Research Article

PCDE-Sync: A Time Synchronization Mechanism Based on Partial Clustering and the Doppler Effect for Underwater Acoustic Networks

Jianping Wang (1,2), Jianwei Ma (1), Yikun Feng (3), Qigao Feng (2), Guohong Gao (2), and Yingying Lv (2)

1 School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China
2 Postdoctoral Research Base, Henan Institute of Science and Technology, Xinxiang 453003, China
3 Henley Business School, Whiteknights Campus, University of Reading, Reading, Berkshire, UK

Correspondence should be addressed to Jianwei Ma; lymjw@163.com

Received 10 November 2021; Accepted 7 February 2022; Published 28 March 2022

Academic Editor: Alexander Hošovský

Copyright © 2022 Jianping Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Time synchronization is the basis of coordination and cooperation in underwater acoustic networks. However, because of the propagation delay, node mobility, and Doppler shift, it is impossible to balance the accuracy and energy consumption simply in water. As a promising technology, partial clustering has high convergence and makes breakthroughs in time synchronization. This paper proposes PCDE-Sync, a novel synchronization mechanism with partial clustering and the Doppler effect. Firstly, a clustering method built on the artificial fish swarm algorithm is presented. It models the cluster construction according to fish's preying, swarming, and following behaviors. Secondly, we design a synchronization mechanism to conduct clock correction and compensation by the Doppler effect. Finally, we compare the performance of PCDE-Sync with the most advanced protocols, namely MU-Sync, MM-Sync, and DE-Sync, in terms of the cumulative error after synchronization, the mean square error under different clock skew and that under distinctive node mobility, and energy consumption. The experimental results show that PCDE-Sync makes a trade-off between accuracy and complexity, which does well in solving synchronization issues.

1. Introduction

The underwater acoustic network (UAN) comprises a series of nodes randomly deployed in designated water and communicates by acoustic signals [1]. It is a promising network-based exploration system for underwater applications, such as marine data collection, environmental monitoring, submarine resource survey, auxiliary navigation, and tsunami monitoring [2]. In a UAN, multiple nodes run simultaneously for collaborative processing to gain complete and comprehensive data, i.e., data collected by heterogeneous nodes must be consistent in time. Unfortunately, the local time deviates from the reference clock when the node runs, which is inevitable [3]. Synchronization is the basis of coordination and cooperation, which offsets the difference between the local clock and the reference time base [4, 5].

Because of the discrete deployment of underwater nodes, clustering becomes an essential method for time synchronization in UANs. The clustering algorithm is used to build clusters of different regions. Each group elects a cluster head (CH) in a fixed period. Time synchronization is achieved based on intracluster and intercluster communication [6]. Considering energy and performance, it is necessary to design a lightweight and high-precision synchronization algorithm in underwater systems. However, all nodes participate in electing CHs throughout the survival time, which results in the network varying in a constant state. It is apparent that the cost increases significantly [7], and the survival time reduces. The partial or local clustering is founded on the cluster of the given period. It compares the observation value with the weight factor to quantify whether to perform reclustering. If the observation value is within the
weight factor, it guarantees the stability of the original cluster as much as possible. Therefore, the reclustering of all nodes is no longer mandatory. The movement of nodes is limited. In other words, the status of nodes is predictable [8] at a specific time. All in all, partial clustering is a promising technique.

The artificial fish swarm algorithm (AFSA) is one of the best optimization methods inspired by fish’s social behaviors [9]. It has many advantages, such as good robustness, global searchability, and tolerance of parameter setting. The Doppler effect occurs when waves radiated, reflected, or received by moving objects shift in frequency [10]. In underwater systems, acoustics are employed for communication. The physical properties of underwater acoustic waves include slow propagation speed and frequency-selective signal-to-heat conversion, causing only low-frequency signals to propagate over long distances. Therefore, the underwater acoustic channel is unreliable with small capacity and long delays [11]. The low propagation speed (about 1500 m/s) causes a significant Doppler shift. With this motivation, we try to model underwater nodes as random fish swarms, use AFSA to achieve partial clustering, and build a novel time synchronization mechanism by the Doppler effect, namely PCDE-Sync. The main contributions are summarized as follows:

(1) We model the UAN as multiple fish swarms and propose a partial clustering algorithm based on AFSA.

(2) We design a CH selection mechanism according to the preying, swarming, and following behaviors of the fish.

(3) We present a time compensation and correction method based on the Doppler effect.

To the best of our knowledge, PCDE-Sync is the first work for time synchronization that combines partial clustering with the Doppler effect in UANs. Based on this work, the bottleneck of the UAN is solved. In addition, the distance from a CH to a data node (DN) is less than that to the surface buoy. At the same time, significant energy saving is achieved. Finally, it maintains the stability of the UAN, which is another necessary guarantee. In short, a trade-off between accuracy and complexity is obtained in PCDE-Sync for solving time synchronization issues.

The rest of the paper is arranged as follows: section 2 describes the related work. Section 3 introduces the partial clustering algorithm. Section 4 presents the time synchronization mechanism. Section 5 discusses the simulation experiments. Finally, Section 6 concludes the paper.

2. Related Work

2.1. The Dynamic Synchronization Mechanisms. The nodes are usually mobile and affected by ocean currents in UANs. For dynamic synchronization algorithms, they generally simulate node mobility by a given time base and use the filter tracking methods to correct and compensate the clock deviations [12–14].

Ni et al. [15] proposed K-Sync, a pairwise synchronization algorithm based on the Kalman filter. The simulation shows that K-Sync is robust to various underwater motion scenes. Liu et al. [16] presented a time synchronization method based on clock skew tracking. The simulation results show that the accumulative root mean square error (RMSE) reduces. Wei et al. [17] introduced a clock compensation algorithm based on the BP network. The synchronization compensation algorithm is established for error back-propagation. Simulation experiments show that accuracy increases, and the mean square error reduces. Wang et al. [18] proposed a joint estimation scheme for fine timing, Doppler scaling factor, and carrier frequency offset based on pilot symbols. The method is performed in a block-by-block fashion. Experimental results show decent detection performance for data with relative transceiver motion.

Syed et al. [19] introduced a time synchronization protocol for a high-latency (TSHL) environment. The simulation results demonstrate that it achieves twice the accuracy with minimum energy consumption at 500 meters. Huang et al. [20] proposed an improved synchronization algorithm that combines the mobility of unknown nodes to reduce the clock deviation. The experiment is practical and feasible. Wang et al. [21] studied a dynamic synchronization algorithm with the current model. Experiments reveal the reliability and practicability of the algorithm. Lin et al. [22] considered a motion-based synchronization method (MM-Sync) for underwater sensor networks. The simulation showed that it was a high-precision, low-cost synchronization scheme. Liu et al. [24] conducted a synchronization scheme for mobile UANs. The results show that it achieves higher accuracy with lower overhead. Sumin et al. [25] presented TSUC, a time synchronization algorithm based on the beacon message interval and the skew estimation of root mean square (RMS) in UANs. Experiments show that it confirms accuracy comparable to TSHL but more straightforward skew estimation. Chen et al. [26] presented a synchronization protocol for underwater acoustic networks. The simulation proves that it achieves better performance than TSHL. Liu et al. [27] established a receiver-only synchronization (ROS) model and designed a method to estimate the clock offset and the deviation of active nodes and silent ones. Simulation shows that it has high accuracy and is more robust.

The studies mentioned above reveal the accuracy and reliability of dynamic synchronization methods in given scenarios. Nevertheless, it is still a significant challenge because of the high propagation delay, the estimation model, and the synchronization framework. On the one hand, the cumulative error of these methods is serious. As nodes increase, the accuracy will be significantly reduced [28]. On the other hand, iterative operations are performed and convergence is poor. In addition, they rely heavily on the reference clock and assume that the time base has good performance. However, they often use the crystal oscillator as the clock source. The frequency of oscillators is easily
affected by electronic noise and the underwater environment and is prone to clock drift [29].

2.2. Cluster-Based Synchronization Methods. Clustering algorithms are the basis for hierarchical network management. It improves scalability and dramatically promotes the coverage and connectivity of underwater systems.

Shivaraman et al. [30] proposed C-Sync, a clustering-based time synchronization protocol that provides resilience against faults with energy-efficient communication. Experiment results show that C-Sync detects and isolates faults to a cluster and recovers quickly. Alsolami et al. [31] designed a cluster-based synchronization method for self-synchronized drone networks. The experiment shows that the proposed algorithm has high prediction accuracy. Omeke et al. [32] presented a protocol called distance and energy-constrained k-means clustering scheme (DEKCS) for clustering, cluster-head selection, and data retrieval to prolong the survival time of underwater networks. Evaluations show that DEKCS outperforms the low-energy adaptive clustering hierarchy (LEACH) protocol.

Chirdchoo et al. [33] suggested a cluster-based synchronization algorithm for mobile UANs (Mu-sync) that considers long-term and time-varying propagation delays and estimates clock deviations through linear regression. The simulation results show that the delay estimation determines the accuracy to a large extent. Wu et al. [34] recommended a cluster-based synchronization algorithm. The simulation results show that the combination of cluster topology reduces network flow and improves convergence. Wang et al. [35] proposed a synchronization scheme that includes three algorithms, i.e., threshold-based intracluster time synchronization (ITTS), forwarding-based intercluster time synchronization (FTTS), and one-way intracluster time synchronization (OTTS). The simulation results show that the synchronization scheme reduces the flow rate and improves the convergence. Based on clustering technology, Jia et al. [36] constructed a chained time synchronization protocol for the sensor network (C-TPSN). Experimental results show that the performance of C-TPSN is better than the timing-sync protocol for sensor network (TPSN). Anand et al. [37] studied an underwater synchronization protocol that uses an autoregressive model to transmit data. The simulation results show that it achieves network-wide synchronization when compared with TSHL. Yuan et al. [38] simulated the three-message synchronization protocol in the UAN. The results show that the protocol performs better overall. Kong et al. [39] proposed a dual-channel synchronization mechanism (DCH-sync) that achieves synchronization through a mobile beacon and dual CHs. Xu et al. [40] proposed a cluster-based secure synchronization (CLUS) method. Simulation results show that it decreases errors and reduces synchronization messages.

Although these studies demonstrate the benefits of cluster-based mechanisms, there are still shortcomings. Firstly, they are most concerned with the CH electing and the network stability. Secondly, frequent communication will cause bottlenecks in centralizing CHs. However, there is no ideal time-compensation solution with sparse nodes in UANs. Some dynamic algorithms have no systematic trigger and reclustering methods. Taking into account the cost and stability, the partial clustering algorithm becomes a strategic choice. Unfortunately, few studies have focused on it [41].

2.3. Time Synchronization Based on the Doppler Effect. Because of the low propagation of acoustic signals, even a slight node movement causes a significant Doppler shift in UANs. Therefore, various synchronization methods are recommended based on the Doppler effect.

Yao et al. [42] proposed CD-Sync, a time synchronization algorithm based on the clustering and Doppler velocity measurement in the water. Experimental results show that CD-Sync shortens the distance among nodes and accelerates convergence speed while effectively improving synchronization accuracy. Yang et al. [43] introduced a non-data-aided Doppler estimation method for underwater systems with M-ary spread. The simulation results show that the proposed algorithm is better than the sliding correlation estimation method with average Doppler compensation. Kim et al. [44] offered a sequencing technique for the joint estimation of accurate timing and the cell ID in underwater acoustic systems with a high Doppler effect. A generalized Zadoff-Chu sequence is decomposed into multiple subsequences for reducing the Doppler effect. Lu et al. [45] presented a joint Doppler estimation and compensation method based on spectrum zooming and correction. The simulation results show that the proposed mechanism improves the estimation accuracy and increases the cross-correlation processing gain.

Lu et al. [46] introduced a Doppler shift-based synchronization protocol (D-Sync) for mobile UANs. The simulation results show that D-Sync is superior to existing methods for accuracy and energy consumption. Sidorkina et al. [47] presented a synchronization algorithm using the fractional Fourier transform (FrFT) in UANs. The simulation shows the synchronization possibility under different Doppler factors. Do et al. [48] proposed a synchronization method in OFDM-based underwater systems. Experiments show that the performance of the proposed method is better than the Schmidts algorithm. Kebkal et al. [49] design a synchronization method during payload exchange in UANs. The results show that data exchange is invoked as a comfortable way to the time synchronization. Wang et al. [50] model the hyperbolic frequency modulation (HFM) signal as channel probes for the Doppler estimation. The results show that the synchronization mechanism has better accuracy. Trubuil et al. [51] propose an estimation technique with a dual-training sequence of the Doppler shift. Experiments show that the method is effective in acoustic communication. Balakhonov et al. [52] present a Doppler estimation algorithm for underwater OFDM signals. It considers the multipath propagation of signals with discrete Doppler scales. Li et al. [53] study an estimation algorithm that combines synchronization and the Doppler scale in underwater acoustic communication with the Zadoff-Chu sequences. The experiment proves the effectiveness and
robustness of the method. Zhou et al. [54] established a Doppler-enhanced synchronization (DE-Sync) scheme for mobile underwater sensor networks. The results show that the accuracy and energy efficiency are more significant than the existing synchronization protocol. Pallares et al. [55] designed a synchronization algorithm that considers the Doppler scale estimation in UANs. Liu et al. [56] proposed a cross-layer synchronization scheme for mobile UANs, which uses the Doppler effect and a Kalman filter to estimate the propagation delay. The simulation results show that it obtains a higher accuracy with minimal overhead. At present, we carry out related research on time synchronization based on the Doppler effect in UANs. We construct a synchronization mechanism based on the Doppler effect under the condition of relative motion of a single node and the buoy. Through data analysis, we draw a conclusion that the depth of the nodes has a greater effect on time synchronization [57].

The above studies consider the Doppler estimation and compensation in time synchronization. They verify the effectiveness in given scenarios from different aspects, such as signal quality, capacity, and attenuation. However, they are complex and rely on advanced hardware infrastructures. Also, most of them ignore timeliness. Finally, it is worth mentioning that there are problems in designing underwater nodes with limited energy.

3. A Partial Clustering Mechanism of the Artificial Fish Swarm Algorithm

3.1. A UAN Scenario for Time Synchronization. Suppose a UAN is deployed in a fixed underwater area and each node is equipped with an anchor chain and an anchor. When the node sinks into the water, the anchor chain scales to a certain length according to the depth gauge. A depth sensor and a pressure sensor are settled in the node. As the node is battery-powered, it fails if the energy is exhausted. When all nodes are in place, the surface buoy or the base station (BS) broadcasts a cluster-initial message and triggers the partial clustering mechanism. After a period, the UAN is divided into independent clusters, and each one elects a CH. After that, the time synchronization starts. The communication scene is shown in Figure 1.

Selecting a temporary head (TH) is the first step of the partial clustering algorithm. Next, the electing of CHs is achieved by modeling the preying, swarming, and following behavior of the fish. Messages designed for the clustering and synchronization algorithm are listed in Table 1. The flow of the algorithm is shown in Figure 2.

3.2. Cluster Initialization. There are three roles for a node in the clustering algorithm, namely DN, TH, and CH. DN is the default role. The BS broadcasts a cluster-initial message during the initialization process. The DN receives the message and triggers the clustering algorithm within the communication range.

The threshold of a TH (TH\textsubscript{Temp,CH}) is included in the cluster-initial message, which is settled on BS for electing TH, as shown in the following equation:

\[
T_{\text{Temp,CH}}(n) = \begin{cases} 
\frac{P}{1 - P \times (T_{\text{Round}} \mod 1/P)} & n \in S, \\
0, & \text{others},
\end{cases}
\]  

where \( P \) represents the CH ratio, \( T_{\text{Round}} \) represents the execution round, and \( S \) represents the set of DN. The node that is selected as TH or CH will be signed in each round. In the next turn, it will never act on these roles. \( T_{\text{Temp,CH}} \) is applied to all nodes. The DN that receives the cluster-initial message will generate a random number \( r \) between 0 and 1. Then, it extracts \( r_{\text{Temp,CH}} \) from the message and compares it with \( r \). If \( r > r_{\text{Temp,CH}} \), then the node preserves the DN role and waits to join a relevant cluster. Otherwise, it immediately changes its role to TH and broadcasts a temporary-head-request message. The communication capability (CC), node ID, and timestamp are encapsulated in this message, where CC is measured by the residual energy, CPU utilization, and storage occupancy rate, as shown in the following equation:

\[
C_{C} = W_{1} \times E_{\text{RES}}(i) + W_{2} \times E_{\text{CPU}}(i) + W_{3} \times E_{\text{Storage}}(i),
\]

where \( E_{\text{RES}}(i) \) is the expectation of residual energy, \( E_{\text{CPU}}(i) \) is CPU utilization, and \( E_{\text{Storage}}(i) \) is the occupancy ratio of storage. \( W_{1}, W_{2}, \) and \( W_{3} \) are weight factors.

Followed by this, TH broadcasts a compete–CH–request message. The DN centered on TH will receive this message, extract its CC according to (2), and respond with a compete–CH–response message. The CC and node ID are encapsulated in this message. The destination of the message is to the specified TH.

A DN may be in the cross-region of multiple clusters. At this time, numerous compete–CH–request messages will be received. The node will select the nearest TH according to the CC of the received compete–CH–request messages. The TH with a minor ID is preferred if these messages have the same CC. Then, TH will send a compete–CH–response message. DN will sign this node as its TH, and other nodes will be ignored.

If nodes are sparse in UAN, there may be no DN in the cluster. In this case, the compete–CH–request message sent by TH will not receive any response. Therefore, a timer starts.
when TH sends the compete–CH–request message. If no message is received within the given period, TH will be marked as an orphan node, and it will immediately change its role to DN and add a lock flag. It means that the node no longer participates in CH selection and only joins other clusters by receiving compete–CH–request messages. The pseudocode of the cluster initialization is shown in Algorithm 1.

### 3.3. CH Selection Based on Preying Behavior

Studies have shown that fish can efficiently identify locations and move synchronously through mutual learning. As an optimization method, AFSA is inspired and abstracted by fish behavior. It is shown that AFSA has parallel computing capabilities, high convergence, flexibility, fault tolerance, and can quickly obtain feasible solutions.

In the initialization phase, multiple clusters are formed, and each cluster is a 3D area with a TH as its center. DN is signed with the ID of TH in the cluster. In a short period, the node status is roughly the same, i.e., there is almost no difference in relative motion. Therefore, the cluster is assumed to be stable for a certain period.

Afterward, it models and simulates the behaviors of fish for electing CHs. The location of the TH is represented as a vector, where $\mathbf{p}_t = [x_t, y_t, z_t]$ is the location that needs to be found. The CH selection based on preying behavior is quantified as the residual energy of THs and their distances to BSs. At first, each CH assigns a location randomly. Assuming that the current location of the TH is $\mathbf{P}_i(t)$, the following location $P_{rand}$ is selected within its sensing range, as shown in the following equation:

$$P_{rand} = P_i(t) + \text{Ran}( \cdot ) \times \text{Visual},$$

where Visual is the visibility range.

If the residual energy of the nearest DN is greater than TH, the node is selected as a preying-based CH, namely $\text{CH}_{prey}$. At this time, the progressive location of TH is shown in (4).

$$P_i(t + 1) = P_i + \text{Rand}( \cdot ) \times \text{CH}_{\text{Step}} \times \frac{P_j - P_i}{\|P_j - P_i\|}$$

where $\text{CH}_{\text{Step}}$ means the forwarding step, and $\|P_j - P_i\|$ is the Euclidean distance between $P_j$ and $P_i$.  

### Table 1: The messages are defined for the clustering and synchronization algorithm.

| Message name                  | Sender  | Receiver | Description                                      |
|-------------------------------|---------|----------|--------------------------------------------------|
| Cluster-initial               | BS      | DN       | To trigger the partial clustering                |
| Compete–CH–request            | TH      | DN       | TH requests to compete for the CH                |
| Compete–CH–response           | DN      | TH       | DNs respond to the request of TH                  |
| Statistics-nodes-request      | CH      | DN       | The CH requests to statistic DNs                  |
| Statistics-node-response      | DN      | CH       | DNs response statistic requests                    |
| Node-location-request         | CH      | DN       | CH requests the location of DNs                    |
| Node-location-response        | DN      | CH       | DNs respond to the location                        |
| Residual-energy-request       | CH      | DN       | CH requests the energy of DNs                      |
| Residual-energy-response      | DN      | CH       | DNs respond to the residual energy                 |
| Cluster-statistics            | CH      | BS       | The CH send the cluster-statistics                 |
| Inter-sync-trig               | BS      | CH       | To trigger the synchronization                     |
| Inter-sync-req                | CH      | BS       | CHs request synchronization                        |
| Inter-sync-ack                | BS      | CH       | BS acknowledges synchronization                    |
| Inter-sync-req                | DN      | CH       | The intracluster synchronization                    |
| Inter-sync-ack                | CH      | DN       | DNs acknowledge synchronization                    |

### Figure 2: The flow of the partial clustering mechanism.
Otherwise, the TH continues to look for another random location, as shown in the following equation:

$$P_i(t + 1) = P_i(t) + \text{Ran}() \times \text{Step}.$$  \hspace{1cm} (5)

If the distance from $$P_i(t + 1)$$ to BS is less than that to TH and the residual energy is greater than TH, then it is elected as CH\_prey. If no DN matches the condition, the TH randomly moves to the next step and marks itself as CH\_prey. After selecting CH\_prey, it broadcasts a statistics-node-request message, and the relevant DNs reply to it with a statistics-node-response message and mark the CH\_prey. Finally, CH\_prey registers the cluster to a designated BS. If no node corresponds to the condition, it returns to the process of cluster initialization. The pseudocode of CH selection based on preying behavior is shown in Algorithm 2.

3.4. CH Selection based on Swarming Behavior. Multiple CHs are elected based on preying behavior in the previous stage. A CH is registered on BS in each iteration to update its status to obtain the network’s global view. In a cluster, a DN sends data to a specified CH\_prey. CH\_prey needs to receive, aggregate, verify, process, and transmit data to a corresponding BS. Therefore, the energy consumption of CH\_prey is rapid. Once the energy is exhausted, CH\_prey fails, and the communication will be interrupted. A threshold $$E_r$$ is settled on CH\_prey to balance energy consumption. If the residual energy of CH\_prey is lower than $$E_r$$, i.e., $$E_{CH_{\text{prey}}} < E_r$$, it will select another CH and name it as CH\_swarm.

In the cluster, CH\_prey broadcasts a node-location-request message. DNs respond to a node-location-response message. Therefore, it is easy to gain the center location $$P_C$$, as shown in the following equation:

$$\begin{align*}
X_C & = \frac{1}{n} \sum_{i=1}^{n} x_i, \\
Y_C & = \frac{1}{n} \sum_{i=1}^{n} y_i, \\
Z_C & = \frac{1}{n} \sum_{i=1}^{n} z_i
\end{align*}$$  \hspace{1cm} (6)

The average residual energy $$E_{\text{average}}$$ of DNs is shown in the following equation:

$$E_{\text{average}} = \frac{1}{n} \sum_{i=1}^{n} \text{Node}_i.$$  \hspace{1cm} (7)

At this time, CH\_prey chooses the DN closest to $$P_C$$, the residual energy greater than $$E_{\text{average}}$$ as the next CH, namely CH\_swarm, and inferences the improved location as shown in the following equation:

$$P_i(t + 1) = P_i(t) + \text{Ran}() \times \text{Step} \times \frac{P_C - P_i(t)}{||P_C - P_i||}$$  \hspace{1cm} (8)

After selecting the CH\_swarm, it broadcasts a statistics-node-request message, and the relevant DNs respond to it with a statistics-node-response message and mark the CH\_swarm. Finally, it registers the cluster to a designated BS. If no node corresponds to the condition, it returns to the process in the previous stage. The pseudocode of the CH selection based on swarming behavior is shown in Algorithm 3.
3.5. CH Selection Based on Following Behavior. After selecting CHs by the swarming behavior, CHswarm is located in the center of a cluster. However, the topology may change in the next iteration. Therefore, CHswarm may no longer be the center of the cluster. CHswarm broadcasts a residual-energy-request message within a given period to detect cluster changes. DNs in the cluster respond to a residual-energy-response message. CHswarm unpacks the message that it receives. The threshold for selecting CHs as \( T_{CH_{follow}} \) based on the following behavior is given, and assume that the number of residual-energy-response messages received by the CHswarm is \( \chi \). If \( \chi < T_{CH_{follow}} \), it means that the cluster has changed a lot. In this case, it is necessary to find the next optimal CH, i.e., CHfollow, according to the following behavior. The residual energy, the node ID, and the location information are encapsulated in the residual-energy-response message.

The CHswarm, firstly, finds a DN with enormous residual energy by processing the residual-energy-response message it receives. Then, it takes this node as the center and calculates other nodes within its communication radius. Suppose that the number of DNs centered on CH is \( N(i)_{count} \). \( T_{crowd} \) represents the congestion factor. If \( N(i)_{count} < T_{crowd} \) and the location is closest to CHswarm, the node will be selected as CHfollow, and the location will be updated, as shown in the following equation:

\[
\text{Algorithm 2: CH selection based on the preying behavior}
\]

\[
\text{Algorithm 3: CH selection based on swarming behavior}
\]
\[ P_i(t + 1) = P_i(t) + \text{Ran}(\text{Step}) \times \frac{P_{\text{Max}} - P_i(t)}{P_{\text{Max}} - P_i}, \quad (9) \]

where \( P_{\text{Max}} \) represents location, the residual energy of which is the largest.

After selecting \( CH_{\text{follow}} \), it broadcasts a statistics-node-request message, and the relevant DNs respond to it with a statistics-node-response message and mark \( CH_{\text{follow}} \). Finally, the \( CH_{\text{follow}} \) registers the cluster to a designated BS. If no node corresponds to the condition, it returns to the process in the previous stage. The pseudocode of the CH selection based on the following behavior is shown in Algorithm 4.

### 4. A Time Synchronization Method Based on the Doppler Effect

#### 4.1. The Procedure of Synchronization Mechanism

After a cluster is established, \( CH \) transmits the statistics-node-request message to the DNs of its domain. The ID of \( CH \) is encapsulated in this message. Since DNs are registered with \( CH \) when the cluster is formed, they alone would receive this message. Then, the corresponding DNs reply to the statistics-node-response message, which records the statistics of DNs. \( CH \) aggregates statistics-node-response messages and sends a cluster-statistics message to BS. Followed by this, it establishes the clock correction and compensation based on the Doppler effect. The procedure mainly includes two phases, i.e., intercluster synchronization and intracluster synchronization. A schematic diagram of the synchronization mechanism is shown in Figure 3.

#### 4.1.1. Intercluster Synchronization Phase

The intercluster synchronization is performed to gain the reference clock base. Firstly, BS broadcasts an intersync-trig message to trigger the synchronization process, and the local clock is encapsulated in this message. Multiple CHs receive intersync-trig messages and send intersync-req messages to BS. After receiving the intersync-req message, BS responds to an intersync-ack message. The intercluster synchronization is achieved by the interaction of intersync-req messages and intersync-ack messages. In this process, when \( CH \) receives the intersync-trigger message, it, firstly, records the signal transmission time and its local clock and estimates the propagation delay to correct the clock skew in the intercluster synchronization phase.

#### 4.2. Time Correction and Compensation Based on the Doppler Effect

If there is motion between the wave source and the observer, the Doppler effect will occur. In UANs, even tiny motions will cause a noticeable Doppler effect. Nevertheless, it provides a novel way for time synchronization. Let \( F_s \) be the frequency of the wave source, \( \lambda \) be the wavelength, \( V_w \) be the wave velocity, \( V_o \) be the velocity of the observer, and \( V_s \) be the velocity of the wave source. Assume that \( V_w, V_o, \) and \( V_s \) have directivity. In the Doppler effect, there are three situations in which the observer is stationary, the wave source is static, and the wave source and the observer move relative to each other.

In the first case, when the wave source is close to a stationary observer, the wave velocity is \( V_w - V_o \), and the wavelength is \( \lambda' = \frac{V_w}{V_w} \). However, the observer recognizes that the wave velocity is \( V_w \) and the wavelength is \( \lambda \). Therefore, the observer gets the frequency \( F = \frac{V_w}{\lambda'} \), as follows:

\[ F = F_s \times \frac{V_w}{V_w - V_s}. \quad (10) \]

In the second case, the observer treats the velocity as \( V_w - V_o \) and the wavelength as \( \lambda = \frac{V_w}{F_s} \). Therefore, the frequency is \( F = \frac{V_w}{V_w - V_o/\lambda} \), as shown in the following equation:

\[ F = F_s \times \frac{V_w - V_o}{V_w}. \quad (11) \]

In the third case, the observer and the wave source are close to each other. Therefore, the frequency is as follows:

\[ F = F_s \times \frac{V_w - V_o}{V_w - V_s}. \quad (12) \]

Assume that the horizontal distance between a CH and the buoy is \( d \) in the beginning, and the CH is at depth \( h \). Let \( V_o \) represent the moving velocity. \( F_s \) is affected by the Doppler effect. The horizontal angle \( \theta \) will affect the frequency \( F(t) \) if the wavelength remains the same, as shown in Figure 4.

Therefore, the frequency \( F(t) \) can be derived, as shown in the following equation:

\[ F(t) = F_s \times \frac{V_w + V_o \cos \theta(t)}{V_w}. \quad (13) \]

where \( F(t) \) is related to \( \cos \theta(t) \). If \( CH \) is infinitely far from the buoy, then \( \theta = 0 \) and \( \cos \theta(t) = 1 \). Therefore, the maximum frequency \( F_{\text{max}} \) is shown in the following equation:

\[ F_{\text{max}} = F_s \times \frac{V_w + V_o}{V_w}. \quad (14) \]

If \( CH \) is orthogonal to the buoy, the intersection angle is \( \theta = \pi/2 \) and \( \cos \theta(t) = 0 \). The frequency in this case is \( F_s \). If \( CH \) is located infinitely behind the buoy, then \( \cos \theta(t) = -1 \).
Therefore, the minimum frequency $F_{\text{min}}$ is shown in the following equation:

$$F_{\text{min}} = F_s \times \frac{V_w - V_o}{V_w}. \quad (15)$$

$T_0$ is given as the detection time when the frequency is close to $F_s$. Therefore, CH reaches just below the buoy at $T_0$, and $F(t)$ is shown in the following equation:

$$F(t) = F_s \times \frac{V_w - V_o \times V_o \times T_c / \sqrt{(V_o \times T)^2 + h^2}}{V_w}. \quad (16)$$

The difference between $F(t)$ and $F_{\text{min}}$ is below the threshold $F_o$. It must comply with the following:

$$F(t) \leq F_{\text{min}} + F_o. \quad (17)$$

Therefore, it solves the range of $T_c$ from (15), (16), and (17). The result is as follows:

$$T_c \geq \frac{1}{V_o} \times \frac{(1 - V_w \times F_s / V_o \times F_o)^2 \times h^2}{1 - (1 - V_w \times F_o / V_o \times F_s)^2}. \quad (18)$$

By finding the minimum time $T_c$, it solves $F_{\text{min}}$, which is closest to $F_{\text{min}}$. Assuming that $n$ dataset has been settled, then $F_{\text{min}}$ is defined in (19).

$$F_{\text{min}} = \frac{1}{n} \sum_{i=1}^{n} F_i(t). \quad (19)$$

After that, it estimates $V_o$ as shown in (20).

$$V_o = V_w \times \left(1 - \frac{F_{\text{min}}}{F_s}\right). \quad (20)$$
When $t = 0$, CH is located at distance $d$ in front of the buoy. If $t = T_0$, CH just passes the buoy and moves at the velocity of $V_0$ during this period. Suppose that $V_0$ is the closest to $V_0$, it is clear that $d$ is the closest to $d$, as shown in the following:

$$\bar{d} = \bar{V}_0 \times T_0. \quad (21)$$

By substituting $\bar{V}_0$ in (13), we get $Z$, which calculates the detection point, as shown in the following equation:

$$Z = \frac{V_w \times (\bar{F}(t) - F_i)}{V_0 \times F_i}. \quad (22)$$

Next, we define $Z$ as $\cos \theta'(t)$, and the intersection angle is easily derived. As there will be some detection errors when $Z$ exceeds the interval $[-1, 1]$, $\cos \theta'(t)$ is defined as follows:

$$\cos \theta'(t) = \begin{cases} 1 & Z > 1 \\ Z & -1 \leq Z \leq 1 \\ -1 & Z < -1 \end{cases} \quad (23)$$

According to the trigonometric function, it transforms $\cos \theta'(t)$ into $\tan \theta(t)$, as shown in the following equation:

$$\tan \bar{\theta}(t) = \frac{\sqrt{1 - \cos^2 \theta'(t)}}{\cos \theta'(t)} \quad (24)$$

The relationship between distance $d$ and depth $h$ is expressed as follows:

$$\tan \bar{\theta}(t) = \frac{h}{d - \bar{V}_0 \times t}. \quad (25)$$

By substituting (21) in (25), $\tan \bar{\theta}(t)$ is expressed as follows:

$$\tan \bar{\theta}(t) = \frac{h}{\bar{V}_0 \times T_0 - \bar{V}_0 \times t} = \frac{h}{\bar{V}_0} \times \frac{1}{\bar{V}_0 - \bar{V}_0 - t}. \quad (26)$$

Assume that $x = 1/T_0 - t$. Then, (26) can be rewritten as follows:

$$\tan \bar{\theta}(x) = \frac{h \bar{V}_0}{\bar{V}_0} \times x. \quad (27)$$

It is a linear relationship between $\tan \bar{\theta}(x)$ and $x$ in (27). The coefficient $h/\bar{V}_0$ is solved by removing the meaningless point $t = T_0$ and substituting $\tan \bar{\theta}(x)$ in (27).

If CH receives the synchronization message, it, firstly, records the start time $T_b$ when BS transmits signals. At the same time, CH records its own time $T_s$. Then, it solves $\bar{V}_0, \bar{d}$, and $h$. Therefore, the propagation delay $T_d$ can be obtained as shown in the following equation:

$$T_d = \frac{\sqrt{\bar{d}^2 + h^2}}{V_w}. \quad (28)$$

As CH is constantly in motion, the propagation delay will change. When $T_d$ is solved, the clock offset between BS and CH is counted as follows:

$$\Delta T = T_b - T_s + T_d. \quad (29)$$

Finally, it adds $\Delta T$ to the system time and completes the synchronization process.

5. Simulation Experiments

In this paper, we construct a UAN scenario in WOSS-NS3, in which 100 nodes are randomly deployed. We compare MU-Sync, MM-Sync, and DE-Sync with PCDE-Sync in the experiment because these algorithms coexist with PCDE-Sync. For MU-Sync, it is the first cluster-based synchronization algorithm. For MM-Sync, it is a representative synchronization method. DE-Sync is a Doppler-enhanced synchronization scheme. The main experimental parameters are shown in Table 2.

5.1. Cumulative Error after Synchronization. In a UAN, underwater nodes obtain synchronized clocks from reference time bases. UAN based on a hierarchical structure usually uses clustering and other methods to build local synchronization, and then gradually expands to the whole system. The cumulative error is usually measured by the time difference between the local node and the reference clock. For PCDE-Sync, each node maintains a physical clock and a global logical clock. The cumulative error of PCDE-Sync mainly occurs when CH requests to establish synchronous communication with the surface buoy, the exchange of synchronization among CHs, and the synchronization between DNs and CHs based on intracluster communication. The cumulative error is shown in the following equation:

$$C_{i}^{*} = C_{i}^{local} - C_{ref}^{i} - C_{i}^{CH}. \quad (30)$$

where $C_{i}^{local}$ represents the local clock of a DN, $C_{i}^{ref}$ means the reference clock, and $C_{i}^{CH}$ is the local clock of the corresponding CH.

Therefore, for an underwater sensor network with $m$ nodes, the average clock skew and offset can be expressed as follows:

$$\alpha_{skew} = \frac{\sum_{i=1}^{m} (C_{i}^{*} - C_{i}^{ref}) (C_{i}^{ref} - \bar{C}_{ref})}{\sum_{i=1}^{m} (C_{i}^{*} - C_{i}^{ref})^2}, \quad (31)$$

$$\beta_{offset} = \bar{C}_{ref} - \alpha_{skew} \times \bar{C}_{i},$$

where $\bar{C}_{i}$ is the average value of the local time and $\bar{C}_{ref}$ is the average value of the reference clock.

Figure 5 shows the cumulative error of the four algorithms after synchronization, which appears with an increasing trend. When the time is before $10^3$, the cumulative error of the four algorithms is approximately the same. When the time is after $10^4$, the cumulative error of MM-Sync and MU-Sync varies quickly. However, those of DE-Sync and PCDE-Sync are flatter. Overall, the cumulative error of PCDE-Sync is about 4.121% smaller than that of MU-Sync, MM-Sync (8.79%), and DE-Sync (1.45%). When the time is after 106, the cumulative error of PCDE-Sync is reduced by
28.3% compared to MU-Sync, MM-Sync (16.1%), and DE-Sync (5.19%).

MU-Sync uses a half-round time to determine the propagation delay, and the synchronization error is the largest. For MM-Sync, it uses the Doppler factor to estimate the relative motion, and the synchronization error is lower than that of MU-Sync. Linear frequency modulation (LFM) and the OFDM symbols are used as the preamble for the initial Doppler scale estimation in DE-Sync. For PCDE-Sync, the partial clustering mechanism is implemented in ASFA. Thus, DNs only exchange with the elected CH, significantly reducing the communication range. In addition, the distances between CHs and BSs are quantified by the Doppler effect, which is beneficial for correcting synchronization errors. Therefore, the cumulative error of PCDE-Sync is the lowest.

5.2. MSE Comparison under Different Clock Skew. After the clustering is finished, CH processes the difference between the local clock and the time base. The initial frequency deviation varies from 10 PPM (parts per million) to 80 PPM in this experiment. We see that the MSE of the four algorithms has a rising trend with the increase in frequency skew, as shown in Figure 6.

Within the skew range, the MSE change of MU-Sync is 35.30%, that of MM-Sync is 27.32%, that of DE-Sync is 13.38%, and that of PCDE-Sync is 8.76%. Among the four algorithms, MU-Sync has the largest MSE, and PCDE-Sync has the smallest one. For MM-Sync, the frequency offset is reduced by the linear fitting. For MU-Sync, it estimates the clock deviation by performing linear regression. Consequently, the clock deviation is lower than that of MM-Sync. For DE-Sync, it estimates the clock offset and offset by substituting the Doppler scale factor into the linear regression. For PCDE-Sync, it forms partial clustering and realizes the clock offset correction and compensation through the Doppler effect in a relatively small region. Therefore, the accuracy is higher than that of DE-Sync.

5.3. MSE Comparison under Differential Node Mobility. Since nodes typically exhibit a certain degree of mobility, their clocks may drift if they are inconsistent because of the changes in water. Therefore, MSE fluctuations may occur. Figure 7 shows MSE comparison under distinctive node mobility. It is worth mentioning that the MSE of the four algorithms fluctuates. In general, the MSE of MU-Sync is about 0.013 s, that of MM-Sync is 0.011 s, that of DE-Sync is 0.009 s, and that of PCDE-Sync is 0.005 s.

For MU-Sync, it considers the relative movement in the two-stage operation, namely the skew and offset acquisition stage and the synchronization stage. For MM-Sync, it...
establishes a mobile model to analyze the location change and propagation delay, and then, it solves the synchronization issue using one-way communication. For DE-Sync, it substitutes the Doppler scale factor into the linear regression directly to estimate the clock drift and offset. However, partial clustering is achieved in PCDE-Sync, and the stability of clusters is reached. The clock correction and compensation are performed by the Doppler effect in a small region. Consequently, the synchronization accuracy is higher.

5.4. The Comparison of Energy Consumption. Energy consumption is an essential issue to be considered in UANs. Some synchronization algorithms rely on high-performance hardware, which will increase the development cost and quickly exhaust the energy. Energy consumption mainly depends on complexity. Figure 8 shows the energy consumption of the four algorithms. When the number of nodes is less than 30, the energy consumption of the four algorithms tends to be the same. However, when nodes increase, the energy consumption will also increase. Among them, those of MU-Sync and DE-Sync change exponentially. When the number is 100, the energy consumption of MU-Sync is as high as 45.3%, that of DE-Sync is 39.7%, that of MM-Sync is 28.6%, and that of PCDE-Sync is 22.9%.

For MU-Sync, it performs linear regression to estimate clock deviation and deploys a gradient descent algorithm to solve multifeature regression with a complexity of $O(n^3)$. For DE-Sync, it substitutes the Doppler scale factor to estimate the clock drift and the offset. Therefore, the complexity is $O(mn^2)$, where $m$ is the number of iterations. For MM-Sync, a one-way communication method is adopted, and there is no need to respond to the synchronization request. Thus, the complexity is $O(n^3)$. For PCDE-Sync, it uses AFSA to form partial clustering. Since CH is the time base of a cluster, the electing of CH is limited to the initial cluster, which narrows the communication range. The communication overhead is reduced through intracluster synchronization and intercluster synchronization. Therefore, the complexity is $O(Kn)$, where $K$ is the number of clusters. In summary, the complexity of PCDE-Sync is the lowest.

6. Conclusions

This paper introduces PCDE-Sync, a novel time synchronization algorithm for underwater acoustic networks. Firstly, we propose a partial clustering method and model the CHs electing process based on the preying, swarming, and following behavior of the fish. Secondly, we design the synchronization procedure and realize the clock correction and compensation founded on the Doppler effect. Finally, we compare the performance of PCDE-Sync with that of MU-Sync, MM-Sync, and DE-Sync in terms of the cumulative error after synchronization, the mean square error under different clock skew and that under distinctive node mobility, and the energy consumption. Simulation results show that PCDE-Sync achieves a trade-off of accuracy and energy consumption when solving synchronization issues. In conclusion, PCDE-Sync provides desirable performance for UANs.

Currently, we are prototyping the underwater nodes of PCDE-Sync to build a lightweight UAN system. We have designed a UAN based on the software-defined networking (SDN) architecture [58–61]. It is worth mentioning that the development process of UAN can be significantly reduced, and the experimental construction can be easily realized based on SDN. Next, field deployment and small-scale experiments will be the main work.

Abbreviations

AFSA: Artificial fish swarm algorithm
APE-Sync: Adaptive synchronization
Sync: Synchronization
BS: Base station
CC: Communication capability
CD-Sync: Clustering- and doppler velocity-based synchronization
CH: Cluster head
CL USS: Cluster-based Secure Synchronization
C-Sync: Clustering-based synchronization
C-TPSN: Chained timing-sync for sensor network
DCH-Sync: Distance and energy-constrained K-means clustering scheme
DE-Sync: Doppler-enhanced synchronization
DN: Data node
D-Sync: Doppler-shift-based synchronization
FIT S: Forwarding-based intercluster synchronization
FrFT: Fractional fourier transform
HFM: Hyperbolic frequency modulation
K-Sync: Kalman-filter-based synchronization
LEACH: Low-energy adaptive clustering hierarchy
LFM: Linear frequency modulation
MM- Sync: Motion-based synchronization
MU- Sync: Cluster-based synchronization for mobile UAN
OFDM: Orthogonal frequency-division multiplexing
OT I S: One-way intracluster time synchronization
PPM: Parts per million
RM S: Root mean square
R MSE: Root mean square errors
ROS: Receiver-only synchronization
TH: Temporary head
TTIS: Threshold-based intracluster synchronization
TPSN: Timing-sync for sensor network
TSHL: Time synchronization for high latency
UANs: Underwater acoustic networks.

Data Availability

The data used to support this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Science and Technology Major Project of Xinxiang City under Grant 21ZD003, in part by the Key Scientific and Technological Project of Henan Province under Grant nos. 222102320181, 222102110011, 212102210422, 212102310087, and 202102210388, in part by the Young Scholar Training Program of Higher Education in Henan Province under Grant 2019GGJS172, and in part by the Key Scientific Research Projects of Colleges and Universities in Henan Province under Grant 20A520002, 21A520001.

References

[1] J. Wang, D. Kong, W. Chen, and S. Zhang, “Advances in software-defined technologies for underwater acoustic sensor networks: a survey,” Journal of Sensors, vol. 2019, Article ID 3470390, 13 pages, 2019.
[2] J. Wang, R. Zhao, and W. Gu, “Time synchronization in networks: a survey,” in Proceedings of the International Conference on Control and Computer Vision (ICCCV), pp. 121–126, South Korea, June 2019.
[3] Y. Zhan, W. Lei, and W. Zhang, “Survey on time synchronization technology in networks,” in Proceedings of the Chinese Control and Decision Conference (CCDC), pp. 2157–2161, Qingdao, China, May 2015.
[4] Z. Gong, C. Li, and F. Jiang, “AUV-aided joint localization and time synchronization for underwater acoustic sensor networks,” IEEE Signal Processing Letters, vol. 25, no. 4, pp. 477–481, 2018.
[5] A. Boukerche and P. Sun, “Design of algorithms and protocols for underwater acoustic wireless sensor networks,” ACM Computing Surveys, vol. 53, no. 6, pp. 1–15, 2021.
[6] W. Zhang, G. Han, and X. Wang, “A node location algorithm based on node movement prediction in underwater acoustic sensor networks,” IEEE Transactions on Vehicular Technology, vol. 69, no. 3, pp. 3166–3178, 2020.
[7] A. Toky, R. Singh, and S. Das, “Localization schemes for underwater acoustic sensor networks - a review,” Computer Science Review, vol. 37, pp. 1–18, 2020.
[8] Z. Jin, M. Ding, Y. Luo, and S. Li, “Integrated time synchronization and multiple access protocol for underwater acoustic sensor networks,” IEEE Access, vol. 7, pp. 101844–101854, 2019.
[9] N. Mehdi, S. epidnam, G. Sargolzaei, and N. T. Adel, “Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications,” Artificial Intelligence Review, vol. 42, no. 4, pp. 965–997, 2014.
[10] Z. Patel and K. Rana, “A review on time synchronization free localization schemes in underwater wireless sensor networks,” in Proceedings of the International Conference on Trends in Electronics and Informatics (ICTEI), pp. 330–335, Tirunelveli, India, May 2018.
[11] S. Jiang, “Networking in oceans: a survey,” ACM Computing Surveys, vol. 54, no. 1, pp. 1–33, 2022.
[12] F. Shi, X. Tuo, S. Yang, H. Li, and J. Lu, “Rapid-flooding time synchronization for large-scale wireless sensor networks,” IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 1581–1590, 2022.
[13] K. Yildirim and A. Kantarci, “Time synchronization based on slow-flooding in wireless sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 1, pp. 244–253, 2014.
[14] D. Huang, W. Teng, and K. Yang, “Secured flooding time synchronization protocol with moderator,” International Journal of Communication Systems, vol. 26, no. 9, pp. 1092–1115, 2013.
[15] X. Ni, T. Lu, and S. Ye, “Pair nodes clock synchronization algorithm based on kalman filter for underwater wireless sensor networks,” Sensors, vol. 21, no. 13, pp. 1–13, 2021.
[16] D. Liu, M. Zhu, D. Li, and X. Fang, “Energy-efficient time synchronization based on nonlinear clock skew tracking for underwater acoustic networks,” Sensors, vol. 21, no. 15, pp. 1–13, 2021.
[17] L. Wei, H. Li, D. Wu, and Y. Guo, “Clock synchronization error compensation algorithm based on BP neural network model,” Acta Physica Sinica, vol. 70, no. 11, pp. 1–11, 2021.

[18] Y. Wang, J. Tao, and L. Ma, “Joint timing and frequency synchronization for OEDM underwater acoustic communications,” in Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC), pp. 272–277, Xiamen, China, July 2021.

[19] A. Syed and J. Heidemann, “Time synchronization for high latency acoustic networks,” in Proceedings of the IEEE INFOCOM, pp. 892–903, Barcelona, Spain, April 2006.

[20] H. Huang, X. Jiang, and W. Su, “The passive hidden mobile node time synchronization algorithm used in underwater acoustic networks,” in Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 1–5, Xiamen, China, October 2017.

[21] S. Wang and M. Gao, “Current model and Doppler effect-based time synchronization algorithm,” in Proceedings of the IEEE International Conference on Computer and Communications (ICCC), pp. 326–330, Chengdu, China, December 2017.

[22] J. Lin, G. Feng, X. Wen, and H. Lv, “MM-sync: A Mobility Built-In Model-Based Time Synchronization Approach for Underwater Sensor Networks,” in Proceedings of the BDCloud-SocialCom-SustainCom, pp. 338–345, Atlanta, GA, October 2016.

[23] F. Zhou, Q. Wang, G. Han, G. Qiao, Z. Sun, and A. Niaz, “APE-Sync: an adaptive power efficient time synchronization for mobile underwater sensor networks,” IEEE Access, vol. 7, Article ID 52389, 2019.

[24] J. Liu, Z. Zhou, and Z. Peng, “Mobi-Sync: efficient time synchronization for mobile underwater sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 2, pp. 406–416, 2013.

[25] J. Sumin and S. C. Kim, “Time synchronization for underwater cellular networks,” in Proceedings of the International Conference on Information Networking (ICIN), pp. 512–517, Da Nang, Vietnam, May 2017.

[26] Z. Chen, H. Chen, and W. Xu, “Simplified time synchronization for underwater acoustic sensor networks with high propagation latency,” in Proceedings of the Oceans, pp. 1–5, Taipei, October 2014.

[27] G. Liu, S. Yan, and L. Mao, “Receiver-only-based time synchronization under exponential delays in underwater wireless sensor networks,” IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9995–10009, 2020.

[28] S. Freeman, L. Emokpae, and G. Edelmann, “High-frequency, highly directional short-range underwater acoustic communications,” in Proceedings of the Oceans - MTS/IEEE Washington, pp. 1–4, Washington, DC, 2015.

[29] A. Gardner and J. Collins, “Advancements in high-performance timing for long term underwater experiments a comparison of chip scale atomic clocks to traditional microprocessor-compensated crystal oscillators,” in Proceedings of the MTS/IEEE Oceans Conference, pp. 1–8, Virginia Beach, VA, 2012.

[30] N. Shivarame, P. Schuster, S. Ramathan, A. Easwaran, and S. Steinhorst, “Cluster-based network time synchronization for resilience with energy efficiency,” in Proceedings of the IEEE Real-Time Systems Symposium, pp. 149–161, Dortmund, Germany, December 2021.

[31] F. Alsolami, F. A. Alqurashi, M. K. Hasan, R. A. Saeed, S. Abdel-Khalek, and A. Ben Ishak, “Development of self-synchronized drones’ network using cluster-based swarm intelligence approach,” IEEE Access, vol. 9, pp. 48010–48022, 2021.

[32] K. Omeke, M. Moll, M. Ozturk, and S. Ansari, “DEKCS: a dynamic clustering protocol to prolong underwater sensor networks,” IEEE Sensors Journal, vol. 21, no. 7, pp. 9457–9464, 2021.

[33] N. Chiridchoo, W. Soh, K. C. Chua, and Mu-Sync, “A time synchronization protocol for underwater mobile networks,” in Proceedings of the Workshop on Underwater Networks (WuWNet), pp. 1–8, San Francisco, California, USA, September 2008.

[34] J. Wu, L. Zhang, Y. Bai, and Y. Sun, “Cluster-based consensus time synchronization for wireless sensor networks,” IEEE Sensors Journal, vol. 15, no. 3, pp. 1404–1413, 2015.

[35] H. Wang, D. Xiong, L. Chen, and P. Wang, “A consensus-based time synchronization scheme with low overhead for clustered wireless sensor networks,” IEEE Signal Processing Letters, vol. 25, no. 8, pp. 1206–1210, 2018.

[36] X. Jia, Y. Lu, and X. Wei, “Improved time synchronization algorithm for wireless sensor networks based on clustering,” in Proceedings of the IEEE 8th Joint International Information Technology and Artificial Intelligence Intelligence Conference (ITAIC), pp. 1211–1215, Chongqing, China, May 2019.

[37] J. Anand, K. Sundeep, and N. Rao, “Underwater sensor protocol for time synchronization and data transmissions using the prediction model,” in Proceedings of the International Conference on Inventive Computation Technologies (ICICT), pp. 762–766, Coimbatore, India, February 2020.

[38] Y. Yuan, F. Zhou, and Y. Fu, “Simulation of time synchronization protocol of underwater acoustic communication network with cluster topology,” in Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 1–4, Xiamen, China, October 2017.

[39] W. Kong and G. Liu, “Cluster-based time synchronization algorithm using dual cluster heads for underwater sensor network,” Computer Engineering, vol. 46, no. 2, pp. 214–220, 2020.

[40] M. Xu, L. Guangzhong, and Z. Daqi, “A Cluster-Based Secure Synchronization Protocol for Underwater Wireless Sensor Networks,” International Journal of Distributed Sensor Networks, pp. 1–13, 2014.

[41] D. N. Sandeep and V. Kumar, “Review on clustering, coverage and connectivity in underwater wireless sensor networks: a communication techniques perspective,” IEEE Access, vol. 5, pp. 11176–11199, 2017.

[42] Y. Yao, G. Liu, and W. Kong, “Time synchronization algorithm of underwater sensor based on Doppler velocity measurement,” Computer Engineering, vol. 47, no. 4, pp. 147–152, 2021.

[43] G. Yang, F. Zhou, G. Qiao, and Y. Lu, “Optimized Doppler estimation and symbol synchronization for mobile M-ary spread spectrum underwater acoustic communication,” Journal of Marine Science and Engineering, vol. 9, no. 9, pp. 1–23, 2021.

[44] Y. Kim, M. Asim, T. Im, and Y. Cho, “Sequence design technique for accurate timing and cell ID estimation in underwater acoustic cellular systems with a high Doppler,” Electronics, vol. 10, no. 19, pp. 1–17, 2021.

[45] J. Lu, Q. Zhang, W. Shi, and L. Zhang, “Doppler estimation and compensation method for underwater target active detection based on communication signal,” Journal of Northwestern Polytechnical University, vol. 39, no. 5, pp. 962–970, 2021.
[46] F. Lu, D. Mirza, and C. Schurgers, “D-Sync: Doppler-based time synchronization for mobile underwater sensor networks,” in Proceedings of the Workshop on Underwater Networks (WuWNet), pp. 1–8, Woods Hole, Massachusetts, USA, February 2010.

[47] J. Sidorkina, R. M. Dyabirow, B. Shakhtarin, and A. Bychkov, “A presented chirp synchronization algorithm based on the Fractional Fourier Transform for underwater acoustic communication,” in Proceedings of the Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHRONINFO), pp. 1–4, Russia, July 2019.

[48] D. Do, Q. Nguyen, D. Ha, and A. Bychkov, “A time synchronization method for OFDM-based underwater acoustic communication systems,” in Proceedings of the International Conference on Advanced Technologies for Communications (ATC), pp. 131–134, Hanoi, 2016.

[49] K. Kebkal, V. Kebkal, and O. Kebkal, “Underwater acoustic modems (S2CR Series) for synchronization of underwater acoustic network clocks during payload data exchange,” IEEE Journal of Oceanic Engineering, vol. 41, no. 2, pp. 428–439, 2016.

[50] K. Wang, S. Chen, C. Liu, and Y. Xu, “Doppler estimation and timing synchronization of underwater acoustic communication based on hyperbolic frequency modulation signal,” in Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC), pp. 75–80, Taipei, April 2015.

[51] J. Trubuil, T. Le Gall, and T. Chonavel, “Synchronization, Doppler and channel estimation for OFDM underwater acoustic communications,” in Proceedings of the Oceans, pp. 1–6, Taipei, 2014.

[52] K. Balakhonov, B. I. Shakhtarin, and R. M. Dyabirow, “Experimental research of Doppler scale estimation algorithms for OFDM signals in fast time-varying underwater acoustic channel,” in Proceedings of Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHRONINFO) pp. 1–5, Minsk, 2018.

[53] Y. Li, Y. Wang, and X. Guan, “Joint synchronization and Doppler scale estimation using Zadoff-chu sequences for underwater acoustic communications,” in Proceedings of Oceans, pp. 1–5, Anchorage, AK, June 2017.

[54] F. Zhou, Q. Wang, and D. Nie, “De-Sync: A Doppler-enhanced time synchronization for mobile underwater sensor networks,” Sensors, vol. 18, no. 6, pp. 1–15, 2018.

[55] O. Pallares, P.-J. Bouvet, J. del Rio, and Ts-Muwsn, “TS-MUWSN: time synchronization for mobile underwater sensor networks,” IEEE Journal of Oceanic Engineering, vol. 41, no. 4, pp. 763–775, 2016.

[56] J. Liu, Z. Wang, and M. Zuba, “DA-Sync: a Doppler-assisted time-synchronization scheme for mobile underwater sensor networks,” IEEE Transactions on Mobile Computing, vol. 13, no. 3, pp. 582–595, 2014.

[57] J. Wang and C. Jiao, “Design of A Time synchronization mechanism for underwater acoustic networks,” in Proceedings of the International Conference on Networking and Network Applications (NaNA), pp. 41–46, Haikou City, China, December 2020.

[58] J. Wang, Y. Lv, and W. Chen, “S. M.-Uan: A software-defined underwater acoustic network of multi-controllers for inland waterway systems,” IEEE Access, vol. 8, pp. 211135–211151, 2020.

[59] J. Wang, G. Gao, P. Qu et al., “A software-defined clustering mechanism for underwater acoustic sensor networks,” IEEE Access, vol. 7, pp. 121742–121754, 2019.