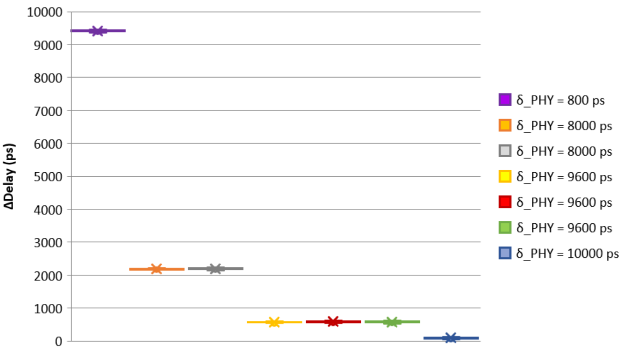


Figure 4 **Normalized $\Delta_{T2\text{-}T1}$ results show a deterministic increasement in latency equal to the current δ_{PHY} values in each execution.**

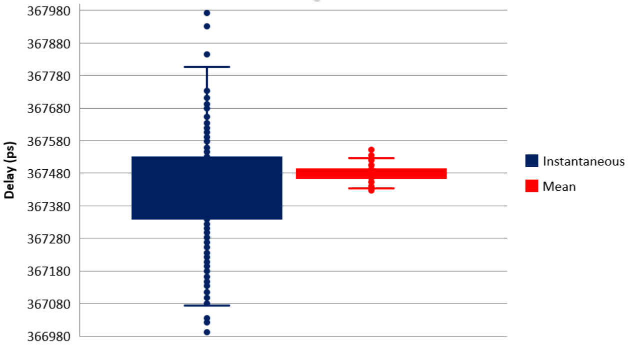


Figure 5 **$\Delta_{T2\text{-}T1}$ results with instantaneous and mean N_A and N_B show better precision when using mean values (δ_{PHY} = 9600 ps).**

δPHY (ps)	10000	9600	9600	9600	8000	8000	800
ΔT2-T1 (ps)	10122	9633	9639	9637	8020	8026	800
Std. Dev. (ps)	17	16	16	15	16	14	15

Table I **Semi-static delay influence in T2-T1 results show sub-nanosecond determinism and single-picosecond timestamping precision.**

The T1 timestamping precision is expected to be equivalent to the clock jitter as it is aligned with the transmission clock edge. The fixed offset between both timestamps ($\Delta_{T2\text{-}T1}$) is equal to the propagation delay plus the synchronization offset. For the sake of clarity, this value is normalized subtracting the minimum value for the propagation delay and the synchronization offset. The measurement stability shown by the σ_{com} is the result of the synchronization and the timestamping precisions.

As shown in Fig. 4 several executions with different semi-static reception delays are performed, i.e., δ_{PHY}. Different δ_{PHY} are forced with link reconnections as they change the bitslide delay value in the Xilinx Zynq FPGA transceiver buffer in the 0 ns to 16 ns range with theoretical 800 ps steps (equivalent to the 1.25 GHz period of the transmission clock in 1 Gbps Ethernet links). This value is extracted from Chipscope and is directly reflected after normalizing $\Delta_{T2\text{-}T1}$. Additionally, measurements with equal δ_{PHY} are shown to illustrate the determinism of $\Delta_{T2\text{-}T1}$. The repeatability, the determinism, and the influence of the semi-static delay in PNTM is characterized.

In Table I, the relationship between the theoretical δ_{PHY} and the normalized $\Delta_{T2\text{-}T1}$ is studied, showing an absolute difference with the theoretical value bounded below ±150 ps. This suggests a static error on the theoretical bitslide value provided by the FPGA transceiver. This error could be characterized and compensated with a look-up table if it is deterministic.

As explained in section III, the DDMTD has two counters that obtain the N_A and N_B values equivalent to the clk_{ASlow} and clk_{BSlow} periods used to calculate the frequency difference between clk_A and clk_B,as shown in (1) and (3). In Fig. 5, a zoom of $\Delta_{T2\text{-}T1}$ shows the precision using the instantaneous values obtained each clk_{ASlow} and clk_{BSlow} periods and the mean N_A and N_B values obtained after 100 clk_{ASlow} and clk_{BSlow} periods. The plots are centered 46 ps apart and the σ_{com} is 145 ps for the instantaneous values whereas σ_{com} is 16 ps using the mean N_A and N_B. The N_A and N_B values are generated thousands of times per second, so the integration time to calculate the mean value is minimal. After averaging the results in Table I, the σ_{ts} contribution of PNTM to σ_{com} is only 1.5 ps using a reference σ_{sync} of 14 ps and a negligible measurement error. This means single-digit picosecond σ_{ts} precision even assuming the worst-case scenario. In consequence, these experiments demonstrate that our modified DDMTD method does not introduce additional errors and it can be used at least with synchronously generated traffic.

C. DYNAMIC RESPONSE AND SYNCHRONIZATION INFLUENCE TEST

At Fig. 3.b, the second experiment outline illustrates how an additional 50 km hop is used in the synchronization path. After the whole chain is synchronized, the synchronization exchange in the long-distance link is stopped, keeping only the syntonization. This leads to a slow drift in the offset between the free-running master that generates the data traffic (T1) and the PNTM timestamper that receives it (T2) but keeping the clk_A frequency differences between devices negligible for the DDMTD.

In order to measure the absolute influence in $\Delta_{T2\text{-}T1}$, a drift in the synchronization offset between the timestamper and the packet generator, which is measured with the counter, by abruptly changing the temperature. This procedure lasted several hours. After reaching a distinguishable synchronization offset, the timestamping capture is launched, recording the $\Delta_{T2\text{-}T1}$ value for a time window larger than 5 minutes.

As observed in Fig. 6, $\Delta_{T2\text{-}T1}$ increases following the increment of the synchronization offset between the packet generator and the timestamper. This proves that the timestamper dynamically reacts to sub-cycle changes in the phase and its consistent behavior in a long-duration test.

At Fig. 7, a detailed view of the -9.3 ns offset dataset is shown. Coherently to the previous experiment, the N_A and N_B mean values after 100 clk_{ASlow} and clk_{BSlow} periods for this dataset offer better stability than the instantaneous ones. The obtained σ_{com} is 257 ps in the instantaneous case and 30 ps in the mean case, showing that the mean values help to mitigate

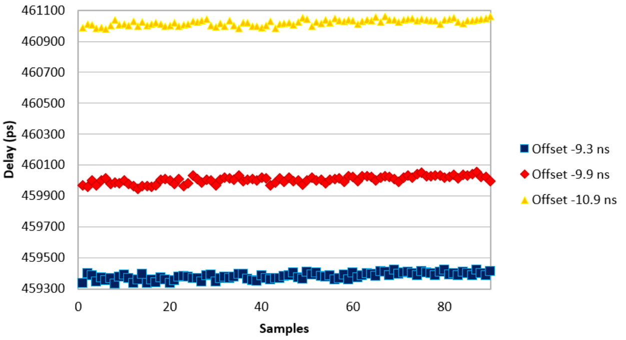


Figure 6 Δ_{T2-T1} results show the feasibility of the timestamper to detect propagation delay changes in the link using phase measurements.

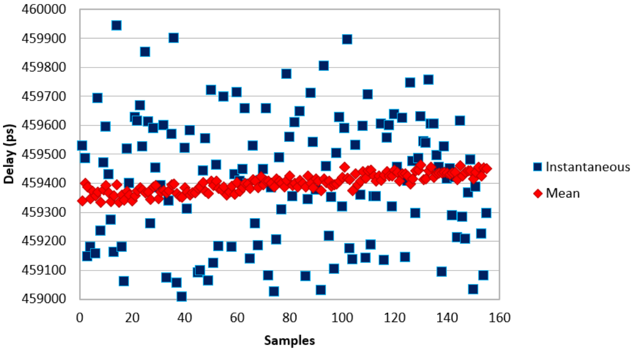


Figure 7 Δ_{T2-T1} results using instantaneous and mean N_A and N_B values show an immediate dynamic response from the timestamper to propagation delay changes in the link caused by temperature.

the effects of higher jitter conditions and asynchronous transmission frequencies. Therefore, it is demonstrated that the mean values improve the precision and will be used in this implementation for the rest of experiments. A rising trend is observed because of the time offset excursion between the devices during the elapsed measurement time and hence validates the need of immediate reaction of the PNTM timestamper and its precision. Larger dispersions in the σ_{com} in comparison to the previous setup are observed because of the excursion and higher clk_A jitter due to the 50 km fiber. This marks the importance of the chosen synchronization mechanism accuracy and characterize the expectable degradation for metropolitan area scenarios.

D. ABSOLUTE ACCURACY AND MULTI-BOX PRECISION TEST

In this experiment, a relative comparison between the timestamps captured by two WR-ZEN TP synchronized devices (T_B-T_A) has been performed to evaluate the consistency of the PNTM timestamping between devices on different locations and its accumulated precision. In order to simplify the test, the packet generator and the timestampers have been synchronized to the same WR reference to avoid frequency drifts as shown in Fig. 3.c. Note that the traffic is still generated in a synchronous way, so this experiment is still only dealing with the method validation on synchronous scenarios (thus without major benefits compared to the standard DDMTD implementation yet). A fiber splitter and symmetric fiber paths are used to duplicate the data packets and send them to the timestampers with equal latencies. A mean synchronization offset of -170 ps between the PNTM timestampers (B-A) and a σ_{sync} of 14 ps is measured with the counter. A fiber path distance difference to each timestamper of 1 cm is measured.

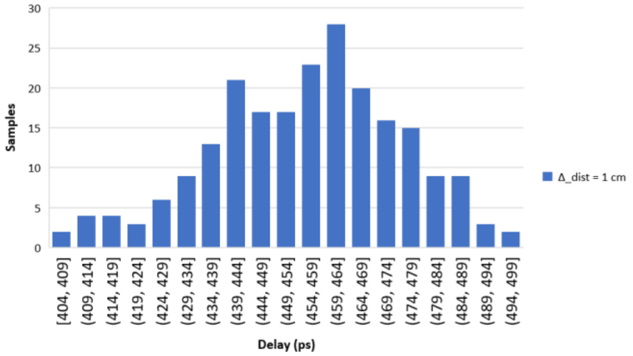


Figure 8 Timestamps offset results with symmetric paths show a 456 ps residual offset and a peak-to-peak dispersion below ±50 ps.

In Fig. 8, after removing the WR time offset, the observed mean offset between both timestamps is 456 ps with a σ_{com} of 19 ps which is explainable due to the additive noise caused by the relative comparison between two synchronized timestampers (σ_{tsA} and σ_{tsB}). This result complies with the sub-nanosecond accuracy specification for the WR-ZEN TPs and shows a σ_{ts} contribution of 5 ps when comparing both timestamps to the total σ_{com} after removing σ_{sync}.

The offset between the expected and the obtained result can be justified by uncontrolled asymmetries in the connectors or the fiber splitter, different SFPs latencies and the indeterminism between link reconnections in the FPGA transceiver as shown in Section IV.B. As the comparison is performed between timestamps taken by equivalent timestampers and the packet generator is synchronized, other effects as the fixed reception delays influence or the clk_A approximation are cancelled. Alternatively, these errors can be measured and removed with absolute calibration, as was stated in the methodology subsection.

As both PNTM timestampers share the same synchronization source, their noise contribution to the additive precision is not fully independent. This results in a slight impact in the obtained σ_{com} and proves the picosecond-level timestamping precision even between different devices.

E. ASYNCHRONOUS PACKET GENERATION TIMESTAMPING TEST

In the last experiment outlined in Fig 3.d, an asynchronous packet generator device has been used. The generated packets are timestamped by two PNTM synchronized timestampers and compared (T_B-T_A), obtaining a relative comparison of the timestamps. After showing that the implemented method works perfectly with synchronous traffic, it is time to demonstrate the method working on more generic and ambitious scenarios. The packet generator allows a ±4 ppm tuning of its transmission frequency with a

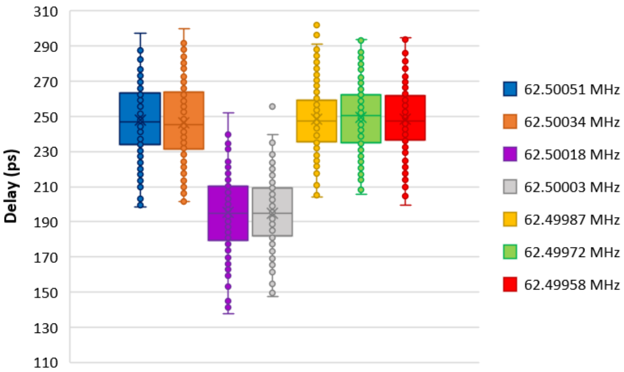


Figure 9 **Absolute timestamps offset results with different transmission frequencies using symmetric paths show sub-nanosecond determinism and single-digit ps timestamping precision.**

Δf_{ClkB} (Hz)	509	348	190	34	-120	-273	-415
T_B-T_A (ps)	247	246	195	195	248	250	248
Std. Dev. (ps)	20	21	22	19	18	18	18

Table II **Transmission frequency offset effect in T_B-T_A show the sub-nanosecond determinism and the single-picosecond precision for packet timestamping generated by asynchronous devices.**

voltage-controlled oscillator. A frequency sweep is performed to characterize the timestamping accuracy and precision. In the syntonized DDMTD case, this sweep would translate in time errors up to ±2.5 ns.

As shown in Fig. 9, the achieved T_B-T_A offsets maintain the expected sub-nanosecond accuracy for all the evaluated transmission frequencies. This confirms the reproducibility of the measurement and provides an indeterminism level overview in different setups, power cycles and measurement conditions. If required, a proper characterization and a calibration procedure could be performed to minimize this indeterminism and remove the existing offsets.

From Table II, the retrieved σ_{com} is consistent between different frequency differences to the nominal 1 Gbps frequency (Δf_{ClkB}), achieving a mean value of 19.5 ps in all the experiments. In comparison to the previous experiment, a minimum σ_{com} increase of 0.5 ps is observed. After removing the 14 ps reference σ_{sync}, the σ_{ts} result is always below 10 ps. This means that the individual σ_{tsA} and σ_{tsB} contributions are below 5 ps each, confirming the single-digit picosecond precision. This is coherent with the results obtained in previous experiments and validates that the timestamping precision remains practically the same when working with asynchronous packet generators compared with the synchronous case. A minimal degradation when working with asynchronous frequencies is observed, as it was claimed in the working hypothesis of this contribution. This test also corroborates that the assumption regarding the linear behavior of the clock drifts during the DDMTD integration time is rather valid. In fact, the main noise contributor to the σ_{com} in the different setups is the chosen synchronization protocol. This is an outstanding result and, to the best of our knowledge, better than any packet timestamping method shown on the literature [33], [35].

Finally, what is noteworthy is the flexibility of the design to work with different frequencies which enables its use in different Ethernet link speeds for packet timestamping of non-synchronized devices, overcoming the limitations of previous synchronous DDMTD implementations that required syntonization.

V. CONCLUSION

This paper has introduced a novel implementation of a packet timestamper called Precise Network Time Monitoring that uses phase measurements with picosecond level accuracy for systems that do not share a common time reference. It is based on a timestamping unit that includes a modified non-syntonized DDMTD implementation using a low-cost FPGA device and works at low clock frequencies without requiring external circuitry or advanced TDC chips. Our modified DDMTD technique does not require syntonized networks, opening the door to its integrability in applications like the FinTech industry, metrology or future 6G networks. Furthermore, PNTM is a low resource FPGA implementation, fully compatible with Ethernet networks, that is compatible with the current solutions widely used by most advanced HFT companies in electronic trading but providing more than one order of magnitude improvement in terms of precision.

A relative comparison measurement model which emulates real-world operating conditions has been chosen and the results have been carefully analyzed, decoupling aspects like accuracy, precision, or the synchronization impact on the timestamps.

The results show that PNTM provides 5 ps timestamping precision in the worst case, equivalent to the best TDCs, but using lower resources and avoiding calibration. Additionally, PTNM shows to be better than any current packet timestamping alternatives found in commercial solutions for non-intrusive timestamping in visibility networks.

Furthermore, this timestamper has been combined with the capabilities of high accuracy synchronization protocols such as WR, enabling packet capturing in distributed networks. This fact shows a new perspective in comparison to previous scientific contributions that have focused on local timestamping, but have not addressed distributed networks and therefore, their accuracy values are only applicable on single locations.

In this case, PNTM can directly provide sub-nanosecond timestamping accuracy by using proper time transfer and calibration techniques on distributed area networks. This facilitates the PNTM design being used for visibility purposes beyond local scenarios and outperforms commercial equipment which use standard PTP or analogic PPS distribution.

As a conclusion for this work, single-digit picosecond level timestamping precision and sub-nanosecond absolute accuracy are achieved representing, to the best of our knowledge, the most advance packet timestamping solution found in the literature. Moreover, it is shown that the proposed linearization of the frequency drift is experimentally supported, providing similar precision and accuracy to the syntonized DDMTD method.

Integration with software tools for easier management of the captured information and different Ethernet link speeds (10 Gbps or 25 Gbps) support are part of future work.

REFERENCES

[1] J. Sanchez-Garrido *et al.*, "Digital Electrical Substation Communications Based on Deterministic Time-Sensitive Networking Over Ethernet," in *IEEE Access*, vol. 8, pp. 93621-93634, 2020.

[2] R. Molina-Masegosa, J. Gozalvez and M. Sepulcre, "Comparison of IEEE 802.11p and LTE-V2X: An Evaluation With Periodic and Aperiodic Messages of Constant and Variable Size," in *IEEE Access*, vol. 8, pp. 121526-121548, 2020, doi: 10.1109/ACCESS.2020.3007115.

[3] "Editorial: Blockchain in Industrial IoT Applications: Security and Privacy Advances, Challenges, and Opportunities," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4119-4121, June 2020.

[4] S. Iqbal, A. H. Abdullah, K. N. Qureshi and J. Lloret, "Soft-GORA: Soft Constrained Globally Optimal Resource Allocation for Critical Links in IoT Backhaul Communication," in *IEEE Access*, vol. 6, pp. 614-624, 2018, doi: 10.1109/ACCESS.2017.2772867.

[5] D. Shrestha, Z. Pang and D. Dzung, "Precise Clock Synchronization in High Performance Wireless Communication for Time Sensitive Networking," in *IEEE Access*, vol. 6, pp. 8944-8953, 2018.

[6] Ó. Seijo *et al.*,"Enhanced Timestamping Method for Subnanosecond Time Synchronization in IEEE 802.11 Over WLAN Standard Conditions," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5792-5805, Sept. 2020, doi: 10.1109/TII.2019.2959200.

[7] A. Dogra *et al.*, "A Survey on beyond 5G network with the advent of 6G: Architecture and Emerging Technologies," in *IEEE Access*, Early Access, June 2020.

[8] I. Parvez *et al.*, "A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3098-3130, Fourth quarter 2018.

[9] D. P. Venmani *et al.*, On the Role of Network Synchronization for Future Cellular Networks: An Operator's Perspective, *IEEE Communications Standards Magazine*, Volume: 1, Issue: 1, 2017

[10] V. Moreno *et al.*, "Commodity Packet Capture Engines: Tutorial, Cookbook and Applicability," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1364-1390, 2015.

[11] Arista Metamux webpage, accessed on Dec. 24th, 2020, URL www.arista.com/en/products/metamux-network-switch-application

[12] Gigamon network taps webpage, accessed on Dec. 24th, 2020, URL https://www.gigamon.com/products/access-traffic/network-taps.html

[13] Cisco Nexus 3550-F Fusion HPT, accessed on Dec. 24th, 2020, URL https://www.cisco.com/c/en/us/products/collateral/switches/nexus-3550-series/datasheet-c78-743875.html?ccid=cc001479&oid=dstdnc022106

[14] LDA technologies Net-Recap: Network Analysis Toolset webpage, accessed on Dec. 24th, 2020, URL https://ldatech.com/solutions/MTCR.php

[15] Napatech smartNICs webpage, accessed on Dec. 24th, 2020, URL https://www.napatech.com/products/napatech-smartnics/

[16] D. He et al., "A Survey to Predict the Trend of AI-able Server Evolution in the Cloud," in *IEEE Access*, vol. 6, pp. 10591-10602, 2018.

[17] F. Girela-López *et al.*, "IEEE 1588 High Accuracy Default Profile: Applications and Challenges," in *IEEE Access*, vol. 8, pp. 45211-45220, 2020.

[18] M. Lombardi *et al.*, "Accurate, Traceable, and Verifiable Time Synchronization for World Financial Markets", Journal of Research of the National Institute of Standards and Technology, Volume 121, 2016

[19] CLOck NETwork - Services Strategy and innovation for clock services over optical-fibre networks, accessed on Dec. 24th, 2020, URL https://www.clonets.eu/clonets-home.html

[20] J. Vojtech *et al.*, "CLONETS-DS – Clock Network Services-Design Study: Strategy and Innovation for Clock Services Over Optical-fibre Networks", *Precise Time and Time Interval (PTTI)*, USA, 2021

[21] L. Sliwczynski *et al*., "Fiber-Based UTC Dissemination Supporting 5G Telecommunications Networks," in *IEEE Communications Magazine*, vol. 58, no. 4, pp. 67-73, April 2020.

[22] Z. Feng *et al*., "Coherent optical and RF receiver for simultaneously transferring frequencies in optical and RF domain," *2019 Joint Conference of the IEEE International Frequency Control Symposium and European Frequency and Time Forum (EFTF/IFC)*, USA, 2019, pp. 1-3.

[23] K. Jaldehag, P. Jarlemark and C. Rieck, "Further Evaluation of CGGTTS Time Transfer Software," *2019 Joint Conference of the IEEE International Frequency Control Symposium and European Frequency and Time Forum (EFTF/IFC)*, USA, 2019, pp. 1-6.

[24] T. T. Thai, G. Signorile and I. Sesia, "INRIM Multi GNSS All-in-View: Software and Results," *2019 Joint Conference of the IEEE International Frequency Control Symposium and European Frequency and Time Forum (EFTF/IFC)*, USA, 2019, pp. 1-3.

[25] M. Lévesque and D. Tipper, "A Survey of Clock Synchronization Over Packet-Switched Networks," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2926-2947, Fourthquarter 2016.

[26] F. M. Anwar, A. Alanwar and M. B. Srivastava, "OpenClock: A Testbed for Clock Synchronization Research," *2018 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, Geneva, 2018, pp. 1-6, doi: 10.1109/ISPCS.2018.8543080.

[27] J. Serrano *et al.*, "The White Rabbit Project" in *Proceedings of ICALEPCS TUC004*, Kobe, Japan, 2009.

[28] J. L. Gutiérrez-Rivas *et al.*, "White Rabbit HSR: A Seamless Subnanosecond Redundant Timing System with Low-Latency Data Capabilities for the Smart Grid," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3486-3494, Aug. 2018.

[29] Bittware timestamping webpage, accessed on Dec. 24th, 2020, URL https://www.bittware.com/resources/timestamping/

[30] M. Brückner and R. Wischnewski, "A Time Stamping TDC for SPEC and ZEN Platforms Based on White Rabbit" in *Proceedings of ICALEPCS THPHA088*, Barcelona, Spain, 2017

[31] Y. Hu *et al.*, "Performance Evaluation of Time Distribution over SerDes-based Interconnections for PET System", in *IEEE Nuclear Science Symposium and Medical Imaging Conference*, Australia, 2018

[32] R. Szplet *et al.*, "An Eight-Channel 4.5-ps Precision Timestamps-Based Time Interval Counter in FPGA Chip," in *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 9, pp. 2088-2100, 2016.

[33] S. Tancock, E. Arabul and N. Dahnoun, "A Review of New Time-to-Digital Conversion Techniques," in *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 10, pp. 3406-3417, 2019.

[34] J. Sánchez-Garrido *et al*., "A White Rabbit-Synchronized Accurate Time-Stamping Solution for the Small-Sized Cameras of the Cherenkov Telescope Array," in *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-14, 2021, Art no. 2000314, doi: 10.1109/TIM.2020.3013343.

[35] P. Loschmidt, Patrick, R. Exel, and G. Gaderer, "Highly Accurate Timestamping for Ethernet-Based Clock Synchronization" in *Journal of Computer Networks and Communications*, 2012.

[36] M. Rizzi. "White Rabbit Clock Synchronization: Ultimate Limits on Close-In Phase Noise and Short-Term Stability Due to FPGA Implementation" in *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*. PP. 1-1. 10.1109/TUFFC.2018.2851842.

[37] T. Wlostowski, Precise time and frequency transfer in a White Rabbit network, Master's thesis, Warsaw University of Technology (2011).

[38] IEEE 802.3-2018 - IEEE Standard for Ethernet

[39] M. Jiménez-López *et al*., "10 Gigabit White Rabbit: Sub-Nanosecond Timing and Data Distribution" in *IEEE Access*, vol. 8, pp. 92999-93010, 2020.

[40] J. Somers, Highly Accurate Synchronization Over Ethernet, Master's thesis, Delft University of Technology (2015).

[41] Seven Solutions Timing Products webpage, accessed on Dec. 24th, 2020, URL https://sevensols.com/index.php/timing-products/

[42] Seven Solutions WR-ZEN TP webpage, accessed on Dec. 24th, 2020, URL https://sevensols.com/index.php/products/wr-zen-tp/

[43] M. Jiménez-López *et al.*, "A Fully Programmable White-Rabbit Node for the SKA Telescope PPS Distribution System," in *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 2, pp. 632-641, Feb. 2019.

[44] Keysight 53210A webpage, accessed on Dec. 24th, 2020, URL https://www.keysight.com/en/pd-1893374-pn-53210A/350-mhz-rf-frequency-counter-10-digits-s?cc=ES&lc=eng

BIOGRAPHIES

FRANCISCO GIRELA-LÓPEZ (girlop@ugr.es) is a PhD student at University of Granada, where he also obtained his Master of Telecommunications Engineering degree. He has worked on synchronization, positioning and security tasks, but he is mainly focused on time transfer systems based on WR technology. ORCID: 0000-0003-2849-1721

EDUARDO ROS (eros@ugr.es) is Full Professor at the University of Granada (Department of Computer Architecture and Technology) since 1999. He has participated as IP of the University of Granada in 7 EU grants in different areas (among them ultra-accurate Time Transfer and Synchronization) and he has an extensive scientific production (more than 70 journal papers) and 3 patents (extended PCT). ORCID: 0000-0001-6613-5256

JAVIER DÍAZ (jda@ugr.es) received his MS in Electronics Engineering and a Ph.D. in Electronics both from the University of Granada. His main interests are image processing architectures, safety-critical systems, highly accurate time synchronization and frequency distribution techniques. He is a University professor and collaborates with research facilities integrating sub-nanosecond time transfer based on WR technology. ORCID: 0000-0002-1849-8068