

Article

# DE-Sync: A Doppler-Enhanced Time Synchronization for Mobile Underwater Sensor Networks

Feng Zhou <sup>1,2,3,†</sup>, Qi Wang <sup>1,2,3,†</sup>, DongHu Nie <sup>1,2,3,\*,†</sup>  and Gang Qiao <sup>1,2,3,†</sup>

<sup>1</sup> Acoustic Science and Technology Laboratory, Harbin Engineering University, Harbin 150001, China; zhoufeng@hrbeu.edu.cn (F.Z.); tony.wangqi@hrbeu.edu.cn (Q.W.); qiaogang@hrbeu.edu.cn (G.Q.)

<sup>2</sup> Key Laboratory of Marine Information Acquisition and Security (Harbin Engineering University), Ministry of Industry and Information Technology, Harbin 150001, China

<sup>3</sup> College of Underwater Acoustic Engineering, Harbin Engineering University, Harbin 150001, China

\* Correspondence: niedonghu@hrbeu.edu.cn; Tel.: +86-138-3613-1826

† These authors contributed equally to this work.

Received: 30 March 2018; Accepted: 22 May 2018; Published: 25 May 2018



**Abstract:** Time synchronization is the foundation of cooperative work among nodes of underwater sensor networks; it takes a critical role in the research and application of underwater sensor networks. Although numerous time synchronization protocols have been proposed for terrestrial wireless sensor networks, they cannot be directly applied to underwater sensor networks. This is because most of them typically assume that the propagation delay among sensor nodes is negligible, which is not the case in underwater sensor networks. Time synchronization is mainly affected by a long propagation delay among sensor nodes due to the low propagation speed of acoustic signals. Furthermore, sensor nodes in underwater tend to experience some degree of mobility due to wind or ocean current, or some other nodes are on self-propelled vehicles, such as autonomous underwater vehicles (AUVs). In this paper, we propose a Doppler-enhanced time synchronization scheme for mobile underwater sensor networks, called DE-Sync. Our new scheme considers the effect of the clock skew during the process of estimating the Doppler scale factor and directly substitutes the Doppler scale factor into linear regression to achieve the estimation of the clock skew and offset. Simulation results show that DE-Sync outperforms existing time synchronization protocols in both accuracy and energy efficiency.

**Keywords:** doppler; sensor node; time synchronization; mobility; propagation delay

## 1. Introduction

In recent years, underwater sensor networks have played an important role in coastal surveillance, environmental monitoring, undersea exploration, disaster prevention, mine exploration, and other fields. Considerable attention has been paid to the research and application of underwater sensor networks. However, because of the characteristics of underwater acoustic channels, the research and application of underwater sensor network communication has suffered from many challenges [1,2]. The time synchronization of underwater sensor networks, as one of the supporting technologies, is also very different from the time synchronization of terrestrial wireless sensor networks.

Time among nodes is inconsistent due to the difference of angular frequency of the crystal oscillator and the difference of boot time of the distributed systems, in which nodes go out of synchronization as time elapses. Most applications in networks require that all sensor nodes have a common notion of time. Depending on the time synchronization level required, these applications can be classified into three categories [3]. Some applications only require the order of events that occur, while other functionalities require the time interval of each of the occurrences. There are also some applications that require an absolute time at which each event occurs, and the implementation of

these applications depends on a precisely synchronized time among network nodes. For example, the time-sensitive data with time stamping information from multiple sensor nodes are then aggregated to a cluster head node, and other applications such as node tracking and location require precise timing. In addition, not only would the application layer require time synchronization; other layers, such as the medium access control (MAC) layer, also have considerable functionalities that benefit from time synchronization.

Numerous time synchronization protocols for terrestrial wireless sensor networks have been proposed in various publications, such as the fine-grained network time synchronization using reference broadcasts (RBS) [4], the timing-sync protocol for sensor networks (TPSN) [5], the flooding time synchronization protocol (FSTP) [6] and others [7–10]. However, these protocols cannot be directly applied to underwater sensor networks. This is because they assume that the propagation delay among nodes is negligible. However, in underwater acoustic communication, with a propagation speed (roughly 1500 m/s in water) of nearly five orders of magnitude lower than terrestrial radio frequency (RF), assumptions about rapid communication are incorrect. Meanwhile, due to the mobility of the sensor nodes that are not propelled or self-propelled, the propagation delay among nodes in water is a dynamic variation, which should be considered in the new approach. In addition, the terrestrial synchronization algorithms typically allow nodes to perform frequent re-synchronization and are not highly dependent on available bandwidth and energy constraints. In contrast, underwater acoustic communication systems may have a bandwidth of only a few hundred kilohertz, while bandwidth would drop to a few kilohertz if the targeted communication range reaches several tens of kilometers [1]. So the bandwidth of the system restricts time synchronization overhead that is not sufficient for frequent re-synchronization. Furthermore, nodes restricted by energy constraints and recharging limitations should avoid frequent re-synchronization.

Time synchronization solutions for underwater sensor networks aim to solve the challenges, including the long and dynamic propagation delay, and especially the dynamic propagation delay caused by the mobility among nodes [11,12]. This is an important issue on which recent synchronization schemes focus. In this paper, through analyzing the existing protocols, we propose a Doppler-enhanced time synchronization scheme for mobile underwater sensor networks, called DE-Sync. Our new scheme is a novel synchronization solution for the context of the dynamic propagation delay.

Different from the existing time synchronization protocols based on the Doppler effect, DE-Sync has two main distinguishing attributes. First of all, DE-Sync considers the effect of the clock skew during the process of estimating the Doppler scaling factor, which compensates for the estimation error introduced by the clock skew and increases the accuracy of estimation in the Doppler scale that affects the accuracy of time synchronization. Secondly, the Doppler scaling factor, instead of the relative moving speed between nodes, is contained in the exchange messages, which reduces the quantity of division computations. Because the acoustic velocity in water is a function of depth, temperature and salinity, our new approach does not directly involve the calculation of acoustic velocity, which reduces the error of estimating the clock skew and offset during the process of linear regression, and improves the accuracy of time synchronization. In our comparison of simulation results, the performance of DE-Sync protocol is better than that of existing time synchronization protocols for mobile underwater sensor networks, in both accuracy and energy efficiency.

DE-Sync proposed in this paper deals with time synchronization based on Doppler scale estimation. For underwater acoustic communication, Doppler scale estimation is typically a critical research topic. There are two popular methods for Doppler scale estimation [13]. One method is to send Doppler insensitive signals for Doppler scale estimation [14–17]. Another method is to send a Doppler sensitive waveform for Doppler scale estimation [18–21]. Both of the two methods have a common application limitation, which is not ideal for real time processing of the estimation algorithm. DE-Sync adopts a combined Doppler scale estimation scheme that involves two steps to enhance the accuracy of the Doppler scale factor acquired in real time [22–24]. In the first step, one linear frequency modulation (LFM) and two identical short orthogonal frequency-division multiplexing

(OFDM) symbols preceded by a cyclic prefix (CP), are used as a preamble for initial Doppler scale estimation. In the second step, fine Doppler scale estimations can be achieved based on the CP of each CP-OFDM block.

The remainder of this paper is organized as follows. In Section 2, we review the previous works of the existing time synchronization protocols for underwater sensor networks. We then provide a description of DE-Sync in Section 3. Simulation results are shown in Section 4. Finally, we give our conclusions in Section 5.

## 2. Related Work

The design of time synchronization protocol for underwater sensor network needs to consider two main factors, one is the long propagation delay during communication between nodes, and the other is the relative mobility between nodes. At present, a few time synchronization protocols for underwater sensor networks have been proposed, such as the time synchronization for high latency acoustic networks (TSHL) [25], the time synchronization protocol for underwater mobile networks (MU-Sync) [26], the efficient time synchronization for mobile underwater sensor networks (Mobi-Sync) [27], the Doppler-based time synchronization for mobile underwater sensor networks (D-Sync) [28], and the time synchronization scheme for mobile underwater sensor networks (TSMU) [29]. However, these protocols have their own application scenarios and they all make different assumptions.

TSHL was the first time a synchronization algorithm was designed for high latency networks, networks in which the algorithm is organized in two phases. TSHL assumes the constant propagation delay among sensor nodes and, essentially, they assume a static network; however, nodes typically move with the ocean current (around 0.83–1.67 m/s) [30], especially in the case of self-propelled vehicles which move faster, such as autonomous underwater vehicles (AUVs); therefore, the protocol cannot handle mobile scenarios.

MU-Sync takes account of time variability in the propagation delay due to the relative motion of nodes using a two-phase operation, namely the skew and offset acquisition phase, and the synchronization phase. MU-Sync assumes that the one-way propagation delay can be calculated as half of the round trip time, which significantly deteriorates the performance of MU-Sync when the nodes move fast, especially when the unsynchronized nodes respond to the cluster head for a long period of time.

Mobi-Sync introduces spatial correlation of the nodes' velocities to estimate the varying time propagation delay. In the protocol, all nodes are classified into three categories: surface nodes, super nodes and ordinary nodes. Surface nodes are equipped with a global positioning system (GPS) to obtain the global time reference, while super nodes can communicate directly with surface nodes to achieve time synchronization. However, Mobi-Sync applies only to certain special scenarios, which is because they assume that the exact correlation among neighboring nodes is known, which is difficultly obtained for many actual networks such as those with a long distance between nodes or on self-propelled vehicles.

D-Sync leverages the Doppler shift caused by the relative motion of nodes in underwater environments. D-Sync does not consider the effect of the skew during the process of estimating the Doppler scaling factor, which reduces the accuracy of the Doppler shift estimation and affects the accuracy of time synchronization. Therefore, as the initial skew increases, the accuracy of time synchronization deteriorates.

TSMU utilizes the Doppler effects caused by the relative motion between nodes during the process of time synchronization. Meanwhile, they consider the effect of the skew during the process of estimating the Doppler scaling factor from the physical layer. Furthermore, the Kalman filter and calibration process are exploited during the process of time synchronization. However, due to the velocity of sound in water as a function of depth, temperature and salinity, the estimation of sound velocity is directly applied into the computation during the process of synchronization, causing extra

errors. In addition, there are three parameters to be adjusted, initial skew, etc., which affect the accuracy of synchronization and efficiency.

The closest work to our proposed solution is D-Sync, and there are two key differences between DE-Sync and D-Sync: one is that DE-Sync considers the effect of the clock skew during the process of estimating the Doppler scaling factor, and the other is that DE-Sync uses the Doppler scaling factor instead of the relative moving speed among nodes when performing linear regression. Therefore, a performance comparison between DE-Sync and D-Sync is given in Section 4.

### 3. Protocol Description

#### 3.1. Overview of the DE-Sync

The clock timer inside each node uses a crystal oscillator operating at a certain angular frequency, which determines the rate at which the clock runs [3]. Typically, the clock inside each node has an intrinsic drift in angular frequency due to the manufacturing process. The numerous synchronization protocols use a common time model that DE-Sync also follows to estimate the skew and offset. The local time of any node is relevant to the reference time by Equation (1).

$$T = \alpha t + \beta \quad (1)$$

where  $T$  denotes the local time of nodes at time  $t$ ,  $t$  is the reference time, i.e., beacon time.  $\alpha$  and  $\beta$  are the relative clock skew and the offset, respectively. Moreover,  $\alpha$  is related to the angular frequency of the crystal oscillator and  $\beta$  is affected by the starting time of the system.

The time synchronization procedure of DE-Sync consists of three phases: data collection, linear regression and calibration. In the data collection phase, the unsynchronized node collects the time stamps from synchronization messages as well as their Doppler scale factors, with two-way synchronization message exchanges. In the linear regression phase, the unsynchronized node performs the first linear regression with LS (least square estimation) to estimate the clock skew and offset. In the calibration phase, the unsynchronized node performs the calibration process by updating the initial skew, and re-performs the synchronization process to obtain the final clock skew and offset. All parameters are listed in Table 1.

**Table 1.** Parameter Description.

Parameters	Description
$A$	Beacon node
$B$	Unsynchronized node
$\alpha$	Clock skew
$\beta$	Clock offset
$a_m$	Doppler scale factor induced by the node mobility
$v_m$	Average relative speed of nodes
$c$	Sound propagation speed
$a_{AB}$	Measured Doppler scaling factor from $A$ to $B$
$a_{BA}$	Measured Doppler scaling factor from $B$ to $A$
$v_{AB}$	Measured relative speed from $A$ to $B$
$v_{BA}$	Measured relative speed from $B$ to $A$
$T_1$	Sending time-stamp of unsynchronized node
$T_4$	Receiving time-stamp of unsynchronized node
$T_2$	Receiving time-stamp of beacon
$T_3$	Sending time-stamp of beacon
$t_2, t_3$	Reference time at $T_2$ and $T_3$
$t_1, t_4$	Reference time at $T_1$ and $T_4$
$T_{backoff}$	Response time, that is $T_3 - T_2$
$\tau_1$	Propagation delay of synchronization request from $B$ to $A$
$\tau_2$	Propagation delay of synchronization response from $A$ to $B$
$\Delta d$	Relative moving distance from $t_2$ to $t_4$

### 3.2. Impact of the Clock Skew

Let us assume that beacon node  $A$  is used as a reference node which has an ideal clock and node  $B$  is an unsynchronized node which has a drifting time—a skew with respect to an ideal clock. Let node  $B$  be synchronized with node  $A$ . The signal received from the baseband, going from node  $A$  to node  $B$ , is given by Equation (2), which can be formulated as the summation of signals arriving along multiple physical paths [29,31].

$$y_{AB}(n) = \sum_{p=1}^{N_p} A_p x_{AB}((1 + a_{AB})n - \tau_p) \quad (2)$$

In this equation,  $N_p$  stands for the path number,  $a_{AB}$  denotes the measured Doppler scale factor, and  $A_p$  and  $\tau_p$  denote amplitude and delay of the  $p$ th path, respectively. Based on Equation (1), due to a drift of the skew between the clocks of nodes, when the time in node  $A$  is expressed as  $t[n] = nT_s$ , the basis time in node  $B$  can be written as  $t[n] = \alpha(nT_s)$ .  $a_{AB}$  depends on both relative motion speed and clock skew difference among nodes, and it is expressed as:

$$1 + a_{AB} = \alpha(1 + a_m) \quad (3)$$

where  $a_m$  denotes the Doppler scale factor induced by the node mobility. Similarly, upon receiving the synchronization message sent from node  $B$ , the basis time in node  $A$  can be written as  $t[n] = (nT_s)/\alpha$ , the Doppler scale factor measured by node  $A$ , namely  $a_{BA}$ , is expressed as:

$$1 + a_{BA} = \frac{1 + a_m}{\alpha} \quad (4)$$

So, based on the measurement of the Doppler scale factor,  $a_m$  is expressed as:

$$a_m = \begin{cases} ((1 + a_{AB})/\alpha) - 1 & A \rightarrow B \\ \alpha(1 + a_{BA}) - 1 & B \rightarrow A \end{cases} \quad (5)$$

As a result, since the clock skew  $\alpha$  in the initial phase is unknown,  $\alpha$  is assigned with an initial value "1". After the first linear regression,  $a_m$  can be calibrated based on the estimation of  $\alpha$ . Moreover,  $a_m$  is used to re-perform the synchronization process to obtain the final clock skew and offset.

### 3.3. Details of the DE-Sync

In DE-Sync, the synchronization process is initiated by the unsynchronized node. When the unsynchronized node  $B$  moves to the coverage area of the underwater sensor networks, it sends a synchronization request to beacon  $A$ . Meanwhile, node  $B$  records the sending time-stamp  $T_1$  obtained from the MAC layer right before the message departs. Upon receiving the synchronization request, beacon  $A$  marks its local time as the receiving time-stamp  $T_2$  and records the Doppler scale factor  $a_{AB}$  obtained from the physical layer. Then, beacon  $A$  needs to back off a random time, namely  $T_{backoff}$ , to avoid message collisions before it transmits a response message back to node  $B$  at time  $T_3$ , which contains  $T_2$ ,  $T_3$  and  $a_{BA}$ . As before, when receiving the synchronization response message, node  $B$  records its receiving time and the Doppler scale factor  $a_{AB}$ . After a few rounds of message exchanges, node  $B$  will collect a set of time-stamps consisting of  $T_1$ ,  $T_2$ ,  $T_3$  and  $T_4$ , and the measured Doppler scale factor consisting of  $a_{BA}$  and  $a_{AB}$ . Figure 1 shows the message exchange process between node  $A$  and node  $B$ .

Now, we will derive a set of equations used by DE-Sync to synchronize nodes.

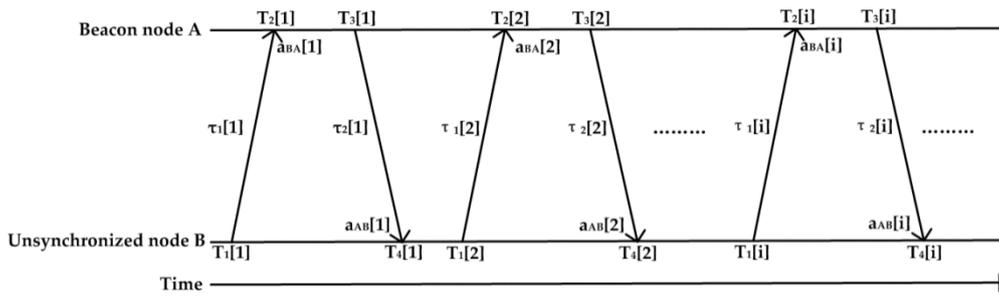


Figure 1. DE-Sync Message Exchange Scheme.

As given by Equation (1), the following set of equations of the node’s local time can hence be derived:

$$\begin{aligned} T_1 &= \alpha t_1 + \beta \\ T_2 &= t_2 \\ T_3 &= t_3 \\ T_4 &= \alpha t_4 + \beta \end{aligned}$$

$t_1, t_2, t_3$  and  $t_4$  denote the reference time, and they derive the following relationship:

$$\begin{aligned} t_2 &= t_1 + \tau_1 \\ t_4 &= t_3 + \tau_2 \end{aligned}$$

Hence, the local time of node A and node B can be expressed as:

$$\begin{aligned} T_1 &= \alpha(T_2 - \tau_1) + \beta \\ T_4 &= \alpha(T_3 + \tau_2) + \beta \end{aligned} \tag{6}$$

Considering sensor node mobility, propagation delay  $\tau_1$  is not essentially equivalent to  $\tau_2$ , and based on Equation (6), we get:

$$\begin{aligned} \tau_1 + \tau_2 &= ((T_4 - T_1)/\alpha) + T_2 - T_3 \\ \tau_2 - \tau_1 &= \Delta d/c \\ \Delta d &= \int_{t_2}^{t_4} v_{AB}(t) dt \end{aligned} \tag{7}$$

Let  $v_m$  be the average of relative speed among nodes from  $t_2$  to  $t_4$ , hence  $\Delta d = v_m(t_4 - t_2)$ . Combining with (6), we get:

$$\tau_2 - \tau_1 = \frac{v_m}{c} \times \left( \left( \frac{T_4 - \beta}{\alpha} \right) - T_2 \right) \tag{8}$$

If we let  $\theta = \frac{v_m}{c}$ , and combine Equations (7) and (8), then  $\tau_1$  and  $\tau_2$  are solved as:

$$\begin{aligned} \tau_1 &= \frac{T_4(1 - \theta) - T_1 + \theta\beta}{2\alpha} + \frac{T_2(1 + \theta)}{2} - \frac{T_3}{2} \\ \tau_2 &= \frac{T_4(1 + \theta) - T_1 - \theta\beta}{2\alpha} + \frac{T_2(1 - \theta)}{2} - \frac{T_3}{2} \end{aligned} \tag{9}$$

$\tau_1$  and  $\tau_2$  are substituted into (6), which give us the following equation:

$$T_1 + T_4(1 - \theta) = \alpha(T_2(1 - \theta) + T_3) + \beta(2 - \theta) \tag{10}$$

Node A at  $t_2$  measures the Doppler scale factor expressed as  $a_{BA} = v_{BA}/c$ , and node B at  $t_4$  measures the Doppler scale factor expressed as  $a_{AB} = v_{AB}/c$ .

Let the average of  $v_{AB}$  and  $v_{BA}$ , namely interpolation, replace the average of relative speed among nodes. Hence  $v_m = 0.5 \times (v_{AB} + v_{BA}) + \varepsilon_v$ , i.e.,  $v_m = 0.5 \times c \times (a_{AB} + a_{BA}) + \varepsilon_v$ , where  $\varepsilon_v$  denotes error due to random noise and the interpolation error. We get:

$$\theta = 0.5 \times (a_{AB} + a_{BA}) + \frac{\varepsilon_v}{c}, \quad (11)$$

and if  $\theta$  is substituted into (10), we get:

$$2T_1 + T_4(2 - (a_{AB} + a_{BA})) = \alpha(T_2(2 - (a_{AB} + a_{BA})) + 2T_3) + \beta(4 - (a_{AB} + a_{BA})) + \varepsilon \quad (12)$$

Based on Equation (12), we perform linear regression to estimate the skew and offset, which is as follows:

$$\begin{aligned} \hat{\Lambda} &= [\hat{\alpha}, \hat{\beta}]^T = (H^T H)^{-1} H^T Y \\ Y[i] &= 2T_1^i + T_4^i(2 - (a_{AB}^i + a_{BA}^i)) \\ H[i, 1] &= 2T_3^i + T_2^i(2 - (a_{AB}^i + a_{BA}^i)) \\ H[i, 2] &= (4 - (a_{AB}^i + a_{BA}^i)) \end{aligned} \quad (13)$$

As given by (13), after  $N$  rounds of message exchanges,  $Y$  is a  $N \times 1$  vector and  $H$  is a matrix with  $N \times 2$  entries. The skew and offset can be estimated by the above equation.

As described earlier, we have discussed the impact of clock skew for the Doppler shift factor, and the skew is assigned with an initial value "1". However, the skew of the unsynchronized node cannot be 1. In the calibration phase, the initial skew is updated with the estimated skew obtained by using the Equation (13). The unsynchronized node carries out the calibration process to correct the measured Doppler scale factor and re-calculate the clock skew and offset. The calibration process will run iteratively until it reaches the final criterion, which is either when the number of runs reach the maximum iteration (for example: 2, according to the later simulation result) or the difference between estimated skews of two runs is less than the set threshold (for example: 50 part per million (ppm)).

### 3.4. Error Analysis

In DE-Sync, there are two main error sources: the error  $\varepsilon_{noise}$  due to random noise in Doppler measurements and the error  $\varepsilon_{interp}$  due to the fact that Doppler measurements are not available continuously. In each round of message exchange, the average of the two measured Doppler scale factors, obtained at the end of each message transmission, is used instead of continuous measurements, which lead to the interpolation error. In Equation (11),  $\varepsilon_v$  is expressed as:

$$\varepsilon_v = \varepsilon_{noise} + \varepsilon_{interp} \quad (14)$$

In Equation (12),  $\varepsilon$  is expressed as:

$$\varepsilon = \frac{\varepsilon_v}{c} \times (T_4 - \alpha T_2 - \beta) \quad (15)$$

Combining with (6), where  $T_4$  is substituted into (14), we get:

$$\varepsilon = \frac{\varepsilon_v}{c} \times \alpha \times (T_3 - T_2 + \tau_2) \quad (16)$$

Combining with Equation (14), where  $\varepsilon_v$  is substituted into (15), we get:

$$\varepsilon = (\varepsilon_{noise} + \varepsilon_{interp}) \times \frac{\alpha(T_{backoff} + \tau_2)}{c} \quad (17)$$

$T_{backoff} = T_3 - T_2$ . When considering (17), and the parameters affecting the estimation of error,  $\varepsilon_{interp}$  is a main factor. Referring to Equation (7), we get:

$$\varepsilon_{interp} = \frac{1}{t_4 - t_2} \times \int_{t_2}^{t_4} v_{AB}(t) dt - 0.5 \times (v_{AB} + v_{BA}) \quad (18)$$

Following the derivation in [28],  $\varepsilon_{interp} \leq \frac{(t_4 - t_2) \times (A_{max}^2 - A_{avg}^2)}{4A_{max}}$ ,  $A_{max}$  and  $A_{avg}$  denote the maximum relative acceleration and average relative acceleration, respectively. Considering equation (18), there is an indication that estimation error in DE-Sync mainly occurs from random noise, i.e.,  $\varepsilon_{noise}$ , if nodes move at a constant speed or uniform acceleration.  $\varepsilon_{noise}$  is a random variable which is subject to Gaussian distribution [20].

## 4. Performance Evaluation

### 4.1. Simulation Setup

First, we compared the performance of DE-Sync and D-Sync because D-Sync is closest to our new scheme; simulation results are obtained with the change of various parameters for a pair of nodes. Furthermore, the post-synchronization error and energy efficiency of the three schemes, namely, DE-Sync, D-Sync and MU-Sync, are compared at the end of this section. In the simulations, the sensor nodes are allowed to move randomly within an area of 1000 m by 1000 m region. The simulation signal exchange scheme is shown in Figure 1. Unless otherwise specified, we use the parameters summarized in Table 2 for our simulations. Without loss of generality, each data point is obtained as an average of the 1000 runs.

**Table 2.** Simulation Parameters.

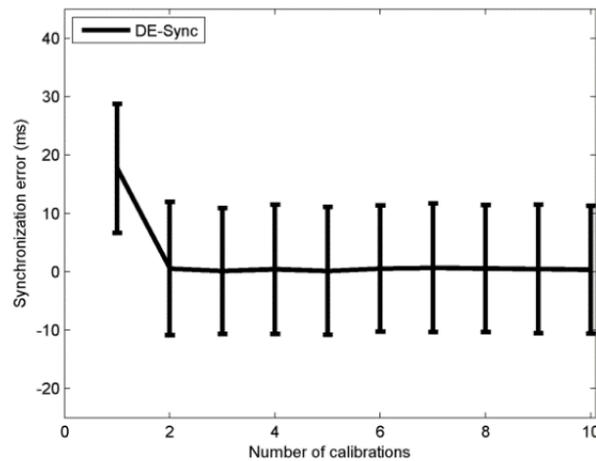
Parameters	Value
Max distance ( $d_{max}$ )	1000 m
Maximum relative speed ( $v_{max}$ )	5 m/s
Maximum relative acceleration ( $A_{max}$ )	0.1 m/s <sup>2</sup>
Maximum clock skew ( $\alpha_{max}$ )	$0.1 \times 10^6$ ppm
Clock offset ( $\beta$ )	80 ppm
Response time ( $T_{backoff}$ )	1 s
Interval between request messages ( $M_{interval}$ )	3 s
Number of messages ( $N$ )	25
Clock granularity ( $T_{gra}$ )	1 $\mu$ s
Reception jitter ( $T_{jitter}$ )	15 $\mu$ s

Note that all simulation results show the synchronization error 2 h after a node is synchronized.

### 4.2. Simulation Results and Analysis

#### 4.2.1. Number of Calibration

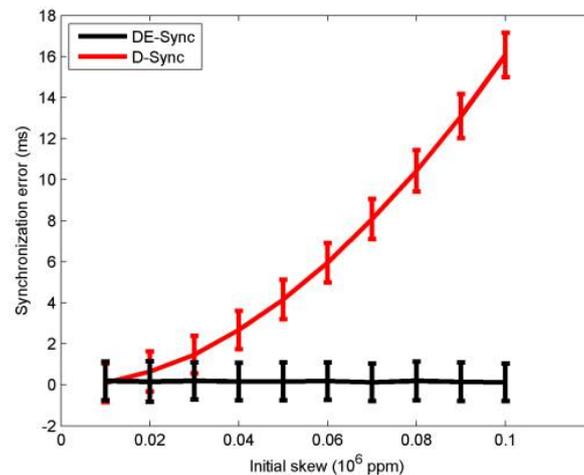
In the calibration process, DE-Sync compensates for the effect of the clock skew in the Doppler scale factor. First, we analyzed the influence of the number of calibrations on the amount of synchronization error. The simulation result is showed in Figure 2, from which we can see that when the number of calibration is 1, the Doppler scale factor is not corrected and the synchronization error is larger. The Doppler scale factor is corrected when the number of calibrations increases, starting from 2. The simulation results show that the synchronization error decreases with an increased number of calibration times, and the performance is enhanced as expected.



**Figure 2.** Performance with number of calibrations.

#### 4.2.2. Initial Skew

Due to the effect of the skew during the process of estimating the Doppler scale factor, the DE-Sync performance with its initial skew needs to be analyzed. According to the references [25–28,32], the inherent skew can be selected with the range from 50 to  $10^6$  ppm. However, the inherent skew rarely can reach  $10^6$  ppm in practice. Without loss of generality, the simulation is performed with the range of the inherent skew from  $10^4$  to  $10^5$  ppm. The simulation results are shown in Figure 3, which shows that the synchronization error of the D-Sync deteriorates with an increased skew and DE-Sync has consistent performance across different skews. Notably, DE-Sync significantly outperforms D-Sync when there is a larger skew. In other words, when the initial skew is less than a certain value ( $10^4$  ppm), the difference of performance between two schemes can be negligible. In the following simulations, we will compare the two schemes using a larger value of the skew ( $0.05 \times 10^6$  ppm).



**Figure 3.** Performance with initial skew.

#### 4.2.3. Response Time

Figure 4 shows the mean and standard deviation of synchronization error for DE-Sync and D-Sync when the response time of a node is varied between 1 s and 25 s. As shown in Equation (17), we know that the error in DE-Sync and D-Sync increases with the response time because Doppler measurements are available only when a message is exchanged between nodes. However, observing the mean of error,

DE-Sync has consistent performance across different response time, because unlike D-Sync, it takes the effect of skew into account during the process of estimating the Doppler scale factor.

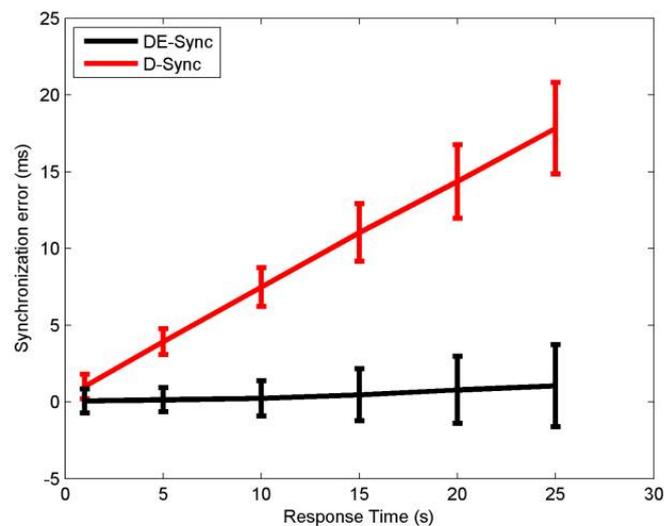


Figure 4. Performance with response time.

#### 4.2.4. Extent of Mobility

Figure 5 illustrates the mean and standard deviation of the synchronization error for DE-Sync and D-Sync when the maximum relative speed of a node is varied between 1 m/s and 5 m/s. Based on Equations (3) and (4), without considering the effect of the skew, D-Sync will introduce an estimation error of relative speed  $\varepsilon_a$ ; it can be formulated as:

$$\varepsilon_a = 0.5 \times \left( \alpha + \frac{1}{\alpha} - 2 \right) \times (v_m + c) \quad (19)$$

Considering Equation (19), if  $\alpha = 1$ , then  $\varepsilon_a = 0$ . However, for unsynchronized nodes,  $\alpha$  is not equal to 1, which causes  $\varepsilon_a$  to be larger with the increase of  $v_m$ . Further, the mean of the synchronization error of D-Sync deteriorates with increased mobility. As mentioned before, DE-Sync still outperforms D-Sync because it reduces the synchronization error induced by the effect of the skew during the process of estimating the Doppler scale factor. From Figure 5, simulation results are as expected.

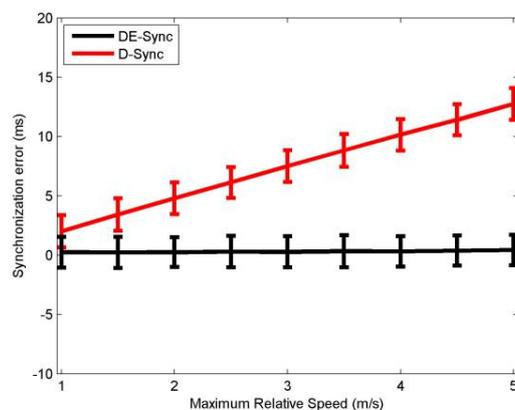


Figure 5. Performance with relative speed.

Figure 6 shows the mean and standard deviation of synchronization error for DE-Sync and D-Sync when the maximum relative acceleration of a node is varied between  $0.01 \text{ m/s}^2$  and  $0.1 \text{ m/s}^2$ . Based on Equation (18), the accuracy of DE-Sync and D-Sync are affected by the maximum relative acceleration of nodes, in which the synchronization error increases with the increase of the maximum relative acceleration. Due to the fact that the maximum relative speed increases with the increase of maximum relative acceleration, considering Equation (19), we know that the increase of maximum relative speed leads to a greater synchronization error in D-Sync than DE-Sync. From Figure 6, simulation results are consistent with expectations that DE-Sync outperforms D-Sync.

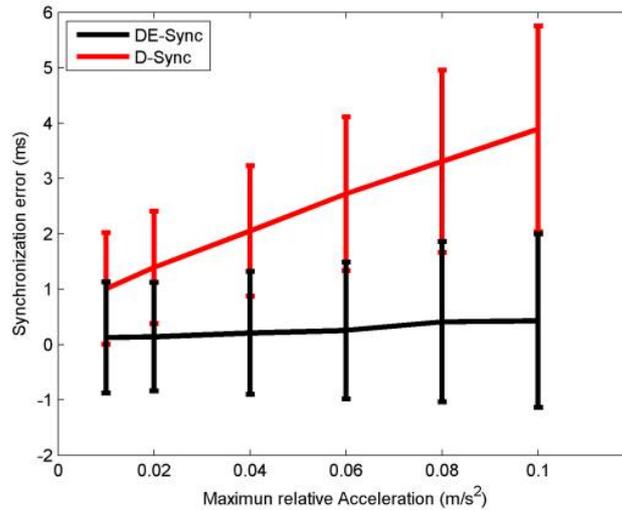


Figure 6. Performance with relative acceleration.

#### 4.2.5. Interval between Request Messages

Figure 7 shows the mean and standard deviation of synchronization error for DE-Sync and D-Sync when the time interval between request messages of a node is varied between 20 s and 120 s. Increasing the interval between request messages is equivalent to increasing the dynamic range of linear regression, which allows enough time to elapse before initiating the next message exchange. Performing Equation (13) to enhance the estimation accuracy of the skew and offset, the synchronization error decreases with the increased interval. As described before, DE-Sync has a lower mean of synchronization error than D-Sync. Observing Figure 7, simulation results are in agreement with the expectation that DE-Sync outperforms D-Sync.

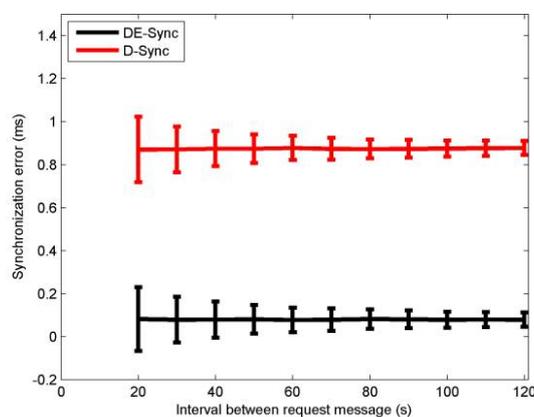


Figure 7. Effect of message interval.

#### 4.2.6. Number of Request Messages

Figure 8 shows the mean and standard deviation of synchronization error for DE-Sync and D-Sync when the number of messages of a node is varied between 5 and 45. The standard deviation of synchronization error of DE-Sync and D-Sync decreases with the increase of the number of messages. Meanwhile, we set the acceleration of the node to be non-zero, and with the increase of the number of request messages, the moving speed among nodes also increases. According to the Equation (19), the increase of moving speed leads to the increase of the estimation error of relative speed among nodes in D-sync, so the mean of synchronization error increases with the increase of the number of request messages in D-sync. DE-Sync has a lower mean of synchronization error than D-Sync does. From Figure 8, simulation results are consistent with the expectation that DE-Sync outperforms D-Sync.

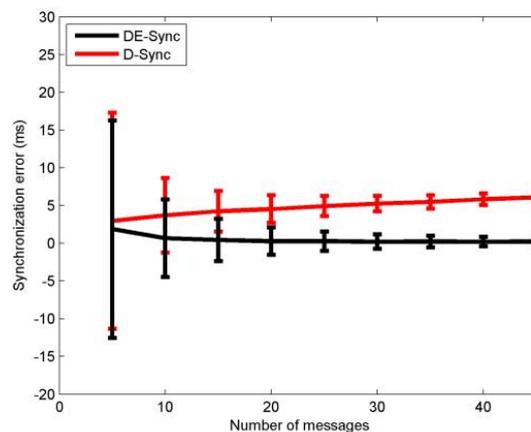


Figure 8. Performance vs number of messages.

#### 4.2.7. Error After Synchronization

Figure 9 illustrates how errors grow after time synchronization completes with the three different schemes, namely, MU-Sync, D-Sync and DE-Sync. From this figure, we can see that DE-Sync works much better than MU-Sync and D-Sync, which achieves a more accurate skew, which affects the degree of synchronization error. MU-Sync assumes that the propagation delay of each round of message exchanges is constant, which causes larger errors. D-Sync ignores the effect of the clock skew when estimating the Doppler scale factor. In other words, when the clock skew is larger, the number of synchronization errors rises notably. As described before, DE-Sync corrects the shortcomings of the other schemes, which is the best among all schemes.

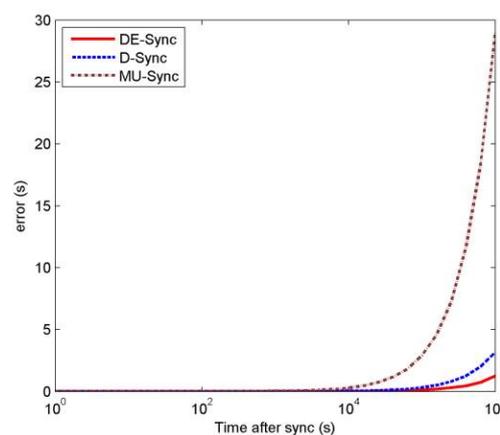


Figure 9. Clock error vs. time after sync.

#### 4.2.8. Comparison of Energy Efficiency

Figure 10 compares the energy efficiency of the three synchronization approaches. The energy efficiency is directly affected by message-exchange overhead and computational cost. However, the energy cost in computation can actually be negligible compared with the consumption of energy used to send and receive acoustic signals in underwater sensor networks. So ignoring computational cost, in this paper, we use Equation (20) as an evaluation formula of energy efficiency [29].

$$\rho = \frac{v}{\kappa\mu\gamma} \quad (20)$$

In this equation,  $v$  denotes a period of time after time synchronization completes,  $v$  is set to  $10^6$  s.  $\mu$  and  $\gamma$  are the number of messages used in performing a linear regression and the packet size, respectively.  $\mu$  and  $\gamma$  are set to 24 messages and 40 Bytes, respectively.  $\kappa$  stands for the number of re-synchronization required within a certain duration of  $v$  to keep synchronization error below a certain clock error tolerance that is defined as  $e$ , which is expressed as follows.

$$\kappa = \frac{v}{\left( \frac{\hat{\alpha}e + (\hat{\beta} - \beta)}{\alpha - \hat{\alpha}} \right)} \quad (21)$$

Note that, through observing Equation (21), we can know that the estimation accuracy of the skew determines the frequency of re-synchronization, which affects the energy efficiency. In other words, the scheme that has higher accuracy time synchronization will have higher energy efficiency. From this figure, with respect to D-Sync and MU-Sync, DE-Sync has higher energy efficiency, which saves power.

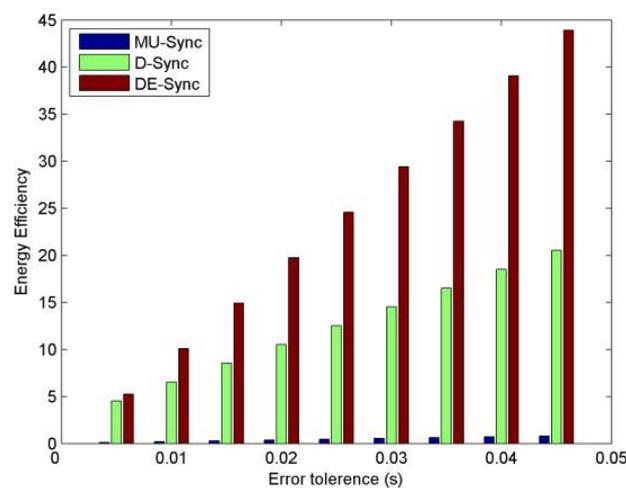


Figure 10. Energy efficiency vs error tolerance.

## 5. Conclusions and Future Work

In this paper, we present DE-Sync, a novel synchronization scheme developed for mobile underwater sensor networks, which is a Doppler-enhanced time synchronization. Compared to the existing schemes, DE-Sync does not directly use the Doppler scale factor to calculate the relative speed between nodes. Meanwhile, it conducts linear regression without using the estimates of sound propagation speed, which reduces errors. Furthermore, it compensates for the effect of the skew when using a Doppler scale factor, which improves the accuracy of time synchronization. Meanwhile, when the initial skew reaches a certain value, the performance of DE-Sync will be significantly better in those conditions. Our simulation results show that DE-Sync is a high-precision time synchronization approach with a low message overhead.

Moreover, we will also explore other schemes to enhance the accuracy of synchronization and to reduce the message overhead for mobile underwater sensor networks.

**Author Contributions:** All authors contributed significantly to the work presented in this manuscript. F.Z., Q.W. and D.N. proposed the new scheme described in this paper; F.Z. and Q.W. completed theoretical deduction; Q.W. and D.N. performed the simulation analysis; G.Q. provided some valuable advice and contribution.

**Funding:** This work is supported by the National Natural Science Foundation of China under Grant No. 61571151. The National Key Research and Development Program of China Grant No. 2017YFC0305702 and National Natural Science Foundation of China under Grant No. 61501134, Grant No. 11774074, Grant No. 61431004.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Akyildiz, I.F.; Pompili, D.; Melodia, T. Underwater acoustic sensor networks: Research challenges. *Ad Hoc Netw.* **2005**, *3*, 257–279. [[CrossRef](#)]
2. Heidemann, J.; Ye, W.; Wills, J.; Syed, A.; Li, Y. Research challenges and applications for underwater sensor networking. In Proceedings of the IEEE Wireless Communications and Networking Conference, Las Vegas, NV, USA, 3–6 April 2006; pp. 228–235. [[CrossRef](#)]
3. Sivrikaya, F.; Yener, B. Time synchronization in sensor networks: A survey. *Netw. IEEE* **2004**, *18*, 45–50. [[CrossRef](#)]
4. Elson, J.; Girod, L.; Estrin, D. Fine-grained network time synchronization using reference broadcasts. In Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Boston, MA, USA, 9–11 December 2002; pp. 147–163. [[CrossRef](#)]
5. Ganeriwal, S.; Ram, K.; Mani, B.S. Timing-sync protocol for sensor networks. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 5–7 November 2003; pp. 138–149. [[CrossRef](#)]
6. Maroti, M.; Kusy, B.; Simon, G.; Ledeczi, A. The flooding time synchronization protocol. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; pp. 39–49. [[CrossRef](#)]
7. Cho, H.; Kim, J.; Baek, Y. Enhanced precision time synchronization for wireless sensor networks. *Sensors* **2011**, *11*, 7625–7643. [[CrossRef](#)] [[PubMed](#)]
8. Shi, F.; Tuo, X.; Yang, S.X.; Li, H.; Shi, R. Multiple two-way time message exchange (ttme) time synchronization for bridge monitoring wireless sensor networks. *Sensors* **2017**, *17*, 1027. [[CrossRef](#)] [[PubMed](#)]
9. Tavares Bruscato, L.; Heimfarth, T.; Pignaton de Freitas, E. Enhancing time synchronization support in wireless sensor networks. *Sensors* **2017**, *17*, 2956. [[CrossRef](#)] [[PubMed](#)]
10. Wang, Z.; Zeng, P.; Zhou, M.; Li, D.; Wang, J. Cluster-based maximum consensus time synchronization for industrial wireless sensor networks. *Sensors* **2017**, *17*, 141. [[CrossRef](#)] [[PubMed](#)]
11. Yoon, S.; Azad, A.K.; Oh, H.; Kim, S. Aupr: An auv-aided underwater routing protocol for underwater acoustic sensor networks. *Sensors* **2012**, *12*, 1827–1845. [[CrossRef](#)] [[PubMed](#)]
12. Khan, J.U.; Cho, H.S. A distributed data-gathering protocol using auv in underwater sensor networks. *Sensors* **2015**, *15*, 19331–19350. [[CrossRef](#)] [[PubMed](#)]
13. Wan, L.; Wang, Z.; Zhou, S.; Yang, T.C.; Shi, Z. Performance comparison of doppler scale estimation methods for underwater acoustic ofdm. *J. Electr. Comput. Eng.* **2012**, *2012*. [[CrossRef](#)]
14. Sharif, B.S.; Neasham, J.; Hinton, O.R.; Adams, A.E. A computationally efficient doppler compensation system for underwater acoustic communications. *IEEE J. Ocean. Eng.* **2000**, *25*, 52–61. [[CrossRef](#)]
15. Yang, T.C. Correlation-based decision-feedback equalizer for underwater acoustic communications. *IEEE J. Ocean. Eng.* **2005**, *30*, 865–880. [[CrossRef](#)]
16. Li, B.; Zhou, S.; Stojanovic, M.; Freitag, L. Non-uniform doppler compensation for zero-padded ofdm over fast-varying underwater acoustic channels. In Proceedings of the Oceans, Aberdeen, UK, 18–21 June 2007; pp. 1–6. [[CrossRef](#)]
17. Li, B.; Zhou, S.; Stojanovic, M.; Freitag, L.; Willett, P. Multicarrier communication over underwater acoustic channels with nonuniform doppler shifts. *IEEE J. Ocean. Eng.* **2008**, *33*, 198–209. [[CrossRef](#)]

18. Johnson, M.; Freitag, L.; Stojanovic, M. Improved doppler tracking and correction for underwater acoustic communications. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, 21–24 April 1997; p. 575. [[CrossRef](#)]
19. Yu, L.; White, L.B. Optimum receiver design for broadband doppler compensation in multipath/doppler channels with rational orthogonal wavelet signaling. *IEEE Trans. Signal Process.* **2007**, *55*, 4091–4103. [[CrossRef](#)]
20. Mason, S.; Berger, C.; Zhou, S.; Willett, P. Detection, synchronization, and doppler scale estimation with multicarrier waveforms in underwater acoustic communication. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 1638–1649. [[CrossRef](#)]
21. Salberg, A.B.; Swami, A. Doppler and frequency-offset synchronization in wideband OFDM. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 2870–2881. [[CrossRef](#)]
22. Ma, L.; Qiao, G.; Liu, S. A combined doppler scale estimation scheme for underwater acoustic OFDM system. *J. Comput. Acoust.* **2015**, *23*, 1540004. [[CrossRef](#)]
23. Liu, S.; Lu, M.; Hui, L.; Chen, T.; Gang, Q. Design and implementation of OFDM underwater acoustic communication algorithm based on omap-l138. In Proceedings of the International Conference on Underwater Networks & Systems, Rome, Italy, 12–14 November 2014; p. 12. [[CrossRef](#)]
24. Ma, L.; Qiao, G.; Liu, S. Heu OFDM-modem for underwater acoustic communication and networking. In Proceedings of the International Conference on Underwater Networks & Systems, Rome, Italy, 12–14 November 2014; p. 14. [[CrossRef](#)]
25. Syed, A.A.; Heidemann, J. Time synchronization for high latency acoustic networks. In Proceedings of the 25th IEEE International Conference on Computer Communications, Barcelona, Spain, 23–29 April 2006; pp. 1–12. [[CrossRef](#)]
26. 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, San Francisco, CA, USA, 15 September 2008; pp. 35–42. [[CrossRef](#)]
27. Liu, J.; Zhou, Z.; Peng, Z.; Cui, J.H. Mobi-sync: Efficient time synchronization for mobile underwater sensor networks. In Proceedings of the Global Telecommunications Conference, Miami, FL, USA, 6–10 December 2010; pp. 1–5. [[CrossRef](#)]
28. 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, Woods Hole, MA, USA, 30 September–1 October 2010; pp. 1–8. [[CrossRef](#)]
29. Liu, J.; Wang, Z.; Peng, Z.; Zuba, M. Tsmu: A time synchronization scheme for mobile underwater sensor networks. In Proceedings of the IEEE Global Telecommunications Conference, Kathmandu, Nepal, 5–9 December 2011; pp. 1–6. [[CrossRef](#)]
30. Xie, P.; Zhou, Z.; Peng, Z.; Cui, J.H.; Shi, Z. SdrT: A reliable data transport protocol for underwater sensor networks. *Ad Hoc Netw.* **2010**, *8*, 708–722. [[CrossRef](#)]
31. Pallares, O.; Bouvet, P.J.; Rio, J.D. Ts-muwsn: Time synchronization for mobile underwater sensor networks. *IEEE J. Ocean. Eng.* **2016**, *41*, 763–775. [[CrossRef](#)]
32. Liu, J.; Wang, Z.; Cui, J.-H.; Zhou, S.; Yang, B. A joint time synchronization and localization design for mobile underwater sensor networks. *IEEE Trans. Mob. Comput.* **2016**, *15*, 530–543. [[CrossRef](#)]

