ACTA IMEKO
ISSN: 2221-870X
September 2015, Volume 4, Number 3, 30 - 35

Hybrid time synchronization for Underwater Sensor Networks

Oriol Pallarés[1], Pierre-Jean Bouvet[2], Joaquín del Río[1]

[1] *SARTI Research group[Electronics Dept.]. Universitat Politècnica de Catalunya (UPC). Rambla de l'Exposició 24, 08800, Vilanova i la Geltrú, Barcelona, Spain*
[2] *Underwater Acoustics Lab ISEN Brest, C.S. 42807, 29228 Brest cedex 2 – France*

ABSTRACT
Time synchronization is a critical part of distributed applications over a sensor network. In this work we investigate time synchronization problems over UWSN deployed in shallow water, taking into account all main communication challenges of the water channel and observing its behavior in simulation and real tests. A hybrid frame was proposed based on time synchronization using both LFM and OFDM communication with channel impulse response equalization. Simulation results show how Hybrid synchronization outperforms existing synchronization protocols and how these results are affected in real underwater tests.

Section: RESEARCH PAPER

Keywords: UWSN; time synchronization; OFDM; acoustic communication

Citation: Oriol Pallarés, Pierre-Jean Bouvet, Joaquín del Río, Hybrid time synchronization for Underwater Sensor Networks, Acta IMEKO, vol. 4, no. 3, article 5, September 2015, identifier: IMEKO-ACTA-04 (2015)-03-05

Editor: Paolo Carbone, University of Perugia, Italy

Received February 14, 2015; **In final form** April 23, 2015; **Published** September 2015

Funding: This work is supported by the projects NeXOS and FixO3 from the European Union's Seventh Programme under grants agreement No 614102 and No 312463 respectively.

Corresponding author: Oriol Pallarés, e-mail: oriol.pallares@upc.edu

1. INTRODUCTION

Underwater acoustics has recently drawn attention from industry and research in the way that many processes can be interconnected and share valuable information. In order to perform this information exchange and create collaborative tasks, time synchronization is one of the main fields to study in Underwater Sensor Networks (UWSN) for distributed applications such as target tracking, data acquisition systems or acoustic beam-forming.

Since a GPS signal cannot reach below 1 meter of water column due to the high attenuation of the electromagnetic waves over the water channel, it is necessary to provide time synchronization to sensors through acoustic communication.

Many cabled and terrestrial wireless synchronization protocols are capable of providing high precision time offset estimation. They are widely used in many systems which need high precision time synchronization and do not have access to a GPS reference. The best known protocol and most common in our daily lives is the Network Time Protocol (NTP), which synchronizes our computers over Ethernet. Another cabled synchronization system that is used in industrial applications, majorly due to its high synchronization precision is the Precision Time Protocol (PTP) IEEE 1588 standard. This protocol has been ported by [1] to a terrestrial wireless network with timing accuracy below a few microseconds. Although these synchronization protocols can provide high timing accuracy in terrestrial networks such as Ethernet and WSNs these protocols cannot be used for the UWSN because they do not face some of the main problems of the underwater channel [2].

Actually, there are few protocols capable of providing time correction to underwater sensors [3], [4], but most of them do not take into account all the underwater physical layer communication challenges such as Doppler Effect correction, drift between clocks or propagation time variability, which become significant with high latency communication caused by low celerity of sound in water channel.

In this work, we present an hybrid synchronization system mixing Linear Frequency Modulation (LFM) for precise time arrival detection and Orthogonal Frequency Division Multiplexing (OFDM) communication with Channel Impulse

Response (CIR) equalization in order to correct drift, Doppler velocity and detect propagation time variations between message exchanges.

This hybrid synchronization yields similar simulation results to [5], but in our case, the study of synchronization is carried out both over simulation and real water experiments whereas results of [5] are limited to simulation. This result will be compared to the ones given by Time Synchronization for High Latency acoustic networks (TSHL) and Timing-sync Protocol for Sensor Networks (TPSN) underwater synchronization protocols which are the ones most used in actual underwater networks.

The paper will be divided in four main sections. First, in Section 2 the challenges faced in UWSN synchronization are presented. Before describing the model design, two of the main underwater synchronization systems and their weaknesses are presented in subsection 2.4. Then, Section 3 presents the system model used to perform simulation. The water tests for verifying time error accuracy of the system, is detailed in Section 4. Conclusions and Future works are discussed in Section 5.

2. PHYSICAL LAYER CHALLENGES IN HIGH LATENCY NETWORKS

When simulating underwater communications, cabled synchronization algorithms techniques can easily be applied, but in underwater real tests, conditions do not fit exactly simulation model, these non-idealities issues affect system performance and rigorous study of such effects is required in order to achieve synchronization errors below 100 microseconds.

At the starting point, we took the synchronization algorithm basic message exchange, represented in Figure 1, and modified this protocol as we went deeper in this section.

This synchronization message exchange base, is used in most cabled and wireless message exchange based protocols.

In order to compute offset between nodes A and B, four time stamps were necessary (*T1*, *T2*, *T3* and *T4*). These timestamps provide an estimate of the propagation time between nodes, leading to a precise synchronization in case of ideal propagation channel as is described in equation (1) for the offset β and equation (2) for the propagation time τ possible. *T2* and *T3* are enclosed in the message in order to make them available at the slave side (node A):

$$\beta = \frac{(T2-T1)-(T4-T3)}{2} \tag{1}$$

$$\tau = \frac{(T2-T1)+(T4-T3)}{2}, \tag{2}$$

where propagation time τ is computed as the average of propagation time from A to B (denoted τ_1) and propagation time from B to A (denoted τ_2). This algorithm makes the assumption that both times are equal to τ.

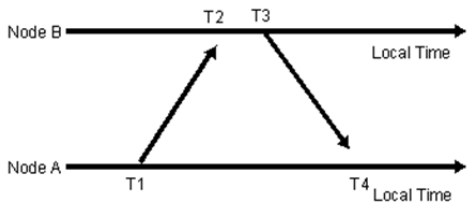

Figure 1. Sender-Receiver synchronization workflow.

We also observed the dependence between time stamping precision and offset estimation. The way how this assumption is affected by non-idealities conditions is described in the following.

First, subsection 2.1 studies the error of each time stamp for the incoming and outgoing messages and how this affects the main algorithm. The time stamping issue is handled also in cabled synchronization algorithms, so we will discuss how this uncertainty is solved for these networks and how it can be applied to UWSN. Then in Sections 2.2 and 2.3 the main problems added by the water channel are introduced, which may be negligible in cabled networks but not in high latency ones such as underwater sensor grids.

2.1. Time stamping

Considering a non-deterministic time stamping for the message input or output, equations (1) and (2) are reformulated as follows:

$$\beta(\varepsilon) = \frac{[(T2+\varepsilon2)-(T1+\varepsilon1)]}{2} - \frac{[(T4+\varepsilon4)-(T3+\varepsilon3)]}{2} \tag{3}$$

$$\tau(\varepsilon) = \frac{[(T2+\varepsilon2)-(T1+\varepsilon1)]}{2} + \frac{[(T4+\varepsilon4)-(T3+\varepsilon3)]}{2} \tag{4}$$

where $\varepsilon1$, $\varepsilon2$, $\varepsilon3$ and $\varepsilon4$ stand for the time stamp uncertainty in both incoming and outgoing messages. This uncertainty used to be around hundreds of microseconds for software time stamps and some nanoseconds for hardware time stamps such as the one presented in PTP. Hardware time stamping has higher precision because the messages are referenced to a time base in the Medium Access Controller (MAC), avoiding in this way unpredictability introduced by Operating System (OS) or medium access algorithms, which can be about hundreds of microseconds.

Hence, it is necessary to port the mechanisms used in cabled time synchronization algorithms for hardware time stamping to the underwater communication algorithms in order to get a similar performance.

A common workaround for solving time stamping ambiguity is to use a preamble before each frame, which can be recognized by the MAC, making possible the time reference acquisition by hardware before any processing of the frame.

2.2. Propagation time variation

Regardless of time stamping issue, propagation time variations between τ_1 and τ_2 will be noticed. This alteration may be due to different communication paths in cabled communications.

Analogously, it happens with a multi-path channel such as the underwater one, with the further difficulty added by moving nodes in a communication with high latency, which increases propagation time variation ($\Delta\tau = \tau_1 - \tau_2$).

Developing (1) and (2) with $\tau1 \neq \tau2$ leads to:

$$\beta(\Delta\tau) = \beta_{ideal} - \frac{\Delta\tau}{2} \tag{5}$$

$$\tau(\Delta\tau) = \tau_1 + \frac{\Delta\tau}{2}, \tag{6}$$

where the real clock offset β_{ideal} is estimated by β with an error equal to $\frac{\Delta\tau}{2}$.

2.3. Clock drift and Doppler effect

Besides clock offset between two nodes, there is another timing factor that must be taken into account when synchronizing two clocks. This is the drift between clocks,

which is the variation between a clock frequency compared to an ideal one, given in parts per million.

This factor gets importance as the network has bigger latency. The reason of this assumption lies in the amount of time that the protocol needs to compute the offset. While synchronization offset is computed, one clock is drifting. Therefore, by the end of the computation, we will have the initial offset to be compensated, plus the drift of this second clock. In a small latency network, time offset estimation is computed fast enough for neglecting the error introduced by the drift, as happens in the NTP protocol. But in high latency networks, a few seconds of propagation time will result in an error caused by drift equal to hundreds of microseconds at the end of the offset estimation protocol.

In literature, most of the synchronization protocols manage the time stamping issue, some of them also offer solutions for propagation time variations, but it is quite difficult to find studies of the drift compensation if we do not go deep in high precision synchronization protocols, such as IEEE-1588 (PTP) or in underwater networks TSMU scheme and TSHL protocol.

The main issue of estimating clock drift in non-cabled networks resides in the mobility of the nodes. When a node has a velocity relative to the medium it is affected by the Doppler Effect, consequently, it suffers from sampling base time variations. As described in [6], this sampling time variation is also produced by drift, which increases the difficulty to estimate drift and Doppler velocity separately.

Equations (7) and (8) describe how β and τ are modified, introducing clock drift (Θ) affecting only one of the clocks:

$$\beta(\Theta) = \frac{[T2-(T1(1+\Theta)]-[T4(1+\Theta)-T3]}{2} \quad (7)$$

$$\tau(\Theta) = \frac{[T2-(T1(1+\Theta)]+[T4(1+\Theta)-T3]}{2}. \quad (8)$$

Table 1 presents a description of all symbols used for the algorithm development.

2.4. Related work

Despite the growing interest, during the last years, in deploying UWSNs, there is only a few literature highlighting underwater wireless high precision synchronization algorithms.

TSHL is designed for high latency acoustic networks. It splits time synchronization in two phases. In the first phase, it estimates the clock drift based on a centralized timebase, and in the second phase, it determines the offset by a message exchange protocol. In order to perform clock drift estimation, a linear regression is computed over several beacon messages sent within a known time. The time deviation of the message arrival at the receiver side is assumed to be caused by clock drift. The main problem of this assumption is that, instead of having all the nodes completely quiet, multi-path and water currents will affect the message propagation time between nodes, causing a wrong measurement of clock drift.

Table 1. Notation Summary.

Symbol	Description
β	Estimated clock offset
β_{ideal}	Real clock offset
$\tau1, \tau2$	Propagation delay of A-B and B-A
$\Delta\tau$	$\tau1 - \tau2$
$\varepsilon1, \varepsilon2, \varepsilon3, \varepsilon4$	Time stamp uncertainty
Θ	Doppler scaling
Δtp	Bidirectional propagation time
t_0	Initial time
t_{wait}	User defined time between messages

Both PTP and TSMU protocols estimate drift jointly with Doppler effect, and then by analysing Doppler scaling, the method presented in an article from the Underwater Acoustics lab of Connecticut's University is capable of determining which part is due to Doppler velocity and which is due to clock drift, as described in [5]. In this way, they are capable of improving TSHL or TPSN performance by introducing Doppler movement to the offset and propagation time equations [6]. However, PTP is designed for cabled networks, so it cannot work directly with high propagation times, and TSMU is a scheme which had not been tested in real tests where the complexity of Doppler scaling estimation increases due to multi-path noise.

3. HYBRID TIME SYNCRHONIZATION MODEL

In this section we present an alternative time synchronization algorithm that can overcome the main underwater channel problems. For this first approach, Doppler scaling effects are not taken into account.

Thus, main issues discussed in this paper are the frame time stamping, the propagation time variations detection, and how to feedback these parameters to the system.

Output framing detection or time stamps, can be performed by most of the commercial DAQ systems, being capable to trigger the DAC output at a certain time with a HW pulse. This means that the output time stamping has a deterministic and negligible error.

Time stamping of an incoming message is performed by main communication algorithms in order to detect where the raw data is placed inside of an acquisition window. In this way, the frame can be sent to a DSP to demodulate the desired data.

Then, these algorithms can be used to detect the exact arrival time. This is done by performing a coarse estimation in order to window a 'Pilot' signal [7], and then analyse the Channel Impulse Response (CIR) of this Pilot for detecting the first sample which arrived, with 1 sample of error.
Consequently, this system will reach a maximum frame detection precision equal to 1/BW where BW denotes the signal bandwidth. As a result, larger BW will result in a better timing accuracy. This could lead us to increase BW to improve timing precision. However, an increase of BW results of a decrease of the robustness against communication channel due to extended Inter Symbol Interference (ISI). Hence, it is necessary to find a trade-off between this robustness and synchronization accuracy.

The workaround proposed in this paper is to use an LFM (chirp signal) in conjunction with CIR frame arrival detection. This will allow us to keep a low BW for communication and increase it in the LFM. Thus LFM is used to detect the reception of frames by performing correlation between the received signal and the expected chirp. Then, the computation of center of gravity improves the accuracy of the correlation peak leading to a precise time stamping. Finally, signal detection is performed, without the necessity of using a threshold, by using Schmidl & Cox algorithm [7] which provides in addition an estimation of the carrier frequency offset which modulates the main signal and the LFM.

We can send the global time stamp in the acoustic OFDM message and detect the frame arrival time with the LFM.

4. RESULTS

In this section, the results with Hybrid synchronization using LFM and CIR are discussed, in order to perform time stamping and estimate β and τ by adding both $\tau 1$ and $\tau 2$ without the indeterministic time errors given by the low BW time stamping procedure.

4.1. Workbench description

In order to perform real underwater tests, we used a water 'test tank' of dimensions 150 x 40 x 40 cm, with 1 meter separation between hydrophones. Since this study is not a matter of communication robustness but time synchronization and detection, we can extrapolate the results obtained in this test tank to shallow water sea communication in terms of timing accuracy.

In order to send data from a computer to the water tank, we used the National Instruments cRIO-9103 with an acquisition module (NI-9201) and a signal generator (NI-9263). This device consists of an FPGA and a real time controller, with the capacity to interconnect directly I/O modules to the FPGA, and control them in the LabVIEW programming environment. The main reason for using this system is the ease of programming the data Input/Output and function triggering. This module is in charge of sending and receiving information to the hydrophones, and to trigger clock readings each time a signal is generated or an acquisition starts.

In order to link the computer and the water channel, the cRIO controller implements a TCP/IP communication in order to communicate the data between the FPGA and the computer. The FPGA has a hardware description consisting of four different loops, two of them running a clock and the other two for acquisition and generation respectively. These loops are triggered pair wise, one trigger for the clock and the generation module and the second trigger for the other clock and acquisition device. This hardware description allowed us to implement the desired HW time stamp for synchronization.

When the signal is sent from a laptop to the cRIO it is saved in a memory register. Once the signal is completely received, an 'I/O' flag is set, causing a trigger in the FPGA generation gear together with a clock reading for the signal output time stamp. Simultaneously, the acquisition engine is triggered with its corresponding clock reading for an input signal time stamp. After a user defined acquisition time elapse, given by the maximum propagation time plus message duration, the 'I/O' flag is cleared. Then the controller sends back all the time stamps information to the laptop, plus the received acoustic raw data.

This hardware application makes it possible to perform a message exchange based synchronization protocol in a laptop, providing both time stamps, the generation and acquisition ones, besides information of the raw data enclosing a global time reference.

To connect the NI-9201 (ADC)/NI-9263 (DAC) and the hydrophones, a power and charge amplifier board has been designed for boosting both the generator power and the received signal. The power amplifier module consists of two operational amplifier stages in order to boost the current given by the NI-9263 generation module and at the same time adapt the impedance of the DAC to the hydrophone one. In this way, the amount of power loss before sending the signal to the water channel is reduced. Likewise, the charge amplifier consists of a two stage low noise amplifier with hydrophone adapted impedance, so that we sent an amplified signal with rejected noise to the NI-9201 acquisition module. The module used for real tests is represented in Figure 2.

We also used a T/R switch for a bidirectional communication using only one hydrophone. With this, we obtained a robust and convenient system to perform underwater acoustic communication.

This module was designed following the dimensions of a compact RIO module and using the same mounting to attach this system to the cRIO-9103. Figure 3 represents the final design used for underwater communication and synchronization as described above.

A block diagram of the communication workflow is presented in Figure 4.

For both simulations and real 'test tank' tests, a master clock and a slave clock were used. For the evaluation of the algorithms a slave clock generated from the master clock was used, in order to know the delay between clocks a priori.

As described in Figure 1, we defined in a Matlab® script T1 as base time (t_0), T2 is given by T1 plus the propagation time from slave to master (τ_1) plus an offset and drift between clocks. T3 equals T2 plus a user defined time between messages (t_{wait}), and finally T4 equals T1 plus both propagation times ($\Delta tp = \tau_1 + \tau_2$) and the user defined time between messages.

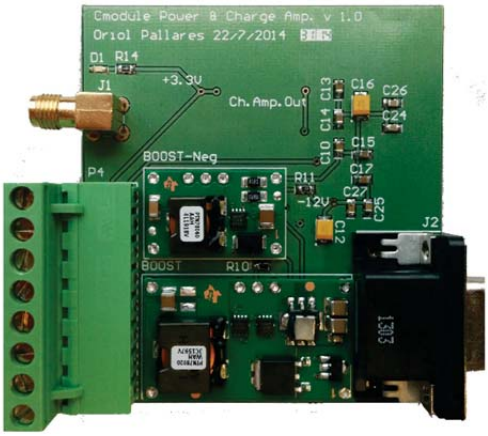

Figure 2. Power and charge amplifier front view.

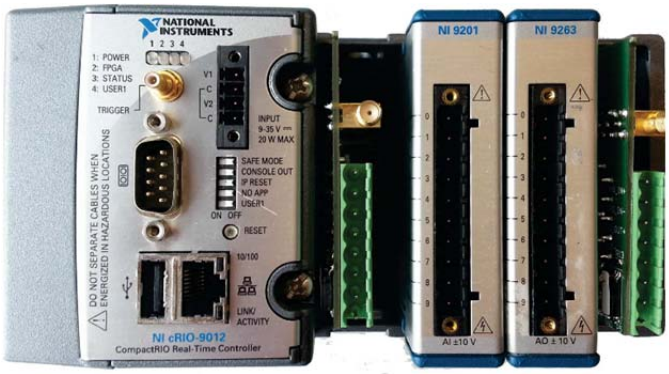

Figure 3. cRIO module plus power and charge amplifier board.

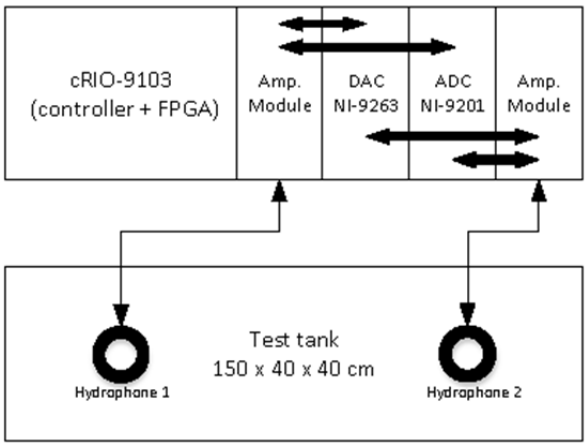

Figure 4. Frame detection workflow diagram for real test.

Equations from (9) to (12) describe how time stamps are defined in the script in order to simulate two different clocks.

$$T1 = t_0 + \varepsilon 1 \tag{9}$$

$$T2 = T1 + tp_1 + \beta_{ideal} + \theta_{ideal} + \varepsilon_2 \tag{10}$$

$$T3 = T2 + t_{wait} + \varepsilon_3 \tag{11}$$

$$T4 = T1 + \Delta tp + t_{wait} + \varepsilon_4 \tag{12}$$

With this definition we were able to simulate time stamp uncertainty, clock drift and offset in the script, and verify the proper detection of our algorithms for each time synchronization uncertainties.

4.2. Main results

Here, we present results given by simulations, which are compared to TPSN and TSHL, and also real tests demonstrating Hybrid-sync performance.

Comparisons with TPSN and TSHL were made by implementing a TPSN-like (TPSN-L) and TSHL-like (TSHL-L) protocol in Matlab® simulations. By this nomenclature, we mean that the simulation captured the essence of both protocols as described in [8]. However, for this work, an uncertainty due to propagation time variations was included, which leads this protocols to bigger uncertainty in synchronization precision.

Figure 5 shows the results of a simulation comparing TPSN-L, TSHL-L and Hybrid synchronization performance. For this test a propagation time variation with standard deviation of 15 parts per pillion (ppm) relative to the distance between nodes was used.

Due to the fact it is possible to estimate propagation times with higher precision, it is possible to compensate $\Delta t/2$, resulting in the Hybrid synchronization outperforming actual system time offset and delay estimation accuracy.

Then, in Figure 6 the results of an underwater test performed in the 'test tank', described in subsection 4.1 are depicted. This verifies the proper performance of the synchronization algorithm over SNR variations. Therefore, the mean estimation accuracy of the synchronization algorithm with an accuracy below 2 microseconds, and the variance, due to propagation time variations, below 100 microseconds for a low SNR, i.e. 5 dB, can be observed

5. CONCLUSIONS AND FUTURE WORK

In this article, we presented a new approach for a frame time stamp, which is a relevant part of frame based synchronization algorithms. Our study leads to a constant synchronization accuracy below 10 microseconds, independent of distance between nodes, by improving the frame time stamping for computing propagation times from master to slave nodes and vice-versa. So, we can conclude that our algorithm overcomes the studied synchronization algorithms found in literature.

Despite the achieved good results, we must take into account the assumption done during all our study of an absence of any Doppler Effect, neither in simulation nor real tests. In order to compare the Hybrid time synchronization accuracy versus the recent studies of time synchronization [9], Doppler scaling should be added to our tests.

Research is actually on-going, by the authors, to compare several Doppler scale estimation algorithm performance, in order to achieve the best performance in Hybrid Time synchronization algorithm.

ACKNOWLEDGEMENT

This work is supported by the projects NeXOS and FixO3 from the European Union's Seventh Programme under grants agreement No 614102 and No 312463 respectively.
The authors would also like to thank National Instruments Spain for their support providing us the necessary hardware used in this study.

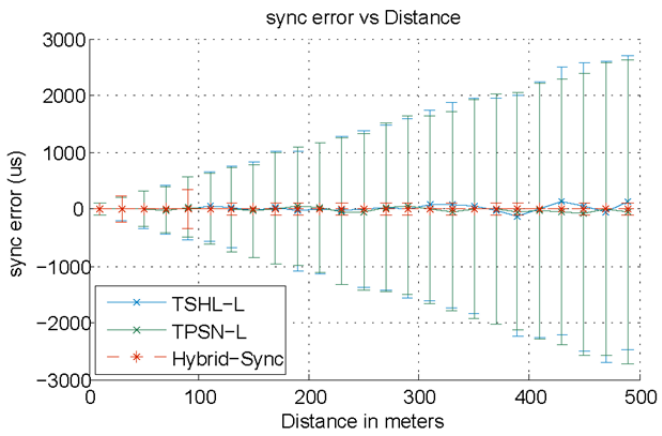

Figure 5. Quiet shallow water synchronization.

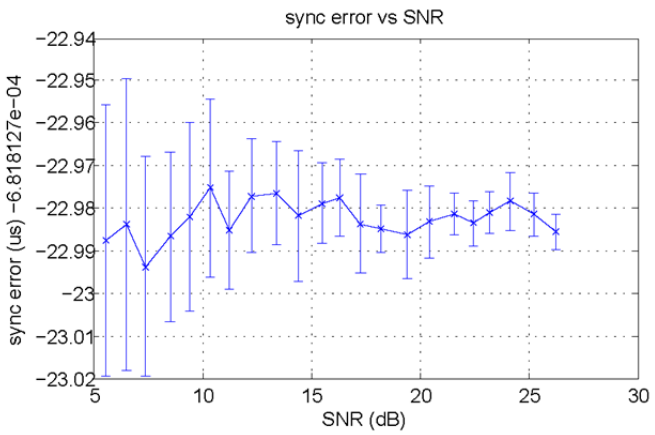

Figure 6. SNR vs synchronization performance.

REFERENCES

[1] Toma, D., Río Fernandez, J. D., Manuel Lázaro, A., Gomáriz Castro, S., & Carreras Pons, N., "Precision timing in TDMA - based Wireless Sensor Network through IEEE 1588 standard." 19th IMEKO TC-4, 2013 pp.683-686.

[2] Akyildiz, I. F., Pompili, D., & Melodia, T. "Challenges for efficient communication in underwater acoustic sensor networks." ACM Sigbed Review, 2004, 1(2), pp. 3-8.

[3] Chirdchoo, N., Soh, W. S., & Chua, K. C. "MU-Sync: a time synchronization protocol for underwater mobile networks." In Proceedings of the third ACM international workshop on Underwater Networks (pp. 35-42). ACM (2008, September).

[4] Lu, F., Mirza, D., & Schurgers, C. "D-sync: Doppler-based time synchronization for mobile underwater sensor networks." In Proceedings of the Fifth ACM International Workshop on UnderWater Networks (p. 3). ACM (2010, September).

[5] Liu, J., Wang, Z., Peng, Z., Zuba, M., Cui, J. H., & Zhou, S "TSMU: a time synchronization scheme for mobile underwater sensor networks." In GLOBECOM, pp. 1-6. IEEE, 2011.

[6] Stojanovic, M., Freitag, L. "Integrated doppler tracking and efficient resampling for phase coherent acoustic communication". Internet draft May 2007, available: http://www.mit.edu/~millitsa/resources/pdfs/joedop.pdf

[7] Schmidl, T.M., & Cox, D.C. "Robust frequency and timing synchronization for OFDM" Communications, IEEE Transactions on 45.12 (1997), pp. 1613-1621.

[8] Syed, A.A., & Heidemann, J.S. "Time Synchronization for High Latency Acoustic Networks." In INFOCOM (2006, April).

[9] Liu, J., Wang, Z., Zuba, M., Peng, Z., Cui, J. H., & Zhou, S. "DA-Sync: A Doppler-Assisted Time-Synchronization Scheme for Mobile Underwater Sensor Networks", IEEE Transactions on Mobile Computing 3 (2014), pp. 582-595.