Implementation and Evaluation of a PTP Transparent Clock Based on White Rabbit Technology

Cesar Prados* ‡
*GSI,Darmstadt Germany
‡Technical University Darmstadt, Darmstadt Germany

***Abstract*—The Rabbit is a new technology that allows to create an Ethernet network with low-latency, deterministic packet delivery and sub-nanosecond accuracy and picoseconds precision synchronization. WR is based on the standards Synchronous Ethernet and IEEE 1588-2008, PTP. Currently the synchronization of the clocks throughout the network is achieved using WR Switches, two-step Boundary Clocks (BC). In this paper we describe the implementation of a Transparent Clock (TC) (both types Peer-to-Peer and End-to-End) based on WR Link Model and existing hardware.**

Finally, we evaluate the suitability of the WR TCs in the Control and Timing System for the new Antiproton and Ion Research facility (FAIR) in GSI.

I. Introduction

Synchronization is an important feature in control and timing systems. The performance depends on the synchronization accuracy between the networking elements throughout the network, specially in systems where time-critical events are accurately scheduled to be executed in distributed nodes.

White Rabbit (WR) technology provides allows to create a reliable and robust Ethernet network with low-latency, deterministic packet delivery and synchronization. WR is based on the standards:Synchronous Ethernet (SyncE) [1], Ethernet (IEEE 802.3) [2]. Besides, it extends IEEE 1588 (PTP) [3] for achieving subnanoseconds accuracy and picoseconds precision. WR will be the control and timing system at GSI [4] and CERN [5]. There are others instituted and facilities (LHAASO, KM3NeT etc...) that are taking an active part in the WR Project [6], as well as commercial companies (Seven Solutions, Integrasys, Elproma, Creotech, National Instruments etc..).

So far in WR, the synchronization is propagated from the Master Clock, WR Master, to the Slaves Clock, WR Slave, in the network using White Rabbit Switches (WRS). A WRS, in terms of PTP, is a two-step Boundary Clocks using the Delay Request-Respond Mechanism [3] for the synchronization. In this paper, the author proposes an implementation of a Transparent Clock (TC) using the existing WR extension to PTP, WRPTP [7] and the existing hardware with WR support [8], [9], and evaluates the principle features that characterizes the performance of a TC.

Since GSI will adopt WR technology for the new control and timing system, the author presents a timing topology making use of TC based on WR.

II. White Rabbit PTP

A. White Rabbit Synchronization

White Rabbit is able to reach sub-nanoseconds accuracy and picoseconds precision between two clocks:

- Characterizing the asymmetry of the links
- Measuring the phase shift of reference clock using clock loopback technique.

The Fig. 1 shows the WRPTP [10] flow and its integration with the standard PTP . Bellow, a resume of the steps the WR synchronization process is outlined. A thoroughly description is presented in [11] and [7].

1) Synthonization: WRPTP initiates with a WR Announcement, in order to discover WR devices. If the other node is a WR device, the Lock procedure is triggered by the master. WRPTP uses SyncE to distribute a common frequency throughout. The slave will be frequency locked to the master. After what, both clocks will calibrate asymmetries present in the link.

2) Asymmetry Calibration: WR calibrates the asymmetries in a optical link taking in consideration:

- Fixed delays due to transmission circuitry
- Asymmetry of the optical transceivers and PHYs
- Chromatic dispersion of the fiber optic

3) Coarse and Fine Delay Measurement: After the calibration process, a first delay measurement, *Coarse Measurement* is issued using the Delay Request-Respond (DRR) mechanism. The next step towards the synchronization requires to figure out the phase shift between the master and the looped-back clock from the sender, round trip phase shift, $phase_{mm}$. The 3 shows how between the master and the TC, the clock is looped back and the phase shift is measured using a phase detector, DDMTD [12]. With the $phase_{mm}$, the delay round trip, $delay_{mm}$ is calculated. The $phase_s$ is the phase shift of the clock adjustment derived form the $offset_ms$. Using the $phase_{mm}$ and $phase_s$ the timestamps on ingress ports can be enhanced, t_{4p} t_{2p}, using an algorithm described in [11]. Only timestamps on ingress ports need to bed enhanced, since they are generated asynchronously to the reference clock domain. The *Fine Delay Measurement*, round trip, is calculated as follows using now the enhanced timestamps :

$$delay_{mm} = (t_{4p} - t_1) - (t_3 - t_{2p}) \tag{1}$$

4) Synchronization: In order to finish the synchronization of the Slave clock to the Master Clock the offset between both the clocks must be calculate, $offset_{MS}$. The round trip can be expressed as function of the delay master to slave, σ_{ms} , slave to master $sigma_{sm}$ and the sum of the fixed delays, δ , obtained during the calibration.

$$delay_{mm} = \Delta + \sigma_{ms} + \sigma_{sm} \tag{2}$$

The ratio between single delays corresponds to the asymmetry of the propagation speed of the different wavelength due to the chromatic dispersion, and is expressed:

$$(\alpha - 1) = \frac{\sigma_{ms}}{\sigma_{sm}} \tag{3}$$

From (1), (2) and (3) the delay master to slave and $offset_{MS}$:

$$delay_{ms} = \frac{1+\alpha}{2+\alpha}(delay_{mm} - \Delta) + \Delta_{txm} + \Delta_{rxs} \tag{4}$$

$$offset_{ms} = t_1 - t_{2p} - delay_{ms} \tag{5}$$

After the initial WRPTP synchronization, a DDR or Peer Delay mechanism produces the timestamps, t_1, t_{2p}, t_3 and t_{4p} (in case Peer Delay, also t_5 and t_{6p}) and the DDMTD tracks changes in the $phase_{mm}$ over time.

III. WR Transparent Clocks

PTP Transparent Clock (TC) modifies PTP messages as they pass through them adding the residence time of the packet inside of the device. Thus, the delay introduced by the network is measured and can be subtracted in the slave clock, which improves distribution accuracy.

In II the WRPTP and WR synchronization steps has been described for master, slave and boundary clock. In this chapter the author presents how a TC uses of the WR synchronization and WRPTP.

A. End-to-End WR Transparent Clock

As the standard End-to-End (E2E) clocks, the WR E2E TC doesn't belong to the master-slave hierarchy and does not synchronized to the WR Master. Therefore, a WR E2E accomplishes only the synthonization and the calibration, and not the rest of the steps since the $delay_{ms}$ is calculate end to end, between master and slave. Thanks to the synthonization there is not errors in the measurement of the residence time and in the accumulative Correction Field(CF). In the figure 2 shows the flow of messages and how the residence time is also calculated taking into account the fixed delays Δ. A two-step WR E2E TC calculates, using (4), the $offset_{ms}$ [1]:

$$CF+ = (TS_{ingess\_port} - \Delta_{rx}) - (TS_{egress\_port} + \Delta_{tx}) \tag{6}$$

$$offset_{ms} = t_1 - t_2 - delay_{ms} - CF \tag{7}$$

[1]No enhancement in the timestamps, in chapter IV this issued will be analyzed

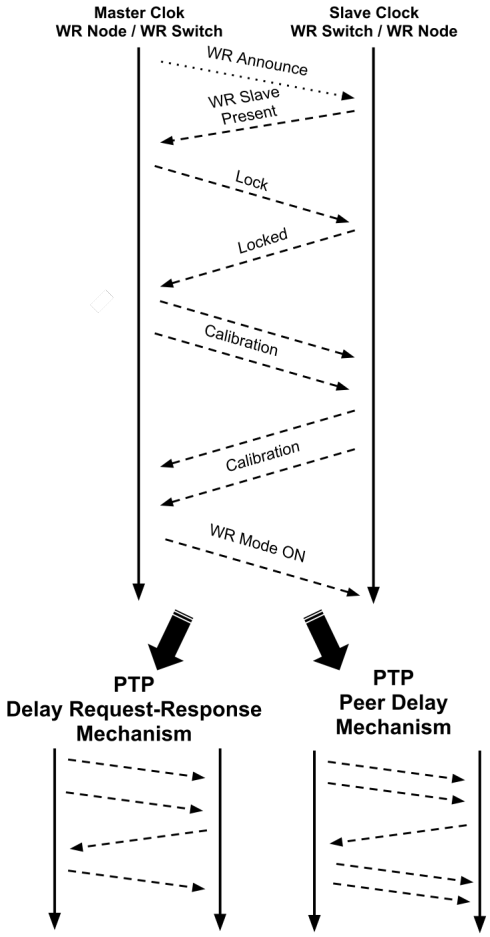

Fig. 1. WR PTP Message Flow and PTP

B. Peer-to-Peer WR Transparent Clock

As the standard Peer-to-Peer (P2P) clocks, the WR P2P TC measures residence time of Sync messages, plus the link delay in both directions. Since the link delay is done between adjacent clocks using the PD mechanism, the fully WR synchronization process can be issued between ports. The Figure 2 shows how the WR P2P initiates with the WRPTP one once the WR Mode is on, and continues afterwards doing the PD mechanism. The steps explained in II occurs also in the TC, thanks to the WR technology.

The measurement of the residence time, like in the E2E should suffer error since the port is syntonized to it master clock, but also the measurement of the link delays as precise as the WR project claims [10] for the Boundary Clocks. The link delay is calculate like in (4) and the fig 3 but using t_3, t_{4p}, t_5 and t_{6p}. The $offset_{ms}$ is calculated :

$$CF+ = residance\_time + delay\_link \tag{8}$$

$$offset_{ms} = t_1 - t_2 - CF - delay\_link \tag{9}$$

[2]This link delay correspond to the las link to the slave

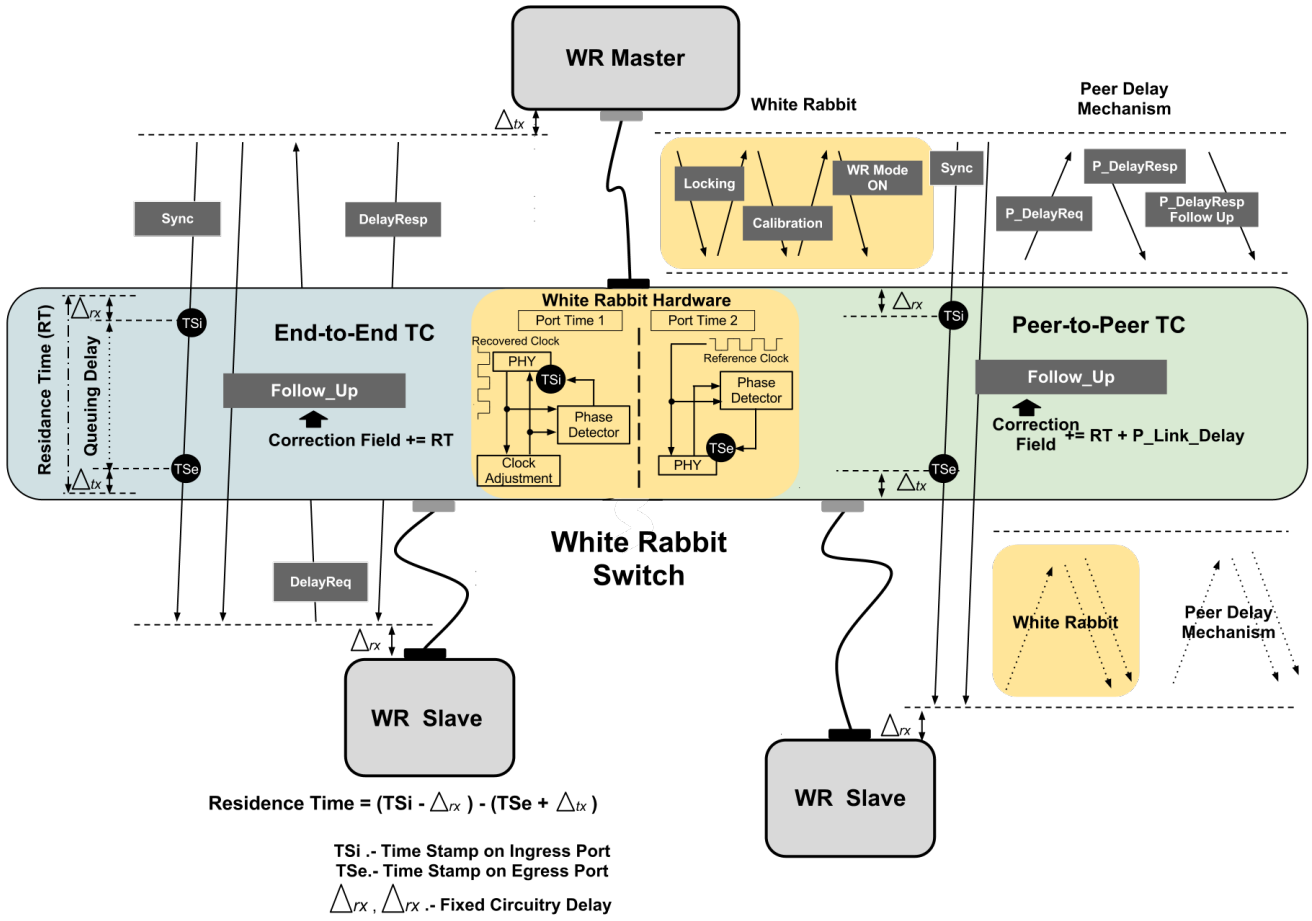

Fig. 2. WR E2E and P2P Transparent Clocks

IV. Implementation Issues and Performance Estimation

The performance of a TC is commonly related to [13], [14] following features:

- Timestamps accuracy
- Correction factor stability
- Maximum update rate
- Rapid reconfiguration after topology changes

It applies as well to a WR TC. Although, WRS has been deceived to be compliant to most common standards related to GbE (e.g. VLAN tags and Quality of Service) and fulfil demanding requirements in terms of upper-bound delivery latency and fault tolerance. This features that can be exploited and used to achieve remarkable performance as a TC.

A. Time Stamping Accuracy

In the chapter II, the author resumes how WR accomplish sub-nanoseconds synchronization and picoseconds jitter. It is based on the characterization the asymmetries of the link a priory, and a clock lookback technique. By doing this WR is able enhance the timestamps on ingress ports.

The figure 3 shows that the Peer Delay timestamps between two adjacent nodes can be enhanced since the $pahse_{MM\_1}$ and $phase_{S\_1}$ to the reference master clock signal are known. But it is not the case for t_2, this timestamp can be only enhanced using $phase_{S\_2}$ since the reference clock signal used in this link, is a synchronized clock to reference master clock signal (the same applies for t_4, in a E2E TC). The error introduced if the t_2 is enhanced using $phase_{S\_2}$ and not $phase_{S\_1}$ is:

B. Correction factor stability and Maximum Update Rate

C. Stability and Continuity of the Synchronization

P2P

D. WR Hardware and Software TC

Currently the WR Switch V3 [9] behaves as Boundary clock. After an exhaustive examination of the WR project, the following features should be added to the existing gateware and software (no modifications in the hardware) for a proof of concept WR TC implementation:

- Peer Delay Mechanism
- WR TC clock behaviour

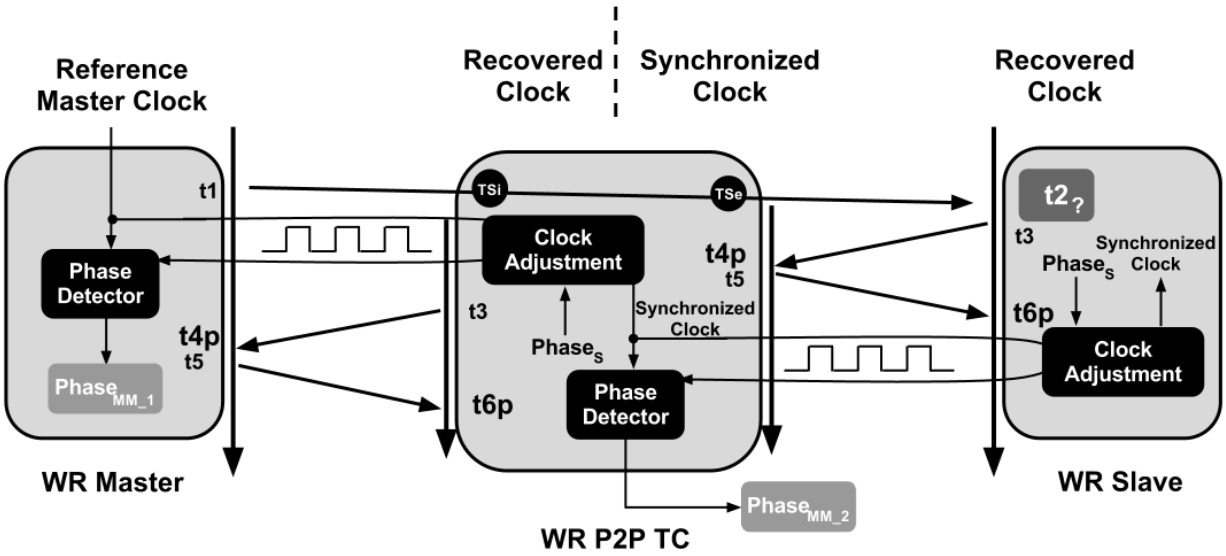

Fig. 3. WR synchronization and P2P TC

- Hardware process of Sync and Follow Up messages
- Enhanced Timestamping across TC using WR

The openness of the project [3], plus the amount of work already done, makes the implementation achievable with a reasonable effort.

V. FAIR and Transparent Clocks

A. Timing System Requirements

B. WR Network in FAIR Timing System

VI. Conclusions

[3]The WRS is licensed under *CERN Open Hardware Licence* [15] and is publicly available in the Open Hardware Repository.

VII. Acknowledgment

The author would like to recognize the vast work done by the WR Community and specially to, T. Włostowski and M. Lipiski, for their support, design and code of the White Rabbit Switch, on which a I have based this paper.

References

[1] *Timing characteristics of a synchronous Ethernet equipment slave clock (EEC)*, ITU-T Std. G.8262, 2007.

[2] *IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems–Local and Metropolitan Area Networks–Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Section Three*, IEEE Std. 802.3-2008, 2008.

[3] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Std. 1588-2008, 2008.

[4] T. Fleck, C. Prados, S. Rauch, and M. Kreider, "FAIR timing system," GSI, Darmstadt, Germany, Tech. Rep., 2009, v1.2.

[5] J. Bau and M. Lipiński, "Discussion On A White Rabbit based CERN Control and Timing Network," http://www.ohwr.org/documents/85, October 2011, v1.1.

[6] "White Rabbit Project. http://www.ohwr.org/projects/white-rabbit.

[7] E. Cota, M. Lipiński, T. Włostowski, E. Bij, and J. Serrano, "White Rabbit Specification: Draft for Comments," http://www.ohwr.org/documents/21, july 2011, v2.0.

[8] Simple PCIe FMC carrier (SPEC). http://www.ohwr.org/projects/spec.

[9] White Rabbit Switch v3. http://www.sevensols.com/whiterabbitsolution/.

[10] M. Lipiski, T. Wostowski, J. Serrano, and P. Alvarez, "White Rabbit: a PTP application for robust sub-nanosecond synchronization," *Proceedings of ISPCS*, 2011.

[11] T. Włostowski, "Precise time and frequency transfer in a White Rabbit network," Master's thesis, Warsaw University of Technology, may 2011.

[12] P. Moreira, P. Alvarez, J. Serrano, I. Darwezeh, and T. Wlostowski, "Digital Dual Mixer Time Difference for Sub-Nanosecond Time Synchronization in Ethernet," *Frequency Control Symposium (FCS), 2010 IEEE International*, 2010.

[13] J. Burch, K. Green, J. Nakulski, and D. Vook, "Verifying the performance of transparent clocks in ptp systems," *Proceedings of ISPCS*, 2009.

[14] J. Han and D.-K. Jeong, "A Practical Implementation of IEEE 1588-2008 Transparent Clock for Distributed Measurement and Control Systems," *IEEE Transactions on Intromentation and Measurement*, vol. 59, no. 2, 2010.

[15] CERN Open Hardware Licence.