SCIENCE AND TECHNOLOGY ORGANIZATION
CENTRE FOR MARITIME RESEARCH AND EXPERIMENTATION

**Reprint Series** **CMRE-PR-2019-111**

# A robust, opportunistic clock synchronization algorithm for ad hoc underwater acoustic networks

Arjan Vermeij, Andrea Munafò

June 2019

Originally published in:

IEEE Journal Of Oceanic Engineering, vol. 40, no. 4, October 2015, pp. 841-852, doi: 10.1109/JOE.2015.2469955

# About CMRE

The Centre for Maritime Research and Experimentation (CMRE) is a world-class NATO scientific research and experimentation facility located in La Spezia, Italy.

The CMRE was established by the North Atlantic Council on 1 July 2012 as part of the NATO Science & Technology Organization. The CMRE and its predecessors have served NATO for over 50 years as the SACLANT Anti-Submarine Warfare Centre, SACLANT Undersea Research Centre, NATO Undersea Research Centre (NURC) and now as part of the Science & Technology Organization.

CMRE conducts state-of-the-art scientific research and experimentation ranging from concept development to prototype demonstration in an operational environment and has produced leaders in ocean science, modelling and simulation, acoustics and other disciplines, as well as producing critical results and understanding that have been built into the operational concepts of NATO and the nations.

CMRE conducts hands-on scientific and engineering research for the direct benefit of its NATO Customers. It operates two research vessels that enable science and technology solutions to be explored and exploited at sea. The largest of these vessels, the NRV Alliance, is a global class vessel that is acoustically extremely quiet.

CMRE is a leading example of enabling nations to work more effectively and efficiently together by prioritizing national needs, focusing on research and technology challenges, both in and out of the maritime environment, through the collective Power of its world-class scientists, engineers, and specialized laboratories in collaboration with the many partners in and out of the scientific domain.



**NOTE:** The CMRE Reprint series reprints papers and articles published by CMRE authors in the open literature as an effort to widely disseminate CMRE products. Users are encouraged to cite the original article where possible.

# A Robust, Opportunistic Clock Synchronization Algorithm for *Ad Hoc* Underwater Acoustic Networks

Arjan Vermeij and Andrea Munafò, *Member, IEEE*

***Abstract*—Proliferation of deployed sea-going autonomous platforms, such as autonomous underwater vehicles (AUVs), unmanned surface vehicles (USV), and sensor nodes anchored to the seabed, make the deployment of true underwater acoustic networks more and more feasible. An important feature of any network is the ability to synchronize the clocks of the participants, for the purpose of, e.g., time-slotted media access control (MAC) and navigation. Terrestrial clock synchronization protocols, such as the well-established network time protocol (NTP), are not readily applicable to underwater acoustic networks, because of long propagation times, low packet delivery success rates, communication ranges that vary over time in an unpredictable manner, and, in the presence of mobile nodes, the *ad hoc* nature of the composition of the network. This paper proposes a continuous estimation of internode clock offset and drift, based on the continuous exchange of modem packets, possibly containing transmission and reception timestamps. The proposed solution takes explicitly into account the limitations of the acoustic communication channel and network node mobility. This robust, opportunistic clock synchronization (ROCS) is robust against modem reset, and will work even if packet delivery success rates are not optimal or if no communication is possible for extended periods of time. Experimental results are given from the COLLaborative Asw Behaviours–Next Generation Autonomous Systems (COLLAB-NGAS14) campaign, held October 19–31, 2014, off the west coast of Italy. During the sea trial, the proposed clock synchronization algorithm was deployed and successfully tested within an underwater acoustic network composed of mobile and fixed nodes.**

***Index Terms*—Acoustic communications, clock synchronization, localization and navigation, multiagent systems, optimization, underwater acoustic networks, underwater sensor network.**

## I. Introduction

THIS paper reports on the formulation and experimental validation of a clock synchronization algorithm for underwater acoustic networks. The goal of this work is that of providing accurate timing to underwater networks with a distributed algorithm based on opportunistic data, without relying on centralized master nodes, dedicated phases, or synchronized actions.

The availability of a shared time is critical for a number of applications ranging from implementation of media access control (MAC) layers in underwater acoustic networks (e.g., it allows the implementation of time-division multiple-access algorithms over multihop networks), to data fusion applications where the availability of a common notion of time allows to put together data coming from different sensors to form a unique result (e.g., bistatic sonar applications, where the receiver needs a precise estimate of the moment the source starts transmitting [1]), and to localization and navigation of autonomous underwater vehicles (AUVs), where synchronized clocks provide the ability to navigate through one-way travel time (OWTT), as opposed to two-way travel time (TWTT). Synchronous-clock acoustic navigation is described, among others, in [2], using high-stability oscillators. The availability of dedicated hardware with high-stability clocks has been so far the most used method in operational underwater sensor networking. Although effective, using dedicated hardware increases the cost and energy requirement of the nodes, inherently limiting the network scale, spatially, because of the few nodes, or temporally, because of the diminished node endurance. Moreover, increasing node requirements directly limits *ad hoc* and free association potential capabilities of the network.

Several clock synchronization protocols have been designed for terrestrial networks; Sundararaman *et al.* [3] give a very useful overview. Although these networks do share many of the problems of the underwater ones (e.g., unreliable packet delivery, limited communication ranges, etc.), clock synchronization protocols for terrestrial wireless networks cannot be easily transposed to the underwater domain, as they rely on assumptions (e.g., negligible signal propagation time) that are not valid underwater or they induce a communication overhead that is not sustainable in acoustic-based communications. Terrestrial clock synchronization protocols, like the network time protocol (NTP), are not readily applicable to underwater acoustic networks, because of long propagation times, low packet delivery success rates, communication ranges that vary over time in an unpredictable manner, and, in the presence of mobile nodes, the *ad hoc* nature of the composition of the network.

In contrast, the algorithm presented herein aims at taking explicitly into account the peculiarities of the underwater acoustic application scenario. The acoustic network considered in this work can be composed of both fixed and mobile nodes that can be far apart from each other, where the maximum communication range between two deployed modems is 7.5 km. Nodes may come into and out of communication range at irregular intervals

Manuscript received February 06, 2015; revised May 07, 2015 and July 31, 2015; accepted August 11, 2015. Date of publication September 18, 2015; date of current version October 09, 2015. This work was supported by the NATO Allied Command Transformation (ACT), Collaborative Anti-Submarine Warfare Programme, Project 1 and Project 2. This work was presented in part at the 2014 Conference on Underwater Communications and Networking (UComms), Sestri Levante, Italy, Sep. 2014.

**Associate Editor: M. Chitre**.

The authors are with the NATO STO Centre for Maritime Research and Experimentation (CMRE), La Spezia 19126, Italy (e-mail: arjan.vermeij@cmre.nato.int; andrea.munafo@cmre.nato.int).

Digital Object Identifier 10.1109/JOE.2015.2469955

due to changes in the acoustic channel, and/or node movement. Some of the nodes might be persistently part of the network, while others may be taken out (e.g., for maintenance) and put back in the water only after a prolonged period of time. Nodes that are taken out for maintenance are often power cycled, which gives rise to a modem (and therefore clock) reset.

The resulting algorithm is able to continuously estimate internode clock offset and drift, based on the opportunistic exchange of modem packets containing transmission and reception timestamps. It operates at application level, exploiting timing features made available by the physical layer (i.e., availability of precise transmission and reception timestamps). The benefit of the approach is that the algorithm can be easily implemented within an underwater sensor network where packet delivery success rates are not optimal and/or no communication is possible for extended periods of time. The approach can be tuned to the requirements of the application, optionally reducing the synchronization communication overhead when more important messages must go through the network. Finally, the algorithm is able to detect and respond to modem resets, and hence is robust against asynchronous events (e.g., interruptions due to maintenance, failures, etc.) affecting the nodes and hence their internal clocks.

The remainder of this paper is organized as follows. Section II describes some of the major design challenges for clock synchronization protocols in terrestrial and underwater wireless sensor networks. Section III describes previous work in this area of research, with particular focus on clock synchronization algorithms that have been designed for underwater acoustic networks. Section IV describes the mathematical framework for the proposed clock synchronization algorithm. Section V goes into the details describing online implementation. Section VI presents the results from the COLLaborative Asw Behaviours–Next Generation Autonomous Systems (COLLAB-NGAS14) sea trial. Postprocessing results are given in Section VII-A, while those obtained online are in Section VII-B. Discussions and conclusions are given in Section VIII.

## II. Design Challenges

Terrestrial wireless sensor networks share many of the problems of the underwater domain, like unreliable packet delivery, intermittent connectivity, and limited communication ranges. A useful survey of clock synchronization in wireless networks is presented in [3]. The survey points out that as wireless sensor networks are typically closely tied to the application, different clock synchronization protocols may be used in different applications. The survey classifies clock synchronization protocols based on two kinds of features: synchronization issues and application-dependent features. Synchronization issues refer to the core challenges of a clock synchronization protocol that are independent of the deployment environment, such as architecture (e.g., peer-to-peer versus master–slave), synchronization action (clock correction versus foreign clock modeling), and certainty measures (probabilistic versus deterministic synchronization). Application-dependent challenges include node mobility (and the resulting varying propagation delay and varying network connectivity) and underlying timing mechanism.

In addition to these clock-synchronization-specific challenges, the underwater acoustic communications environment has its own peculiarities:

- the bandwidth is limited and distance dependent;
- the packet propagation times are long due to low speed of sound;
- the low speed of sound, coupled with a relatively (in a proportional sense to other domains) high spatial variation in sound speed, induces complicated ray bending and multipath effects;
- node movement during packet exchanges influences propagation times significantly, due to the low propagation speed compared to synchronization accuracy;
- node movement (i.e., underwater navigation and localization) is difficult to estimate onboard where a node may not have advanced hardware for such purpose;
- the feature sets offered by readily available off-the-shelf acoustic modems is disparate, making reliance on "specialized" features such as synchronous message transmission, as in [4], difficult for interoperability and portability.

As highlighted in [5], together all these constraints make the acoustic channel an extremely difficult communication medium, with poor quality and high latency, combining the worst attributes of terrestrial mobile and satellite radio channels.

### A. Application Scenario

The typical operating scenario where robust, opportunistic clock synchronization (ROCS) is targeted at addressing includes a heterogeneous network of underwater nodes, typically with network size $\approx$10, with internode spacing as large as 8 km. The nodes are heterogeneous in their mobility (stationary and mobile nodes), their surface expression (some have GPS and/or WiFi connections), and their self-localization capabilities (not all have onboard navigation in all deployments). Currently, all nodes share a similar modem type, to allow node-to-node communications, though there are parallel efforts in providing interoperable solutions for manufacturer-disparate hardware [6]. Additionally, in experimental deployments, network nodes may experience periods of zero connectivity (due to the combination of spacing and environmental conditions) or power cycling events (during hardware interventions). It is through the lens of these operating parameters that ROCS is conceived.

### B. Goals

ROCS is to be used for synchronizing, in real time, clocks between different underwater assets, for purposes of augmenting navigation capabilities and providing live data synchronization. The primary purpose of ROCS is to provide sufficient synchronization such that relative ranging can be performed between nodes using OWTT measurements, an important feature in scenarios where traditional navigation aides (such as IMU and DVL) do not perform well alone. Additionally, data synchronization is advantageous when data being transmitted across the network must be correlated between disparate nodes, during collaborative missions.

At a lower level, for the sake of portability and practical adoption, it is beneficial for a clock synchronization algorithm to rely

on a minimum feature set of the target hardware. For this reason, ROCS relies only on asynchronous timestamping of receptions and transmissions; in the presented implementation, this occurs at the physical layer. Importantly, because of the highly variant liveliness of nodes (with regards to power cycles and connectivity), the presented solution focuses on being distributed, opportunistic, and continuous.

1) Distributed: In the typical application scenario there is neither a requirement nor a distinction of the "true" time. What is most important is determining the relation between any given pair of nodes. For this reason, a true clock correction (i.e., hard set) is not required, only the modeling of the difference between any given node pair.
2) Opportunistic: The acoustic network bandwidth is typically fully utilized, so additional overhead required for a clock synchronization algorithm should minimally impact the core network functions and users. So, rigid synchronization cycles or phases are an impractical approach. This additionally decouples MAC operation from clock synchronization.
3) Continuous: Acoustic connectivity is highly variant, and nodes may be inserted or removed from the network at any time. In this application, again, initialization phases or distinct startup cycles are impractical.

It is on these high-level goals and concepts that ROCS is founded.

## III. Previous Work

The majority of prior literature in the field of clock synchronization for underwater networks reports on the results from numerical simulations of presented algorithms, often without considering some of the major challenges present in the marine environment. Few papers consider high probability of packet loss, and the unreliability of the acoustic channel, and even fewer papers report on deployed experimental evaluation of their approaches. In the following, we summarize some of the most relevant references. Except where noted, all approaches use a series of two-way packet exchanges, followed by linear regression to estimate the clock offset and drift.

Time synchronization for high latency (TSHL) [7], which is, to the best of the authors' knowledge, the first approach to underwater clock synchronization, is a clock synchronization algorithm for short-range high-latency acoustic networks. It is the only method found that is not based on a sequence of two-way packet exchanges. TSHL employs two phases: during phase 1, a clock drift is calculated by performing linear regression over multiple beacon packets; and during phase 2, a clock offset is calculated by a single two-way packet exchange with MAC-level timestamping. TSHL assumes a constant propagation time, and therefore nodes must be stationary. The paper presents simulation results for short ($< 500$ m) ranges.

MU-Sync [8] aims to calculate the clock offset and drift, also using a two-phase approach. In the first phase, called the skew and offset acquisition phase, the clock offset and drift (skew) are estimated by applying linear regression twice over a set of $n$ reference modem packets (beacons), transmitted by the so-called cluster head, and replied to by all neighboring nodes. In the second phase, the synchronization phase, the cluster head broadcasts the clock offset and drift of all the neighboring nodes. The MU-Sync approach is, through the requirement of the presence of a cluster head, essentially a master–slave approach.

Synchronized-time–synchronized-location (STSL) algorithm [9] is a sequential algorithm for clock synchronization and localization. Each to-be-localized node is within direct communication range of a number of anchor nodes. An anchor node has an accurate position sensor. STSL assumes that nodes are equipped with a dead-reckoning navigation system. The quality of the dead-reckoning navigation system determines the length of the localization window. STSL relies on periodic packet exchanges during the localization window. Simulation results are presented, as well as results from a sea trial. The effect of packet loss on the performance of STSL is not considered.

Light-Sync [10] assumes that the acoustic network is stationary, consisting of a number of nodes anchored at the seabed, and one stationary gateway buoy on the surface which has accurate timing information, e.g., through a GPS receiver. Light-Sync is primarily aimed at reducing power consumption. It employs two phases. The first phase involves a series of two-way modem packet exchanges between the gateway buoy $P$ and a bottom node $A$, initiated by $A$. The time between reception and transmission $\epsilon$ on $P$ is a fixed quantity known to $A$. Modem packet transmission timestamps are encoded in the packets themselves. Phase 1 ends when node $A$ has estimated its own clock parameters with respect to the clock of node $P$ with sufficient accuracy. At the end of phase 1, node $A$ broadcasts its clock parameters. Some other node $B$, which is within communication range of node $A$ but not node $P$, can now estimate its own clock parameters.

Mobi-Sync [11] considers networks that consist of three different node types: surface buoys, equipped with GPS; supernodes, working as reference clocks, as they are always clock synchronized with surface buoys; and inexpensive ordinary nodes of low complexity. Mobi-Sync uses the spatial correlation between mobile nodes to estimate the propagation time, using a model to estimate an ordinary node's speed as a function of the speed of a set of supernodes. During the packet exchange phase, an ordinary node broadcasts a clock synchronization request message. Each neighboring supernode schedules two response packets, containing the supernode's recorded velocity vector and the packet's MAC layer timestamp. The packet exchange phase ends when the ordinary node has gathered sufficient data points for linear regression with a preset accuracy. After a fixed interval $t_{r1}$, each supernode transmits its first response packet, relying on the fact that each supernode is at a different distance from the ordinary node to exclude collisions. After a fixed interval $t_{r2}$, the supernodes transmit their second response packet. Mobi-Sync does not take into account packet loss, nor receiver node mobility while a packet travels through water. The assumption of the existence of an internode speed correlation model breaks down for self-propelled nodes. The proposed values for $t_{r1} = 2$ ms and $t_{r2} = 6$ ms are incompatible with the modem packet length of $\approx 1$ s typically found when communicating over larger distances with lower frequency acoustic modems.

In D-Sync [12], a cluster head initiates the synchronization process by broadcasting a reference packet, to which all neighboring nodes reply. D-Sync uses Doppler measurements to estimate the relative speed between the cluster head and its neighboring nodes, thus reducing the error in the estimation of the clock offset and drift. D-Sync is, like MU-Sync, essentially a master–slave approach. Lu *et al.* [12] also describe broadcast D-Sync, which aims to combine the accuracy of D-Sync with the energy efficiency of TSHL. To achieve this, broadcast D-Sync employs an even more complicated master–slave approach, which makes it very sensitive to packet loss. Lu *et al.* show simulation results that demonstrate D-Sync's superior accuracy when compared to MU-Sync.

DA-Sync [13], like D-Sync, uses Doppler velocity measurements but employs a Kalman filter to refine the velocity estimation, assuming a kinematic model for the relative motion between nodes. Liu *et al.* show that, in simulation, DA-Sync achieves some marginal improvements over D-Sync.

JSL [14] is a joint clock synchronization and localization algorithm. It takes into account the sound-speed profile to mitigate the effect on range estimates that sound waves have if not traveling in a straight line, which is relevant in scenarios with a significant difference in depth between nodes.

RSUN [15] employs a reference-node initiated two-way packet exchange with linear regression. Khandoker *et al.* recognize that packet collisions and subsequent retries decrease the measurement accuracy, caused by node mobility, and propose the use of Cook's distance to eliminate outliers. Khandoker *et al.* do not describe why this would be better than simply discarding all two-way packet exchanges with a duration exceeding some threshold.

While the work reported in this paper is based on similar ideas, it differs in some key aspects from the previous work in this field. Primarily, ROCS does not rely on fixed phases or cycles of synchronization, instead continuously estimating internode clock differences via the opportunistic transmission of timing information. These timing data are piggybacked on an established network traffic, with no feedback on MAC or physical-layer timing. ROCS accepts timing information from lower layers, but itself introduces no new timing requirements to those layers. Additionally, ROCS is highly distributed, with no sense of hierarchy or "stratum"—nodes simply estimate the errors between their own clocks and that of any foreign node. Last, ROCS has been tested in real-world, online sea trials.

## IV. Problem Setting

To calculate the offset between two modem clocks, a few modem features are assumed. First, each modem has a clock with a near-constant drift. Second, each modem is capable of accurate registration of send-and-receive timestamps. Third, each modem is capable of accurate determination of the speed relative to the transmitting modem, typically through the Doppler shift of incoming packets [16]. At first glance, this last feature may seem to be an excessive demand on the modem, but as the modem cannot decode incoming packets without the accurate determination of the Doppler shift, this feature comes at a negligible cost.

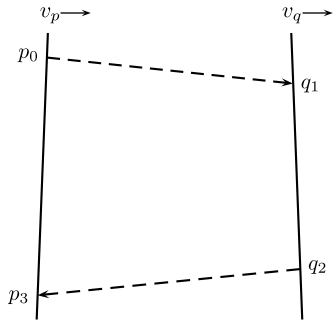

Fig. 1. Two-way packet exchange.

At this point, it is important to realize that a node can have more than one clock. Depending on the implementation, the application, transport, network, MAC, and physical layer can all have their own clock. This paper assumes that the timestamps reported by the modem are physical-layer timestamps. In terrestrial wireless sensor networks, at the transmitter node, the delay due to packet transport from the application layer, down the communications stack, to the physical layer, causes errors in clock synchronization. At the receiver node, the packet transport from the physical layer, up the communications stack, to the application layer, introduces additional errors. The use of modem-generated physical-layer timestamps eliminates these sources of errors [4].

Most clocks have a nonconstant drift that varies mainly as a function of temperature [17]. However, our experience has shown little to no practical effect of changes in water temperature on the clock drift of the nodes currently employed, perhaps due to the location of the electronics, which are housed in a closed cylinder with an apparent constant internal temperature. This has been checked experimentally by comparing the modem clock drift over a typical mission deployment with a baseline low-drift high-accuracy clock.

Let $t_m$ be the time reported by modem $m$'s clock. At time $t_{p,0}$, or $p_0$ for short, modem $p$ sends a packet to modem $q$, which receives it some time later, at time $t_{q,1}$, or $q_1$. At $t_{q,2}$ (or $q_2$) modem $q$ responds with a packet of its own, which modem $p$ receives at time $t_{p,3}$ (or $p_3$). This sequence is referred to as one two-way packet exchange (see Fig. 1).

This paper focuses on nodes that are relatively far apart from each other. The maximum range is assumed to be 7.5 km, a common value for commercially available low-frequency acoustic modems. For a typical value of the speed of sound in water, $c = 1500\ \mathrm{m} \cdot \mathrm{s}^{-1}$, this maximum range corresponds to a one-way travel time of 5 s. Thus, the maximum two-way packet exchange time cannot be less than 10 s. Requiring node $q$ to respond immediately will likely conflict with other constraints imposed on underwater communication. Therefore, we allow for some delay in response, and we assume the maximum two-way packet exchange time to be about 1 min. If the two-way packet exchange time is much larger than that, certain second-order effects can no longer be ignored. The nodes are allowed to travel through water with moderate speed, around a few $\mathrm{m} \cdot \mathrm{s}^{-1}$. At larger speeds, certain second-order effects can no longer be ignored.

Modem $m$ is attached to node $m$, which is assumed to have some computing capability. The modem reports the timestamp

of each packet that is sent or received, to its node, in the local time $t_m$. After a two-way packet exchange, node $p$ knows timestamps $p_0$ and $p_3$, and node $q$ knows timestamps $q_1$ and $q_2$. Node $q$ may then send timestamps $q_1$ and $q_2$ to node $p$ in the payload of a later packet.

The problem is then to derive, on node $p$, a function $f_{pq}$ that maps timestamps reported by modem $q$ to modem $p$'s clock

$$t_p = f_{pq}(t_q). \tag{1}$$

### A. Modem Clock Model

In this paper, modem time $t_m$ is modeled as a linear function with an offset $\Delta_m$ and a constant drift $\delta_m$, relative to the true and absolute time $t$

$$t = (1+\delta_m)t_m + \Delta_m. \tag{2}$$

Therefore, the times of modem $p$ and $q$ can be related as

$$(1+\delta_p)t_p + \Delta_p = (1+\delta_q)t_q + \Delta_q \tag{3}$$

and (1) can be rewritten as

$$t_p = (1+\delta_{pq})t_q + \Delta_{pq} \tag{4}$$

where

$$1+\delta_{pq} = \frac{1+\delta_q}{1+\delta_p} \quad \text{and} \quad \Delta_{pq} = \frac{\Delta_q - \Delta_p}{1+\delta_p}. \tag{5}$$

The problem of finding $f_{pq}$ is thus reduced to estimating the offset of modem $p$'s clock relative to the offset of modem $q$'s clock $\Delta_{pq}$, or $\Delta$ for short, and the relative drift between both modems $\delta_{pq}$, or $\delta$ for short.

### B. Estimating Clock Offset and Drift

This section is concerned with estimating the offset between two modem clocks using data gathered from a single two-way packet exchange, and the drift using data gathered from a sequence of two-way packet exchanges. To simplify the analysis, the following assumptions are made.

- Sound speed is considered to be known and fixed for the duration of a deployment, as the variation (spatially and temporally) in sound speed over deployment time scales tends to be small with respect to the synchronization accuracy. This implies that the sound speed is also considered to be constant during a two-way packet exchange. This relieves the constraint of nodes being required to be able to measure sound speed directly. (See a sample of trial sound speed data in Section VI.)
- For the moving nodes, heading and speed are assumed to be constant for the duration of a two-way packet exchange.

Since both nodes might be moving, we choose a 1-D coordinate system describing the position $x_{q,i}$ of node $q$ with respect to node $p$ at $t_{p,i}$, with origin at $x_{p,0} = 0$, positive in the direction of node $q$. Recall that $i$ in $t_{m,i}$ indexes one of the discrete sequences in the two-way packet exchange, as in Fig. 1, where $i \in \{0,1,2,3\}$. The range $r_{q,i}$ between nodes $p$ and $q$ at $t_{p,i}$ is $x_{q,i} - x_{p,i}$. The relevant part of the velocity of moving nodes is the range-wise velocity, i.e., the first derivative of $x$. Let $\dot{x}_{m,i} = v_{m,i}$ be the range-wise velocity of node $m$ at $t_{p,i}$. Based on the Doppler effect that the node calculates and reports, the range-wise velocity differential between nodes $p$ and $q$ at $t_{p,i}$ equals $\dot{r}_{pq,i} = \dot{r} = v_{q,i} - v_{p,i}$. Finally, let $c$ be the average speed of sound in water, relative to modem $p$'s clock, during this exchange.

Then, assuming that $v_p$ and $c$ are constant for the duration of one two-way packet exchange, the outbound packet travel time $t_{p,1} - t_{p,0}$, or $p_1 - p_0$ for short, can be calculated as

$$p_1 - p_0 = \frac{x_{q,1} - x_{p,0}}{c} = \frac{x_{q,1}}{c} = \frac{x_{q,0} + (p_1 - p_0)v_q}{c} \tag{6}$$

from which it follows that

$$p_1 = p_0 + \frac{x_{q,0}}{c - v_q}. \tag{7}$$

In the same way, the inbound packet travel time can be calculated as

$$\begin{aligned} p_3 - p_2 &= \frac{x_{q,2} - x_{p,3}}{c} \\ &= \frac{x_{q,0} + (p_2 - p_0)\dot{r} - (p_3 - p_2)v_p}{c}. \end{aligned} \tag{8}$$

The position of node $q$, $x_{q,0}$, can be rewritten as

$$x_{q,0} = \frac{c - v_q}{2c}\left((c+v_p)(p_3 - p_0) - (c+v_q)(1+\delta_{pq})(q_2 - q_1)\right). \tag{9}$$

With this the clock offset $\Delta_{pq,1} = q_1 - p_1$ can be formulated as

$$\begin{aligned} \Delta_{pq}(p_1) = &\frac{1}{2}\left(-(p_0 + p_3) + q_1 + q_2 + \delta_{pq}(q_2 - q_1)\right) \\ &+ \frac{\dot{r}}{2c}(1+\delta_{pq})(q_2 - q_1) \\ &- \frac{v_p}{2c}\left(p_3 - p_0 - (1+\delta_{pq})(q_2 - q_1)\right). \end{aligned} \tag{10}$$

With

$$\begin{aligned} \Delta_s &= \frac{1}{2}\left(-(p_0 + p_3) + q_1 + q_2 + \delta_{pq}(q_2 - q_1)\right) \\ t_{\overline{w}} &= (1+\delta_{pq})(q_2 - q_1) \\ t_w &= (p_3 - p_0 - (1+\delta_{pq})(q_2 - q_1)) \end{aligned}$$

equation (10) can be written as

$$\Delta_{pq,1} = \Delta_s + \frac{\dot{r}}{2c}t_{\overline{w}} - \frac{v_p}{2c}t_w \tag{11}$$

where $\Delta_s$ represents the clock offset for stationary nodes, $t_{\overline{w}}$ is the time between the arrival of the first packet and the transmission of the second packet of a two-way packet exchange, and $t_w$ is the time the packets of a two-way packet exchange travel through water. The longer $t_{\overline{w}}$, the larger is the impact of errors in $\dot{r}$, and the more important it is that $\dot{r}$ be constant during a two-way packet exchange. The longer $t_w$, the larger is the impact of the uncertainty in $v_p$.

From (11), it is possible to calculate the relative clock drift $\delta_{pq}$ using the least square method over a series of two-way packet exchanges to fit a linear regression model (constant clock drift assumption). Equation (11) can be rewritten as

$$\begin{aligned} &\frac{1}{2}\left[p_0 + p_3 + \frac{v_p}{c}(p_3 - p_0)\right] \\ &\quad = \left[q_1 + \left(1 + \frac{\dot{r} + v_p}{c}\right)\frac{q_2 - q_1}{2}\right](1+\delta_{pq}) + \Delta_{pq}. \end{aligned} \tag{12}$$

Equation (12) can be expressed as the linear matrix equation

$$\mathbf{B}\boldsymbol{\theta} = \mathbf{b} + \boldsymbol{\epsilon} \tag{13}$$

with $\boldsymbol{\theta} = [\boldsymbol{\theta}(1), \boldsymbol{\theta}(2)]^\top = [1 + \delta_{pq}, \Delta_{pq}]^\top$, where $\mathbf{B}$ is a two-column matrix with row $j$

$$\mathbf{B}_j = \left[ q_1 + \left(1 + \frac{\dot{r} + v_p}{c}\right) \frac{q_2 - q_1}{2},\ 1 \right] \tag{14}$$

$\mathbf{b}$ is a column vector with values

$$\mathbf{b}_j = \frac{1}{2}\left[ p_0 + p_3 + \frac{v_p}{c}(p_3 - p_0) \right] \tag{15}$$

and $\boldsymbol{\epsilon}$ is a column vector representing the error terms $\boldsymbol{\epsilon} = [\epsilon_\delta, \epsilon_\Delta]$. Each row in $\mathbf{B}$ and $\mathbf{b}$ describes a two-way packet exchange, where $p_0$, $q_1$, $q_2$, $p_3$, and $\dot{r}$ are the corresponding measurements of the $j$th two-way packet exchange. Applying the ordinary least squares estimator

$$\hat{\boldsymbol{\theta}} = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{b} \tag{16}$$

yields an estimate for the relative clock drift $\hat{\delta}_{pq}$ and the offset $\hat{\Delta}_{pq}$.

The unknown quantity in $\mathbf{B}$ and $\mathbf{b}$ is $v_p$, representing the range-wise velocity of node $p$. Let $V_p$ and $V_q$ be the maximum speed of node $p$ and $q$, respectively. Then, with $\dot{r} = v_q - v_p$ provided by the acoustic modem

$$v_{p,l} = \max\left(-V_q, -(V_p - \dot{r})\right) - \dot{r}$$

and

$$v_{p,u} = \min\left(V_q, (V_p + \dot{r})\right) - \dot{r} \tag{17}$$

where $v_{p,l}$ and $v_{p,u}$ are the lower and upper bounds for $v_p$, respectively.

Our solution is to use an iterative approach, with iterator $k$, where each iteration cycle gradually improves on $p$'s estimated range-wise velocity $\hat{v}_p$. For the first iteration, $v_p$ is set to the center of its interval

$$v_{p,k=0} = \frac{v_{p,l} + v_{p,u}}{2}. \tag{18}$$

For subsequent iterations of $k$, $v_p$ is set to the value in the interval that is closest to the value constrained by the linear regression model $\hat{v}_{p,k}$

$$v_{p,k>0} = \begin{cases} v_{p,l}, & \text{if } \hat{v}_{p,k-1} < v_{p,l} \\ v_{p,u}, & \text{if } \hat{v}_{p,k-1} > v_{p,u} \\ \hat{v}_{p,k-1}, & \text{otherwise} \end{cases} \tag{19}$$

where for a given iteration $k$

$$\begin{aligned} \hat{v}_{p,k} = {} & \frac{c}{p_3 - p_0 - (q_2 - q_1)(1 + \hat{\delta}_{pq,k})} \\ & \times \Bigg( 2\hat{\Delta}_{pq,k} - p_0 \\ & \quad - p_3 + (1 + \hat{\delta}_{pq,k}) \left[ q_1 + q_2 + \frac{\dot{r}}{c}(q_2 - q_1) \right] \Bigg) \end{aligned} \tag{20}$$

where the $k$th iteration of the least squares estimator $\hat{\boldsymbol{\theta}}_k = [1 + \hat{\delta}_{pq,k}, \hat{\Delta}_{pq,k}]^\top$. The iteration continues in this way, until the output of the linear regression model has converged to a stable value. The following pseudocode describes the process.

---

Construct matrices $\mathbf{B}$ and $\mathbf{b}$ using (18)
Calculate $\hat{\boldsymbol{\theta}}_{k=0}$ using (16)
**for** $k = 1, \ldots, K-1$
  Construct matrices $\mathbf{B}$ and $\mathbf{b}$ using (19) and (20)
  Calculate $\hat{\boldsymbol{\theta}}_k$ using (16)
  **break if** $|\hat{\delta}_k - \hat{\delta}_{k-1}| < \epsilon_\delta$ **and** $|\hat{\Delta}_k - \hat{\Delta}_{k-1}| < \epsilon_\Delta$
**end**

---

In this pseudocode, variable $K$ is the maximum number of iterations, and variables $\epsilon_\delta$ and $\epsilon_\Delta$ are the heuristically chosen values to make the algorithm terminate once $\hat{\delta}_{pq}$ and $\hat{\Delta}_{pq}$ have converged to a stable value.

## V. Clock Synchronization: Online Implementation

When the algorithm described in Section IV has to be implemented within an operational underwater acoustic network, some additional issues need to be addressed. The nodes, in general, do not have a way of knowing packet transmission times *a priori*, nor do the modems automatically encode the transmission or reception timestamps, at the physical level. Rather, each packet generates the necessary timestamps, and all these data are communicated in the payload of later packets. This is an implementation requirement that creates two issues. First, the application level at each node must be responsible for the encoding of the transmission and reception timestamps into dedicated messages. Second, given that each node receives a set of transmission and reception timestamps from others, it has to correctly associate them to form sequences of packet round-trips that feed into the described clock synchronization algorithm. This association procedure is complicated by the prevalence of packet loss. Note that, in some networks and using some specific modems, some of these advanced features (e.g., automatic timestamp encoding, or scheduled transmissions) might be available (e.g., [4]). However, the aim of this work is to be largely hardware and network independent. This section goes into details describing the implementation problems as they relate to the online implementation of the approach.

### A. Transmission of Timestamps

Whenever a node transmits or receives a modem packet, it registers the corresponding timestamp, a $p_0$ for a transmission, a $p_3$ for a reception (see Fig. 1). At regular intervals, each node encodes its modem's address and the previously registered timestamps into a message to be broadcast. For each reception timestamp, the corresponding source address is included. Given the limited bandwidth available in acoustic communications, some ideas described in [18] are put into practice, including some efficient encoding schemes. This allows the user to trade off the clock synchronization performance and the communication requirements. More specifically, the parameter "encoded size" is used to limit the number of bytes available for the encoding of timestamps. Not more than $N$ transmission timestamps are encoded, and the remaining space is filled with as many reception timestamps as will fit in, up to a maximum of $M$. Another user-

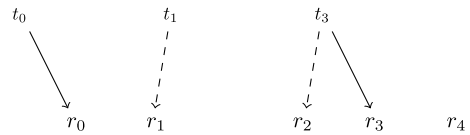


Fig. 2. Two partial bijections $f_1(t_0) = r_0$, $f_1(t_3) = r_3$ (solid lines), and $f_2(t_1) = r_1$, $f_2(t_3) = r_2$ (dashed lines).

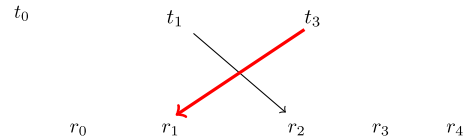


Fig. 3. Constraint 1: Modem packets associations must be ordered (the figure shows an invalid association).

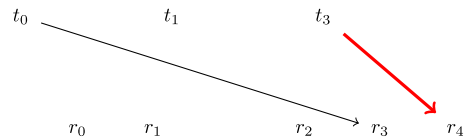


Fig. 4. Constraint 2: There are physical limits to the change in distance between two associations.

specified parameter ("granularity") can be used to reduce the resolution of the reported timestamps to match the application's needs. Furthermore, since the acoustic modem clock might wrap around too frequently, depending on hardware specifics (in our case, the acoustic modem has a 32-b microsecond clock that wraps around every $2^{32}/10^6$ s, or approximately 1 h and 12 min), the network reports and transmits the so-called "unwrapped" timestamps, by adding $2^{32}$ to the reported timestamps every time the microsecond clock wraps around. An "upper bound" property can be used to adapt the maximum wrap-around time to match the application requirements. Finally, "relative encoding" is used to further reduce the communication footprint of the encoding of transmission timestamps. Within each message the newest transmission timestamp is encoded as is; for all other timestamps within the same message, the difference with the newest timestamp is encoded. The "timestamp span" property sets the maximum difference between the first and last timestamps. Timestamps that are older than the newest timestamp minus the span are discarded.

The source addresses associated with reception timestamps are encoded in 4 b. Some additional bits are required to encode the number of included transmission and reception timestamps.

### B. Associating Timestamps

Whenever a node transmits or receives a modem packet, it registers the corresponding timestamp. Each node receives transmission and reception timestamps from other nodes, through the messages generated as described in the previous section. Denote a sequence of $n$ transmission timestamps $T_a$ of node $a$ as $T_a = \{t_0, \ldots, t_{n-1}\}$, and a sequence of $m$ reception timestamps $R_b$ of node $b$ as $R_b = \{r_0, \ldots, r_{m-1}\}$. To work out the clock offset and drift between nodes $a$ and $b$, it needs to be understood which $t_i$ corresponds to which $r_j$ or, in other words, which $t_i$ and $r_j$ were generated by the same modem packet. The association problem is to find the largest partial bijection(s) $f : T_a \to R_b$. The bijection is partial because, in general, it cannot be guaranteed that $T_a$ contains all transmission timestamps, or that $R_b$ contains all reception timestamps, due to the limited capacity of underwater communications (i.e., packet loss or limited bandwidth). Note that only on node $a$ will $T_a$ contain all transmission timestamps. The solution does not necessarily have to be unique, in particular, smaller sets $T_a$ and $R_b$ are likely to produce multiple solutions (see Fig. 2).

There are several constraints that help to narrow down the association problem. Modem packets cannot overtake each other. For example, associations cannot "cross" each other, as in Fig. 3. Let $t_i \mapsto r_j$ stand for a transmission and reception timestamp pair that was generated by the same modem packet. Let both sequences be sorted in the ascending order, i.e., $t_{i-1} < t_i$ and $r_{j-1} < r_j$. Then, if $t_i \mapsto r_j$, then $\forall k < i, l > j, \neg(t_k \mapsto r_l)$, and $\forall k > i, l < j, \neg(t_k \mapsto r_l)$ (constraint 1).

Additionally, there are physical limits to how much the distance can change between two associations (see Fig. 4). Assume $t_i \mapsto r_j$ and the existence of $t_k, k \neq i$, and $r_l, l \neq j$. The elapsed time between $r_j$ and $r_l$ can be calculated as $|r_l - r_j|$, and the change in distance as $|t_k - t_i + r_j - r_l|c$, where $c$ is the sound speed. The estimated speed $s_{k,l}$ is the change in distance divided by the elapsed time. Let $v_a$ and $v_b$ be the maximum speed of nodes $a$ and $b$, respectively. Then, it follows that $t_i \mapsto r_j$ and $t_k \mapsto r_l$ only if $s_{k,l} \leq v_a + v_b$ (constraint 2).

With these constraints the association problem can be formulated as a linear assignment problem (LAP). Assume $t_i \mapsto r_j$. The cost function $C_{k,l}$ to be minimized for the association of $t_k$, $k \neq i$, to $r_l$, $l \neq j$, is defined as

$$C_{k,l} = \begin{cases} s_{k,l}, & \text{if constraints 1 and 2 are met} \\ \infty, & \text{otherwise.} \end{cases} \tag{21}$$

In (21), the choice of the specific cost values when constraints 1 and 2 are met does not play a key role, as long as the cost values are greater than zero. The specific choice made here finds the association at minimum node velocity. Given that, in most underwater networks, nodes will typically be stationary or moving at slow speeds, if only to conserve energy, this choice is considered to be reasonable. Note also that the result of the optimization may be impossible, because it is not guaranteed that the cost between any two subsequent mappings is less than $v_a + v_b$ ("gate threshold"). In this case, an extra check must be performed to discard impossible results.

From an implementation standpoint, the association of transmission and reception timestamps is done using the Jonker–Volgenant algorithm [19], which is a faster implementation of the Hungarian method [20], also known as the Munkres algorithm, for solving the LAP.

The following steps are done in sequence by the association algorithm to eventually find a solution.

1) For each $i, j$, partial bijections $f_{i,j}$ are calculated and impossible ones are discarded. Only those bijections that associate the largest number of $t_k$'s to an $r_l$ are kept. The algorithm tends to find identical bijections more than once because if $f_{i,j}(t_k) = r_l$, then it is likely that $f_{k,l}(t_i) = r_j$.
2) The set of different bijections is calculated, and the number of times each bijection is found is recorded.

3) Only one of these bijections is selected. Let this selected bijection be $f$. Three cases are distinguished.
   a) Only one bijection was found. This bijection is selected.
   b) Several different bijections $f_k$ were found, but only one of those (let this be $f_{\hat{k}}$) associates a $t_i$ to each $r_j$, i.e., $\forall j, 0 \leq j < m, \exists i, 0 \leq i < n, t_i \mapsto r_j$. This bijection is selected ($f = f_{\hat{k}}$).
   c) In all other cases, the largest common subset of all found bijections is selected (i.e., the associations present in all found bijections).
4) It is established whether the selected bijection $f$ is a valid association. An association is considered valid only if it is of sufficient size. On node $a$, $T_a$ contains all transmission timestamps. In this case, the selected bijection is considered valid only if it associates a $t_i$ to each $r_j$, i.e., $\forall j, 0 \leq j < m, \exists i, 0 \leq i < n, t_i \mapsto r_j$. When a (receiver) node does not have all transmission timestamps, the association is considered to be valid if $|f|$ is greater than a user-specified "minimum solution size."

Since the computation time increases with the number of timestamps that have to be associated, it is necessary to define the "maximum number of timestamps" $H_{\max} = |T_a| + |R_b|$ to be used for the optimization. This is equivalent to having a running window of timestamps of length $H_{\max}$. This provides two additional benefits: since the algorithm only considers a time-limited history, it is able to adapt to changing conditions (e.g., change in the quality of the acoustic channel causing intermittent communications), and it helps improve performance in the presence of modem clock resets.

### C. Linear Regression

Given bijections $f : T_a \to R_b$ and $g : T_b \to R_a$, a series of two-way packet exchanges $p_0 = t_{a,i}$, $q_1 = f(t_{a,i})$, $q_2 = t_{b,j}$, $p_3 = g(t_{b,j})$ can be derived. The "maximum round-trip time" property imposes a limit on the duration of two-way packet exchanges $g(t_{b,j}) - t_{a,i} = p_3 - p_0$ that will be used to estimate the clock offset and drift.

The clock offset and drift are then estimated as described in Section IV-B, but with two simplifications: no attempt is made to iteratively refine the drift estimate, as it was found that the impact of iteration was minimal, and no attempt is made to iteratively fine tune the intervals caused by node mobility, again because the impact of this iteration was found to be minimal.

## VI. Field Trials

ROCS was tested in the field during the COLLAB-NGAS14 experimental campaign, which was held October 19–31, 2014, off the west coast of Italy. The objectives of the campaign included testing AUV autonomous behaviors, sonar signal processing, and AUV navigation and localization, among other objectives. For the purpose of this work, we focus on the COLLAB-NGAS14 activities that took place from October 29 to 31, 2014.

### A. Site Description

The data presented in this paper were collected at the site shown in Fig. 5 near 43°46′31″ N, 10°2′0″ E, off the coast of Tuscany, Italy. The entire area of operation is a 7.5 × 7.5 km square. Water depth in the area decreases gradually from around 25 m (the northeast region) to around 60 m (the southwest region).

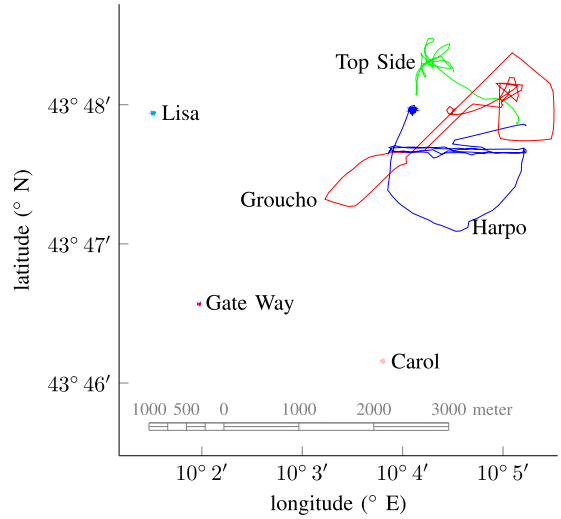


Fig. 5. Platforms deployed on October 31, 2014, during the COLLAB-NGAS14 campaign. Similar deployment used throughout the experimental period.

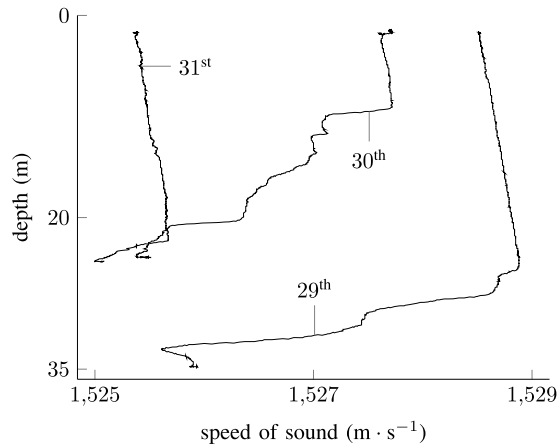


Fig. 6. Sound-speed profile measured on October 29–31, 2014.

The sound-speed profile, as measured at a location close to the northeast corner of the operational area, is shown in Fig. 6. Note that in the first 25 m, the sound speed is nearly constant for October 29 and 31, 2014, with more variability registered on October 30, 2014.

### B. Experimental Setup

The deployed acoustic network was composed of one moored gateway buoy, two wave gliders (Carol and Lisa), two CMRE Ocean Explorer (OEX) AUVs (*OEX Groucho* and *OEX Harpo*), and one modem mounted on *NRV Alliance*. Fig. 5 shows the deployment on October 31, 2014, with very similar deployments installed for other operational days. All nodes were equipped with the EvoLogics 7–17-kHz acoustic modem. The modems of the gateway buoy, wave gliders, and *NRV Alliance* were all deployed at a fixed depth of 25 m. These nodes were equipped with a GPS receiver that provided accurate position information. *OEX Groucho* was commanded to navigate at 25-m depth; *OEX Harpo* was kept at a depth of 15 m. Both vehicles relied on

TABLE I
USER-SELECTED ROCS SETTINGS DURING COLLAB-NGAS14

| Property | Value | Remarks |
|---|---|---|
| *encoded size* | 58 bytes | the maximum size of an EvoLogics modem packet is 64 bytes, 6 bytes are required for the network header [18] |
| *number of $p_0$'s* | 5 | |
| *number of $p_3$'s* | 1000 | large number is chosen to ensure all available payload is filled with reception time stamps |
| *granularity* | 100 μs | |
| *upper bound* | $2^{36}$ μs | a little over 19 hours |
| *timestamp span* | $3 \times 10^8$ μs | 5 minutes |
| *maximum number of timestamps* | 60 | this limits the computing resource footprint of the algorithm |
| *gate threshold* | 5 m · s$^{-1}$ | the OEX vehicles have a maximum speed of about 1:5 m · s$^{-1}$, an additional 1 m · s$^{-1}$ was added to allow for currents |
| *minimum solution size* | 10 | 1/3 of the maximum solution size |
| *maximum round trip time* | 70s | this effectively limits a two-way packet exchange to two packets transmitted within one TDMA frame |

a high-quality inertial navigation unit for positioning and navigation (navigation error $\sim 0.1\%$ of the distance traveled). Note that although the wave gliders are mobile assets, they were station keeping throughout the trial.

Table I summarizes all the user-selected settings for the clock synchronization algorithm that were in place during the COLLAB-NGAS14 trial. With these settings, the newest transmission timestamp requires $\lceil \log_2(2^{36}/100) \rceil = 30$ b, and the other, relatively encoded transmission timestamps require $\lceil \log_2(3 \times 10^8/100) \rceil = 22$ b. A reception timestamp requires $\lceil \log_2(2^{36}/100) \rceil = 30$ b plus 4 b for the source address $= 34$ b. If five transmission timestamps are included, the remaining space allows for the inclusion of additional nine reception timestamps. This adds up to $4 + 30 + 4 \times 22 + 9 \times (30 + 4) = 428$ b. Parts of the remaining $58 \times 8 - 428 = 36$ b include encoded book-keeping information such as the number of transmission and reception timestamps.

## VII. RESULTS

A series of experiments were carried out to evaluate the performance of the clock synchronization algorithm. Two types of results are discussed in this section, using data collected on October 29 and 30, 2014. The synchronization performance using timestamps collected onboard and applying the algorithm in postprocessing is discussed first. The postprocessing estimates all clock offsets and drifts using the entire history of collected timestamps, as if all transmission and reception timestamps were known on all nodes, with perfect knowledge of transmission–reception timestamp association. Second, online results are presented, where the offsets and drifts were estimated in real time on the in-water assets based on imperfect knowledge. The postprocessing results are intended to form a ground truth against which the online results can be compared.

### A. Postprocessing Results

Results from October 29, 2014, are shown in Fig. 7 for two fixed nodes, wave glider Carol (node 12) and the gateway buoy (node 10). Results of the synchronization between the two OEX AUVs are shown in Fig. 8. The close fit of the model shows that indeed the linear clock model is a reasonable assumption, with the difference between clocks being dominated by a linear clock skew.

The postprocessing clock synchronization result between *OEX Groucho* (node 2) and wave glider Lisa (node 11) for October 30, 2014, is shown in Fig. 9. The distance between the nodes evolved from a maximum of 3.6 km at the beginning of the mission, to a minimum of 1 km (midmission), and then increased again to 2.5 km. The intermittent behavior of the acoustic communication between the two nodes is clearly visible, with gaps of around 1 h throughout the day. During the same time, *OEX Groucho* achieved better communication with the modem of *NRV Alliance* (see Fig. 10), which was at a distance of about 1 km. Similar results were obtained for other node pairs. For a quantitative measure of the fidelity of the postprocessing results, consider a directed graph with a vertex for each node, and an edge from vertex `A` to vertex `B` if node `A` was able to estimate the clock offset and drift with respect to node `B`. Each edge `AB` has two properties: the calculated clock drift $\delta_{\mathtt{AB}}$, and the time elapsed between the start of the first two-way message exchange and the end of the last two-way

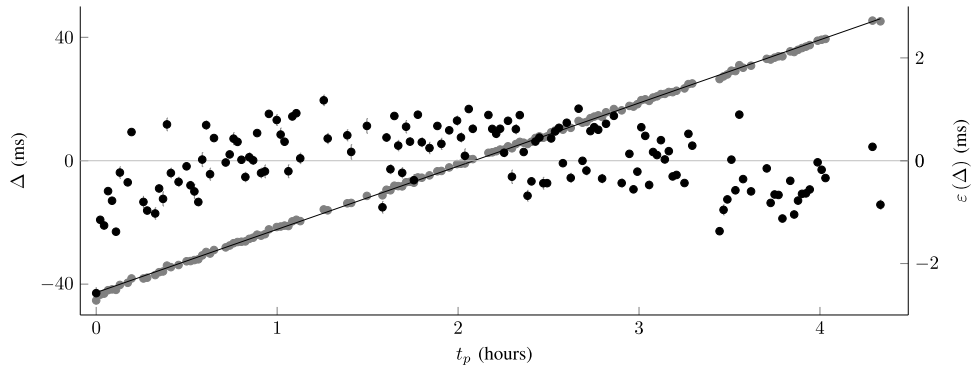


Fig. 7. Clock synchronization results obtained on October 29, 2014, between wave glider Carol (node 12) and the gateway buoy (node 10). Distance between the nodes is around 3 km. Gray dots are the calculated $\Delta_{pq}$. The diagonal black line shows the result of the linear regression calculated according to (11), normalized such that the line is centered around 0. The black dots show the residual $\varepsilon(\Delta_{pq})$, and the vertical bars show the uncertainty due to range rate uncertainty. Because the gateway buoy is stationary, the uncertainty in range rate is small, and only few vertical bars are even visible.

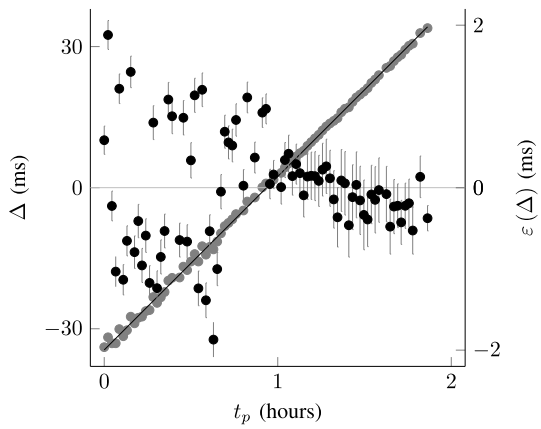


Fig. 8. Gray dots show the calculated $\Delta_{pq}$ for the clock offset between *OEX Groucho* (node 2) and *OEX Harpo* (node 3) on October 29, 2014. The diagonal line shows the result of the linear regression calculated according to (11), normalized such that the line is centered around 0. The black dots show the residual $\varepsilon(\Delta_{pq})$, and the vertical bars show the interval due to the uncertainty in the nodes' speed. The vertical bars are small because the distance between nodes 2 and 3 is short, from 250 to 750 m. The large number of outliers is due to the vehicles being on the surface during the first hour.

TABLE II
CYCLICAL CLOCK DRIFT POSTPROCESSING RESULTS

| day | min ($\tau_C$) (hh:mm) | № cycles | $\overline{\Sigma\delta}$ (ms · h$^{-1}$) | max($\Sigma\delta$) (ms · h$^{-1}$) |
|---|---|---|---|---|
| 29 | 0:40 | 77 | 0.61 | 1.74 |
| 29 | 1:10 | 46 | 0.54 | 1.67 |
| 30 | 4:00 | 295 | 0.16 | 0.51 |
| 30 | 7:20 | 104 | 0.11 | 0.37 |
| 31 | 3:10 | 344 | 0.25 | 0.91 |

message exchange $\tau_{\mathtt{AB}}$. In graph theory, a walk is a sequence of vertices, with each vertex connected to the next vertex through an edge, and a cycle is a walk that starts and ends at the same vertex but never repeats another vertex or an edge. For example, in a network composed of three nodes `A`, `B`, and `C`, and assuming that on each node the clock offset and drift have been calculated for both nodes, three cycles $\mathtt{A} \to \mathtt{B} \to \mathtt{A}$, $\mathtt{B} \to \mathtt{C} \to \mathtt{B}$, and $\mathtt{A} \to \mathtt{C} \to \mathtt{A}$ of length 2 as well as two cycles $\mathtt{A} \to \mathtt{B} \to \mathtt{C} \to \mathtt{A}$ and $\mathtt{A} \to \mathtt{C} \to \mathtt{B} \to \mathtt{A}$ of length 3 can be calculated. For each cycle $C$, two properties are calculated: its cumulative clock drift $\Sigma\delta$, which is the sum of the clock drifts of all edges, and its duration $\tau_C = \min(\tau_{ij})$, which is the minimum of the duration of each of the edges. The mean and maximum of $\Sigma\delta$ of all cycles of a particular minimum duration are a measure of the fidelity of the postprocessing results. See Table II, to be read as follows: on October 30, 295 cycles had a duration of 4 h or more. The average cumulative drift over the 295 cycles is 0.15 ms · h$^{-1}$, the maximum 0.50 ms · h$^{-1}$. The results show that the longer the duration of a cycle, the smaller is its cumulative clock drift, and the more suitable the calculated clock drift is for use as the ground truth.

Table III is similar to Table II, but shows the results calculated without taking into account Doppler velocity measurements, i.e., with $\dot{r}$ in (11) set to 0. The significant deterioration of the results clearly shows the importance of the Doppler velocity measurements.

### B. Online Results

The online clock synchronization results obtained on October 29, 2014, during the last hour of operation between *OEX Groucho* and *OEX Harpo* are shown in Fig. 11. The figure shows how the incremental solution is updated as newer measurements become available. Note that no Doppler data were transmitted during this sea trial and hence Doppler data were not used for the online calculations. This decreased the clock estimation accuracy considerably and is the main reason for the discrepancy between the online and offline results.

On October 29, 2014, only three wrong associations were calculated. It did not affect the result in terms of the estimated clock offset and drift, because the wrong associations either could not be paired with another association to make a round trip of sufficiently short duration (less than the "maximum round-trip time"), or the calculated propagation time was not plausible (around −700 s; this is negative 700, and not a typo).

## VIII. DISCUSSION AND CONCLUSION

This paper reports on the details of the design, implementation, and deployment of the clock synchronization algorithm ROCS for underwater acoustic networks. The algorithm is able to continuously estimate internode clock offset and drifts, based on the opportunistic exchange of modem packets, including transmission and reception timestamps. ROCS operates at the application level, with some reliance on physical-layer timestamps provided by the modem. The benefit of ROCS is

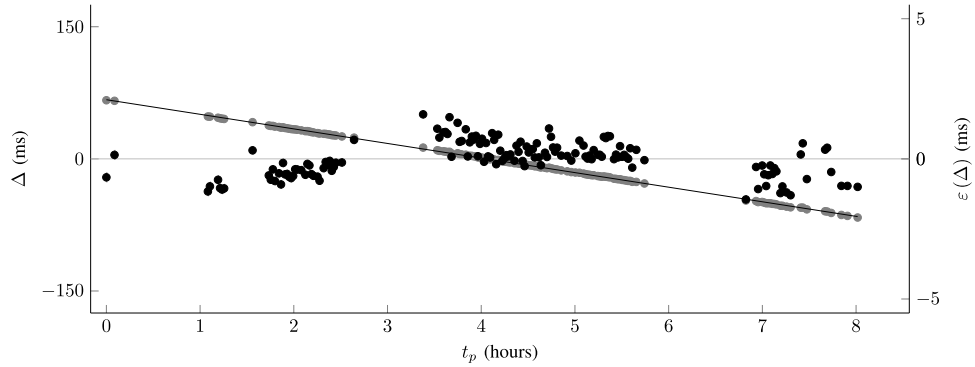


Fig. 9. Clock synchronization result obtained on October 30, 2014. The gray dots show the calculated $\Delta_{pq}$ for the clock offset between *OEX Groucho* (node 2) and wave glider Lisa (node 11). The diagonal line shows the result of the linear regression calculated according to (11), normalized such that the line is centered around 0. The black dots show the residual $\varepsilon(\Delta_{pq})$, and the vertical bars show the uncertainty due to range rate uncertainty. Because Lisa was station keeping, the uncertainty in range rate is small, in fact so small that none of the black bars are even visible. Distance between the nodes ranged from 3.6 km (beginning of the mission) to 1 km (midmission), and back to 2.5 km.

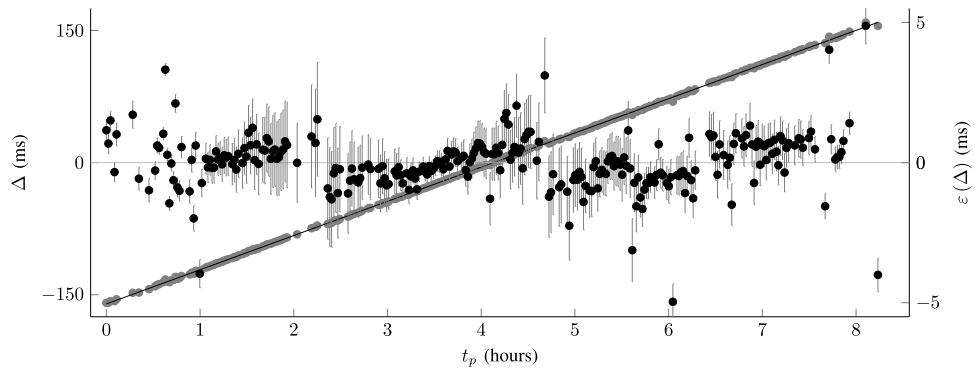


Fig. 10. Clock synchronization result obtained on October 30, 2014. The gray dots show the calculated $\Delta_{pq}$ for the clock offset between *OEX Groucho* (node 2) and *NRV Alliance* (node 1). The diagonal line shows the result of the linear regression calculated according to (11), normalized such that the line is centered around 0. The black dots show the residual $\varepsilon(\Delta_{pq})$, and the vertical bars show the uncertainty due to range rate uncertainty. Distance between the nodes was, on average, around 1 km.

TABLE III
CYCLICAL CLOCK DRIFT POSTPROCESSING RESULTS, WITHOUT DOPPLER

| day | min ($\tau_C$) (hh:mm) | Nº cycles | $\overline{\Sigma\delta}$ (ms · h$^{-1}$) | max($\Sigma\delta$) (ms · h$^{-1}$) |
|---|---|---|---|---|
| 29 | 0:40 | 77 | 12.20 | 39.82 |
| 29 | 1:10 | 46 | 10.89 | 33.88 |
| 30 | 4:00 | 295 | 0.76 | 2.94 |
| 30 | 7:20 | 104 | 0.73 | 2.94 |
| 31 | 3:10 | 344 | 1.37 | 5.80 |

that it is able to deal explicitly with the constraints of acoustic communications, such as long propagation times, intermittent communications, and node mobility. Some additional acoustic overhead is incurred by ROCS, but is intended to be distributed among network traffic in an opportunistic manner, occupying payload space after higher priority traffic has been ensonified. Depending on the application, this may be justified, for example, when the same bandwidth is used for the purpose of navigation. Additionally, ROCS is robust to some levels of packet loss, as the underlying timestamp measurement history is transmitted repeatedly, and the clock model estimation is done even with incomplete data. This means that single packet failures do not necessarily prevent the estimation of the internode clock dynamics.

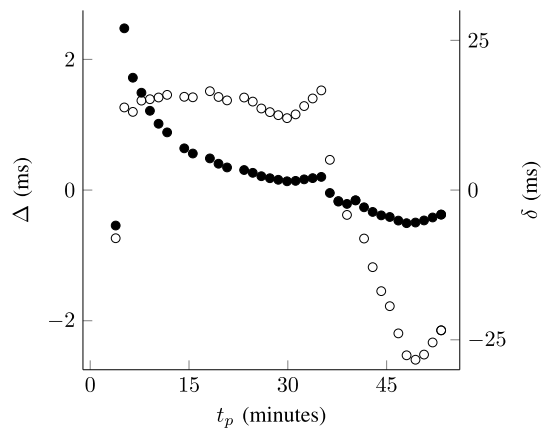


Fig. 11. Incremental output of ROCS on *OEX Groucho* (node 2), October 29, 2014, for the clock offset and drift between *OEX Groucho* (node 2) and *OEX Harpo* (node 3). The open dots shows the difference between the incrementally calculated offsets and the offset calculated during postprocessing. The closed dots show the difference between the incrementally calculated drifts and the drift calculated during postprocessing.

Details on how the approach was implemented in an operational acoustic network are reported, including some preliminary results from experimental deployments. These experimental results include both ROCS results as produced from postprocessing, and run online in the deployed nodes.

There is additional work anticipated on this topic, with more improvements to this first version of ROCS.

The cost function used to associate transmission and reception timestamps at each node biases the solution toward a solution with lower speeds. There is no theoretical justification for this, and alternative cost functions, without this bias, may be evaluated.

The association problem is somehow secondary to the core formulation of ROCS, and could be trivialized with a more creative book-keeping mechanism. For example, an identifier could be derived from a checksum of each packet, and then transmitted with each transmission and reception timestamp. To minimize overhead, a checksum of only one single bit could be used, reducing the number of potential associations for a timestamp $t_i$ by a factor of 2. Another alternative is to reduce the number of total timestamps in the synchronization history. This reduces the communications overhead (there are less timestamps that need to be transmitted), and also simplifies the association problem. This may be done by only recording and transmitting timestamps of packets that have particular attributes, such as a special checksum value, flag, or hash code.

Future research will focus on the exploitation of the concepts presented here to support navigation of mobile nodes. ROCS lends itself to providing internode ranging information, with the obvious measurement being the direct two-way node range. However, timestamps received from a third node might provide additional information on the location of nodes. If a node `A` is able to establish the clock offset and drift with respect to some other node `B`, as well as the clock offset and drift for node `B` with respect to a third node `C`, then node `A` is able to calculate the difference in arrival time of a modem packet transmitted by node `B` and received by both nodes `A` and `C`. Additionally, a network with well-synchronized clock parameters may have sufficient fidelity to perform one-way ranging for localization and navigation purposes.

## Acknowledgment

The authors would like to thank everyone involved in the COLLaborative Asw Behaviours–Next Generation Autonomous Systems (COLLAB-NGAS14) sea trial, K. LePage and R. Goldhahn (Scientists In Charge), everyone in the Engineering Department, the captain and crew of *NRV Alliance*, T. Furfaro, and M. Collins.

## References

[1] G. Ferri, A. Munafò, R. Goldhahn, and K. Lepage, "A non-myopic, receding horizon control strategy for an AUV to track underwater targets in bistatic sonar scenarios: Proposed approach and experimental results from COLLAB13 exercise," in *Proc. Int. IEEE Conf. Decision Control*, Jun. 2014, pp. 5352–5358.

[2] R. Eustice, H. Singh, and L. L. Whitcomb, "Synchronous-clock, one-way-travel-time acoustic navigation for underwater vehicles," *J Field Robot.* vol. 28, no. 1, pp. 121–136, 2011.

[3] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Netw.* vol. 3, no. 3, pp. 281–323, 2005.

[4] K. Kebkal, O. Kebkal, V. Kebkal, and R. Petroccia, "Synchronization tools of acoustic communication devices in control of underwater sensors, distributed antennas, and autonomous underwater vehicles," *Gyroscopy Navig.* vol. 5, no. 4, pp. 257–265, 2014.

[5] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: Applications, advances and challenges," *Philosoph. Trans. A, Math. Phys. Eng. Sci.* vol. 370, no. 1958, pp. 158–175, Jan. 2012.

[6] J. Potter *et al.*, "The JANUS underwater communications standard," in *Proc. Underwater Commun. Netw. Conf.*, Sestri Levante, Italy, Sep. 2014, DOI: 10.1109/UComms.2014.7017134.

[7] A. A. Syed and J. S. Heidemann, "Time synchronization for high latency acoustic networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2006, DOI: 10.1109/INFOCOM.2006.161.

[8] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "MU-Sync: A time synchronization protocol for underwater mobile networks," in *Underwater Networks*, M. Stojanovic, P. Schniter, and W. Ye, Eds. New York, NY, USA: ACM, 2008, pp. 35–42.

[9] R. Diamant and L. Lampe, "Underwater localization with time-synchronization and propagation speed uncertainties," *IEEE Trans. Mobile Comput.* vol. 12, no. 7, pp. 1257–1269, Jul. 2013.

[10] D. Zennaro, B. Tomasi, L. Vangelista, and M. Zorzi, "Light-Sync: A low overhead synchronization algorithm for underwater acoustic networks," in *Proc. OCEANS Conf.*, May 2012, DOI: 10.1109/OCEANS-Yeosu.2012.6263491.

[11] J. Liu *et al.*, "Mobi-Sync: Efficient time synchronization for mobile underwater sensor networks," *IEEE Trans. Parallel Distrib. Syst.* vol. 24, no. 2, pp. 406–416, Feb. 2013.

[12] F. Lu, D. Mirza, and C. Schurgers, "D-Sync: Doppler-based time synchronization for mobile underwater sensor networks," in *Proc. 5th ACM Int. Workshop UnderWater Netw.*, 2010, pp. 3:1–3:8, DOI: 10.1145/1868812.1868815.

[13] J. Liu *et al.*, "DA-Sync: A Doppler-assisted time-synchronization scheme for mobile underwater sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 582–595, Mar. 2014.

[14] J. Liu *et al.*, "JSL: Joint time synchronization and localization design with stratification compensation in mobile underwater sensor networks," in *Proc. 9th Annu. IEEE Commun. Soc. Conf. Sensor Mesh Ad Hoc Commun. Netw.*, 2012, pp. 317–325.

[15] T. Khandoker, L. Fang, D. Huang, and V. Sreeram, "A robust time synchronization approach for underwater networks with inconsistent timestamps," in *Proc. Int. Symp. Commun. Inf. Technol.*, Oct. 2012, pp. 1086–1091.

[16] K. G. Kebkal and R. Bannasch, "Sweep-spread carrier for underwater communication over acoustic channels with strong multipath propagation," *J. Acoust. Soc. Amer.* vol. 112, no. 5, pp. 2043–2052, 2002.

[17] H. Marouani and M. R. Dagenais, "Internal clock drift estimation in computer clusters," *J. Comput. Netw. Commun.* 2008, DOI: 10.1155/2008/583162.

[18] A. Vermeij, T. C. Furfaro, A. Munafò, and J. Alves, "The digital communications stack, from NEMO towards a NATO underwater networked communications architecture and implementation," CMRE, NATO Unclassified, Tech. Rep. CMRE-MR-2014-008, Oct. 2014.

[19] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing* vol. 38, no. 4, pp. 325–340, 1987.

[20] J. R. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957.

**Arjan Vermeij** received the M.Sc. degree in software engineering from Delft University of Technology, Delft, The Netherlands, in 1988.

He is currently with the Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy. His research interests include underwater acoustic communication and networking, underwater acoustic localization, and underwater vehicle autonomy.

**Andrea Munafò** (M'09) received the B.Sc. degree in computer science engineering and the M.Sc. degree in automation engineering from the University of Pisa, Pisa, Italy, in 2002 and 2005, respectively, and the Ph.D. degree in automation, robotics and bioengineering from the Department of Electric System and Automation, University of Pisa, in 2009.

He worked as a Postdoctoral Student at the Interuniversity Center on Integrated Systems for the Marine Environment from 2009 to 2013. He is currently a Research Scientist at the NATO STO Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy. His main research interests span over underwater communication, multiagent cooperation, and underwater localization.

Dr. Munafò is a member of the IEEE Oceanic Engineering and Robotics and Automation Societies.

# Document Data Sheet

| Security Classification | | Project No. |
|---|---|---|
| | | |
| **Document Serial No.** | **Date of Issue** | **Total Pages** |
| CMRE-PR-2019-111 | June 2019 | 12 pp. |


**Author(s)**

Arjan Vermeij, Andrea Munafò


**Title**

A robust, opportunistic clock synchronization algorithm for ad hoc underwater acoustic networks


**Abstract**

The proliferation of deployed sea-going autonomous platforms, such as autonomous underwater vehicles (AUVs), unmanned surface vehicles (USV), and sensor nodes anchored to the seabed, makes the deployment of true underwater acoustic networks more and more feasible. An important feature of any network is the ability to synchronize the clocks of the participants, for the purpose of, e.g., time-slotted media access control (MAC) and navigation. Terrestrial clock synchronization protocols, such as the well-established network time protocol (NTP), are not readily applicable to underwater acoustic networks, because of long propagation times, low packet delivery success rates, communication ranges that vary over time in an unpredictable manner, and, in the presence of mobile nodes, the ad hoc nature of the composition of the network. This paper proposes a continuous estimation of internode clock offset and drift, based on the continuous exchange of modem packets, possibly containing transmission and reception timestamps. The proposed solution takes explicitly into account the limitations of the acoustic communication channel and network node mobility. This robust, opportunistic clock synchronization (ROCS) is robust against modem reset, and will work even if packet delivery success rates are not optimal or if no communication is possible for extended periods of time. Experimental results are given from the COLLaborative Asw Behaviours–Next Generation Autonomous Systems (COLLAB-NGAS14) campaign, held October 19–31, 2014, off the west coast of Italy. During the sea trial, the proposed clock synchronization algorithm was deployed and successfully tested within an underwater acoustic network composed of mobile and fixed nodes.



**Keywords**

Acoustic communications, clock synchronization, localization and navigation, multiagent systems, optimization, underwater acoustic networks, underwater sensor network



**Issuing Organization**

NATO Science and Technology Organization
Centre for Maritime Research and Experimentation

Viale San Bartolomeo 400, 19126 La Spezia, Italy

*[From N. America:*
STO CMRE
*Unit 31318, Box 19, APO AE 09613-1318]*

Tel: +39 0187 527 361
Fax:+39 0187 527 700

E-mail: library@cmre.nato.int