Network Time Synchronization for Detector Data Acquisition Electronics

W. Hennig, XIA LLC, 31057 Genstar Rd, Hayward, CA 94544, USA

2017 IEEE NSS
N-23-036, # 3698
whennig@xia.com
www.xia.com

Background

Large scale nuclear physics experiments often use arrays of radiation detectors, which can be physically separated, for example in separate rooms or along a beamline. Time synchronization of the detector readout electronics is essential to detect related events, which is often accomplished by sharing clock, trigger and reset signals within and between racks of digitizing electronics. This works well over short distances, but requires dedicated cabling and/or modules and becomes cumbersome for widely separated arrays.
As the detector readout electronics is in many cases operated by computers linked over standard data networks, an alternative to dedicated clock distribution trees is the synchronization of clocks over the data network. Network time synchronization protocols like the IEEE 1588 Precision Time Protocol (PTP) [1] or White Rabbit (WR) [2] are reported to achieve low/sub nanosecond timing resolution, but are not always tested in nuclear physics applications such as timing of coincident pulses.

Approach

Our goal in this project is to explore several network time synchronization techniques and quantify their performance for nuclear physics data acquisition, making use of existing hardware and software where possible. Specifically, PTP is widely used for network time synchronization and has several open source implementations. The recently developed XIA Pixie-Net spectrometer [3] is a network based data acquisition module for detector readout and pulse processing. The work reported here therefore consisted of adding various PTP time stamping unit (TSU) implementations to the Pixie-Net, and characterize the time synchronization in detector measurements. For reference, equivalent measurements were performed with a Pixie-4e shared clock configuration and an external clock derived from a WR demo kit. The techniques tested were:

1. Software TSU provided by Linux software (ptpd) [4]
2. Hardware TSU built into the Pixie-Net's Zynq processor [5]
3. Hardware TSU built into the TI DP83640 Ethernet PHY. [6]
   … optionally with synchronized Ethernet (sync-E).
4. Firmware TSU in the Pixie-Net's FPGA using a SoC-e IP core [7]
5. External clock derived from a WR demo kit [8]
6. Pixie-Net or Pixie-4e shared clock [9]

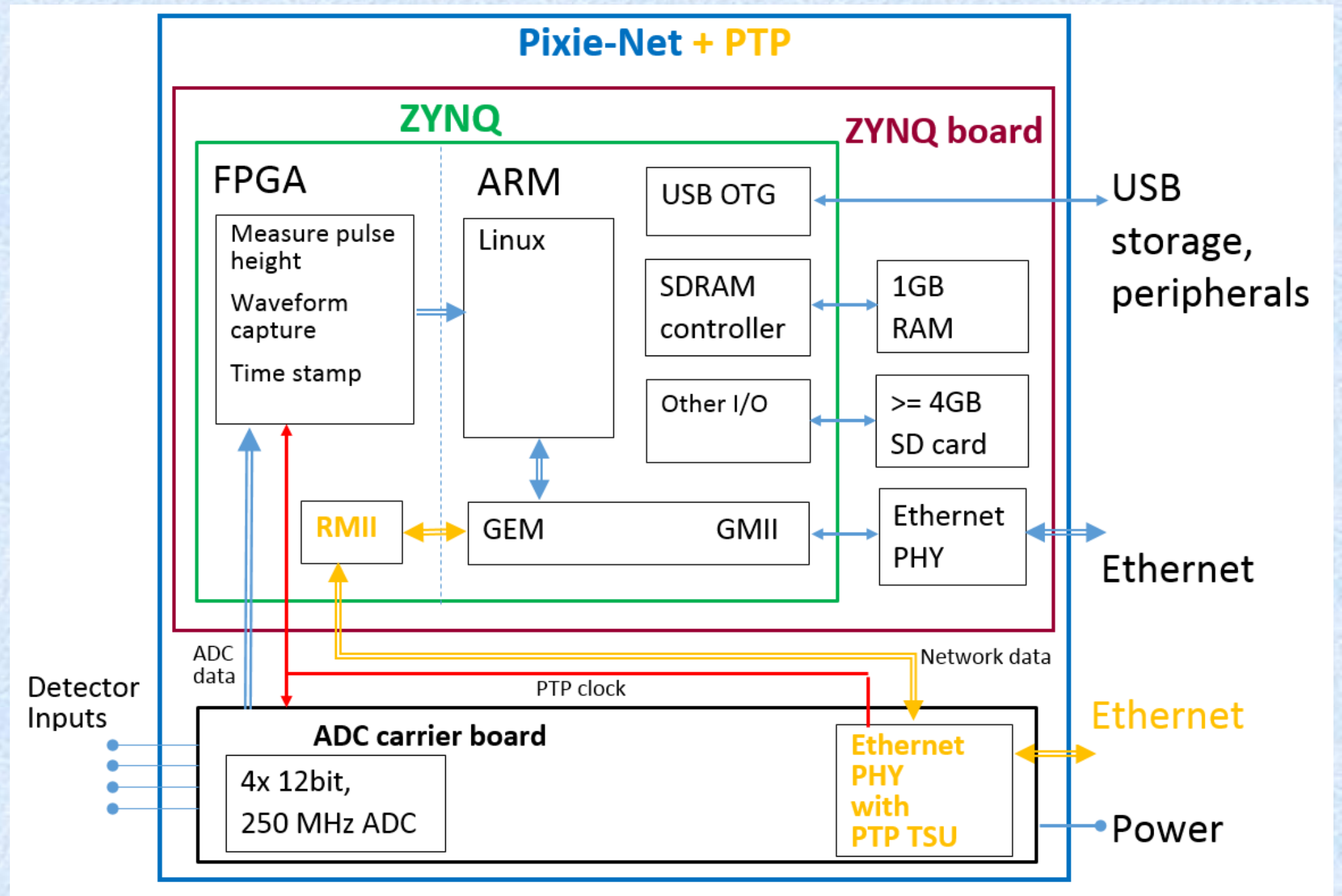


Figure 1. Block diagram and picture of Pixie-Net, modified for DP83640 PHY TSU.

As an example, the implementation for technique 3 is shown in Figure 1. The experimental setup to measure the time resolution consisted of a pair of $LaBr_3$ scintillators illuminated by a Na-22 source and read out by a pair of Pixie-Nets synchronized over the network (Figure 2). Two channels of a Pixie-4e were connected in parallel for reference. The network connection was either

> **Non-PTP synchronized**: connecting each Pixie-Net to the local network that has no PTP capable routers and switches. There will be unpredictable delays and latencies as PTP messages are transmitted through the network.
>
> **PTP synchronized**: connecting the two Pixie-Net modules back to back. This is the optimum case of a network in which each node is PTP synchronized with its neighbors, essentially as if all routers and switches would be PTP enabled. Network delays to the next node are minimal and consistent, and each device only has to synchronize with the next node. In cases where synchronized devices are separated by more than one node, we expect timing uncertainties to add for each node

The open source software ptpd or LinuxPTP [10] was installed on the Pixie-Net's Zynq ARM processor to manage the PTP synchronization. The measurement setup is shown in Figure 2.

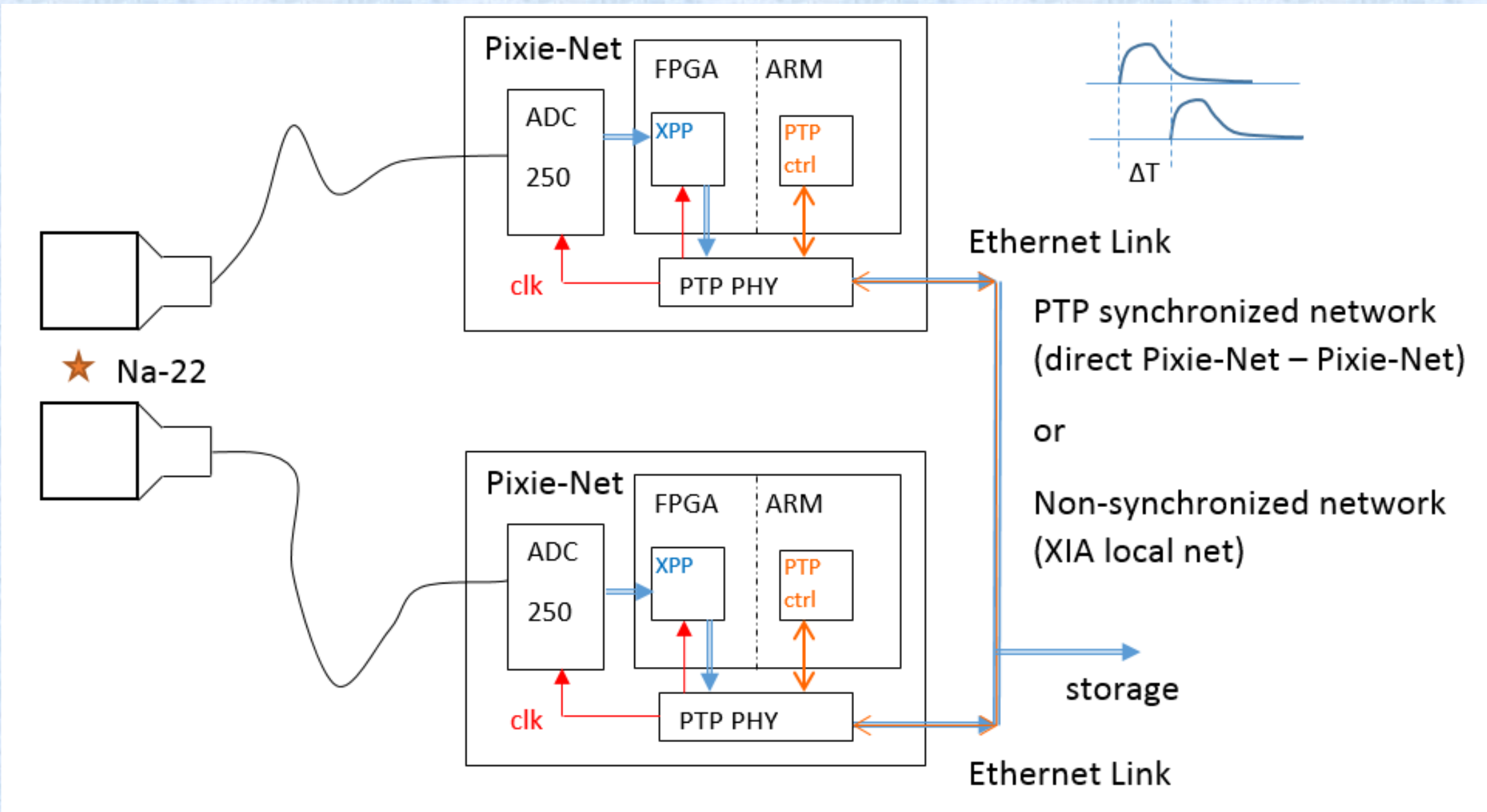


Figure 2. Experimental setup for time resolution measurements. A pulser was used in place of the scintillator pair in cases where the detector limited the precision (due to pulse shape variations from noise, light collection, etc).

Random Coincidence Rejection

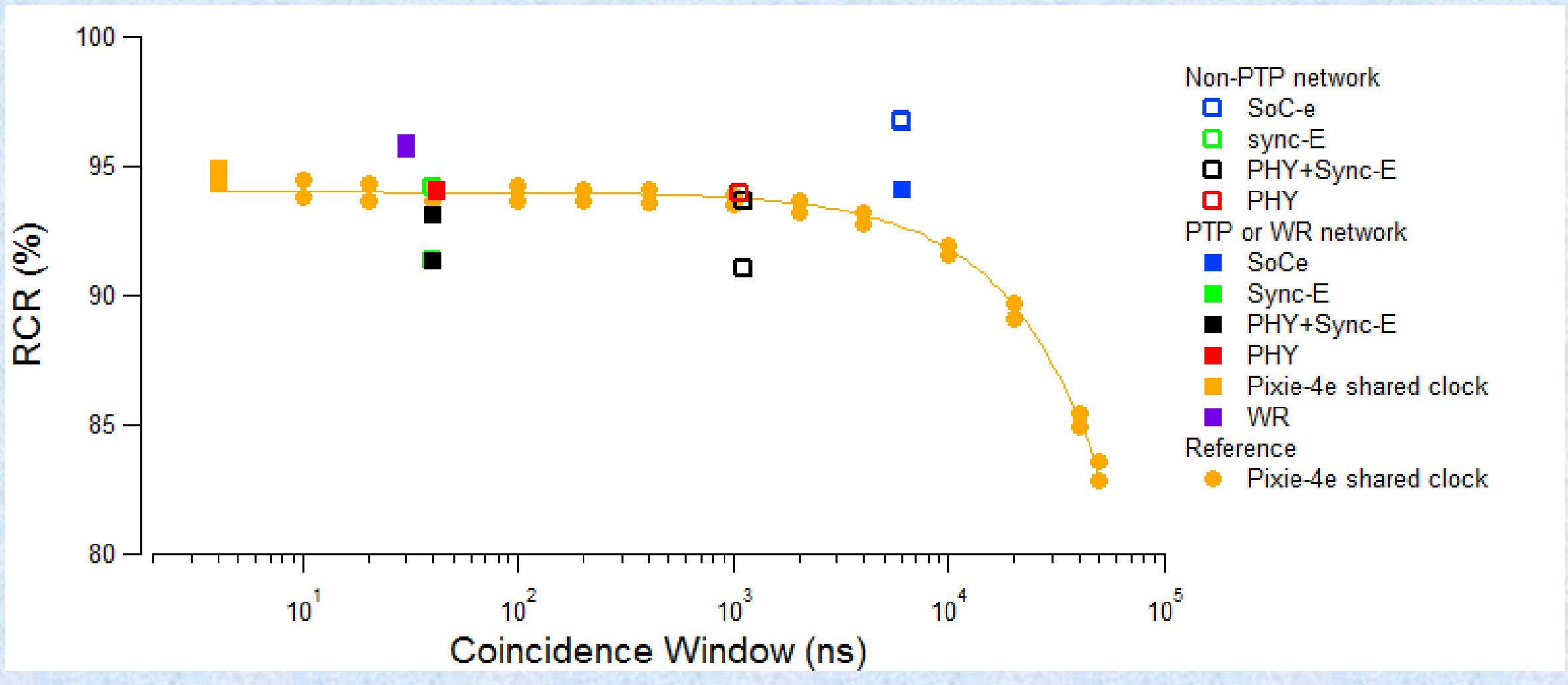


Figure 3. Rejection of random coincidences as a function of coincidence window. Minimum window is set by timing resolution

In Na-22 measurements, we can distinguish "unwanted" random coincidences e.g. from background as those pulse pairs detected within a coincidence window with E > 511 keV. The ratio of the number of these unwanted coincidences (Nu) to the number of (all) singles events with E> 511 keV (Na) is the fraction of unwanted coincidences. We define the random coincidence rejection RCR as

$$RCR = 1 - Nu/Na$$

RCR is primarily a function of the coincidence window width, but as less precise synchronization requires a larger window, it serves as our third measure of performance for the synchronization techniques.
Only the SoC-e TSU is close to the "edge" of allowing significantly more random coincidences into the spectrum, which implies that it is not always necessary to have the best possible timing resolution. This can save significant costs, for example by not requiring a network with PTP routers/switches.

References and Acknowledgments

[1] https://en.wikipedia.org/wiki/Precision_Time_Protocol
[2] https://www.ohwr.org/projects/white-rabbit/wiki
[3] http://www.xia.com/Pixie-Net.html
[4] https://sourceforge.net/projects/ptpd/files/
[5] "Zynq-7000 AP SoC - Precision Timing with IEEE1588 v2 Protocol", www.xilinx.com
[6] www.ti.com/product/DP83640/technicaldocuments
[7] http://soc-e.com/ieee-1588v2-ordinary-and-boundary-clock/
[8] http://sevensols.com/index.php/products/wr-len/
[9] equivalent to A. Fallu-Labruyere et al, NIM A 579 (2007), p247.
[10] http://linuxptp.sourceforge.net/

This work was supported by the US DOE through award DE-SC0017223.

Time Resolution from Software Log

The LinuxPTP and ptpd software periodically report the measured clock offset between network master and network slave clocks. This is derived from the PTP messages and useful as a first measure to evaluate performance. Histogramming hundreds or thousands of reported offset values gives a distribution around an average value, and the FWHM of the distribution is our first measure of performance for the synchronization techniques.

- ➢ **Resolutions in the PTP network are an order of magnitude better than in the non-PTP network.**
- ➢ **Resolutions can reach ~10ns FWHM or below.**
- ➢ **PHY timestamping is better than FW timestamping which is better than SW timestamping**

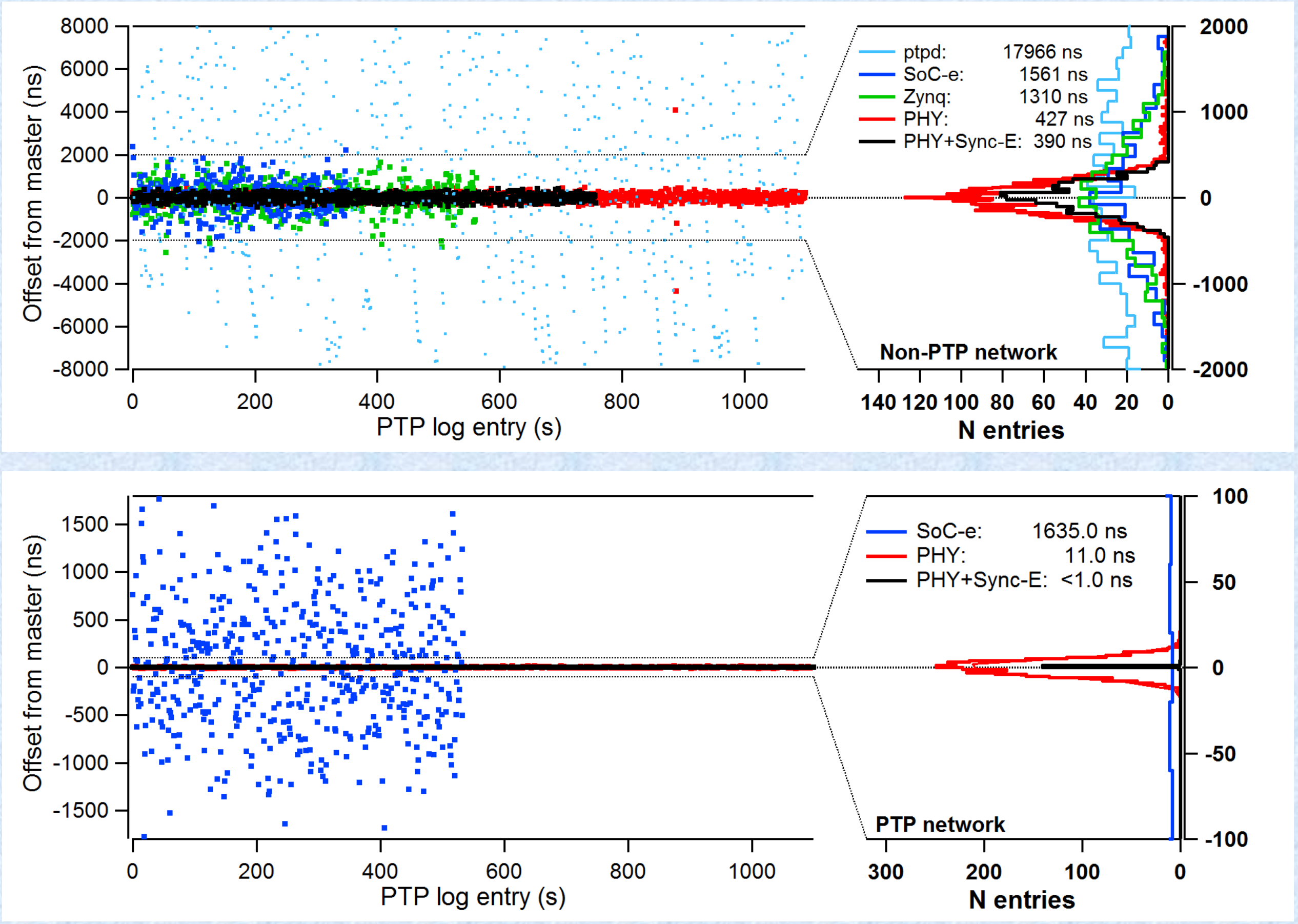


Figure 4. Time offsets reported from PTP stack software and histograms for the overall measurement. Top: Non-PTP network Bottom: PTP network

Note the right y scale is magnified for better detail of the histograms, as indicated by the grid lines.

Time Resolution from Coincident Pulses

When coincident scintillator pulses are captured in the two Pixie-Net modules, with time stamps T1 and T2, there will be a small difference in time stamps ΔT = T2-T1 (from cable delays etc). ΔT is nominally a constant (cables don't change) but with small variations due to imperfect clocking and variations in pulse shape. Histogramming ΔT from hundreds of thousands of pulses and applying a Gaussian fit to determine the FWHM distribution (Figure 6) is our second measure of performance for the synchronization techniques. A CFD algorithm is applied in cases where the PTP precision approaches the clock period.

- ➢ **ΔT drifts in the same direction for short sections of the measurement (Figure 5), following the periodic adjustments of the PTP clock.**
- ➢ **Resolutions with sync-E or WR are an order of magnitude better than in the PTP network.**
- ➢ **Resolutions can reach ~200 ps FWHM.**
- ➢ **Detector jitter can be the limit for achievable precision, not the timing technique.**

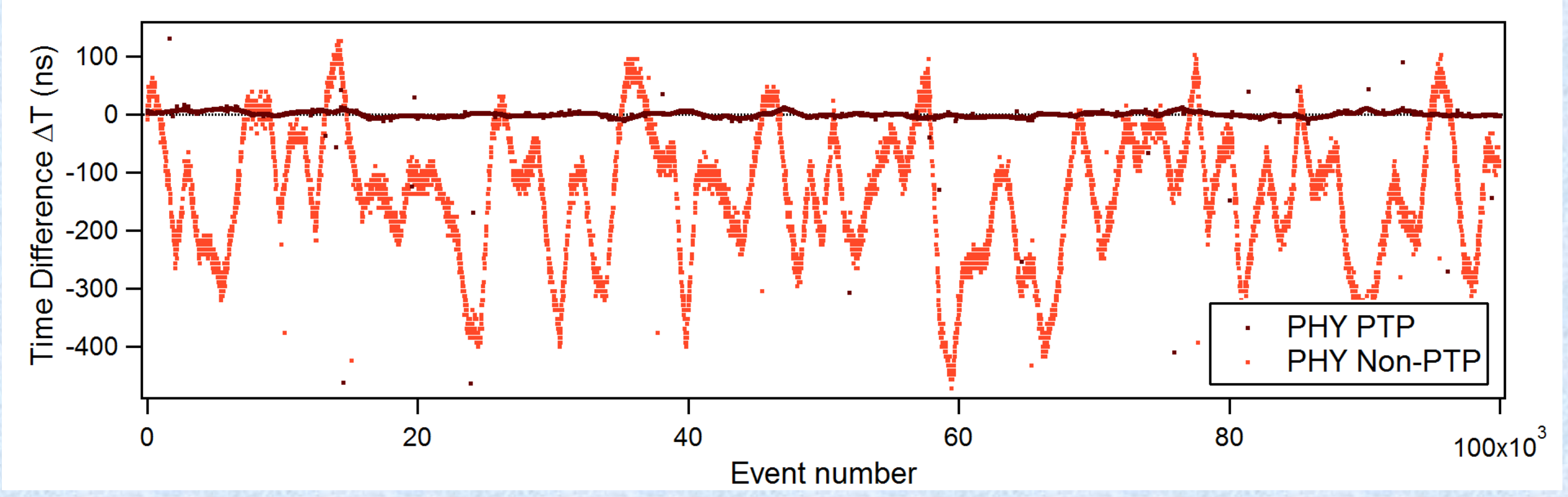


Figure 5. Magnified section of the coincidence pulse ΔT to show the structure of periodic adjustments. (From DP83640 PHY $LaBr_3$ tests).

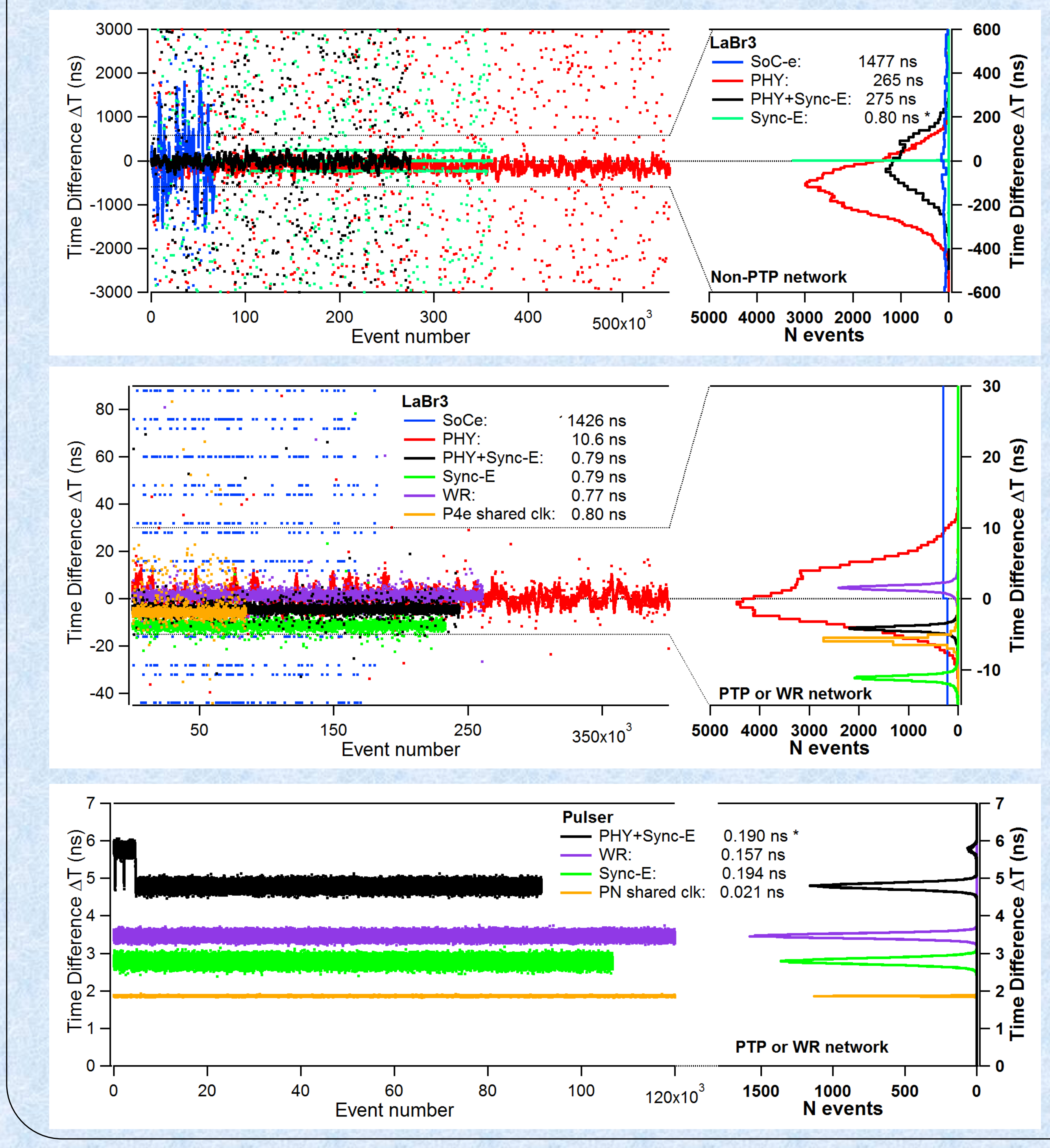


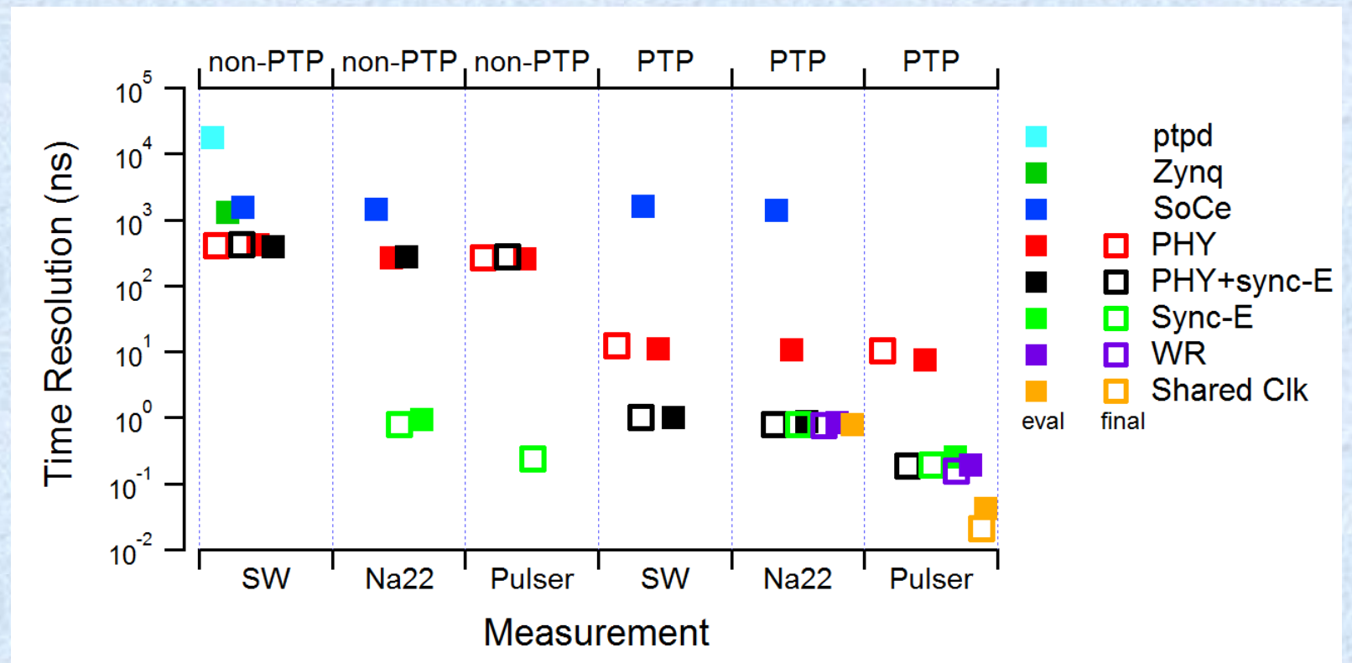
Figure 6. Coincidence pulse ΔT for the various synchronization techniques, and histograms of the overall measurement.

Top: Non-PPT network with scintillator

Middle: PTP or WR network with scintillator

Bottom: PTP or WR network with pulser

Summary and Conclusions



The two most promising network time synchronization techniques for nuclear physics applications were WR and PTP PHY timestamping with the DP83640. The DP83640 obtained time resolutions of ~10 (~400) ns in a (non-)PTP network. In sync-E mode, it reached ~800ps with a pair of $LaBr_3$ detectors and ~200ps with a pulser. With the WR demo kit, resolutions were ~800ps with a pair of $LaBr_3$ detectors and ~190ps with a pulser. A traditional distributed clock reached ~800ps with a pair of $LaBr_3$ detectors and ~20 ps with a pulser.

The DP83640 is therefore a good solution for applications that do not need the most precise timing (e.g. coincidence background rejection); it is a $10 part that can easily be integrated into the DAQ electronics and does not necessarily need a special network infrastructure. For more demanding applications, such as detector array event building, the DP83640 can be used in a PTP enabled network (which adds cost to the infrastructure, but not the electronics). For highest precision applications, such as time-of-flight measurements, White Rabbit or synchronous Ethernet can be used, but traditional shared clock solutions still provide better performance which may be necessary for some applications (where the detectors support it).

This precision network time synchronization allows a new kind of data acquisition, with software based trigger and recording decisions rather than hard wired logic.