Hindawi
Journal of Sensors
Volume 2017, Article ID 5768651, 13 pages
https://doi.org/10.1155/2017/5768651

Research Article

Hybrid Localization Approach for Underwater Sensor Networks

Pei-Hsuan Tsai, Rong-Guei Tsai, and Shiuan-Shiang Wang

Institute of Manufacturing Information and System, National Cheng Kung University, Tainan City 70101, Taiwan

Correspondence should be addressed to Pei-Hsuan Tsai; peihsuan.tsai@gmail.com

Received 1 June 2017; Revised 5 September 2017; Accepted 24 September 2017; Published 2 November 2017

Academic Editor: Hana Vaisocherova

Underwater Wireless Sensor Networks (UWSNs) are widely used to collect data in the marine environment. Location and time are essential aspects when sensors collect data, particularly in the case of location-aware data. Many studies on terrestrial sensor networks consider sensor locations as the locations where data is collected and focus on sensor positioning when sensors are fixed. However, underwater sensors are mobile networks and the sensor locations change continuously. Localization schemes designed for static sensor networks need to run periodically to update locations and consume considerable sensor power and increase the communication overhead; hence, they cannot be applied to UWSNs. This paper presents a hybrid localization approach with data-location correction, called Data Localization Correction Approach (DLCA), which positions data without additional communication overhead and power consumption on sensors. Without loss of generality, we simulate the ocean environment based on a kinematic model and meandering current mobility model and conduct extensive simulations. Our results show that DLCA can significantly reduce communication costs, while maintaining relatively high localization accuracy.

1. Introduction

The ocean is a vast natural resource with areas that are yet to be thoroughly explored. With the development of ocean engineering and network technology, the ocean has become a focus for research, and underwater sensor networks have received considerable attention from researchers. Applications of underwater sensor networks range from the oil industry to aquaculture and include instrument monitoring, pollution control, climate recording, prediction of natural disturbances, search and survey missions, and the study of marine life [1–4]. A node must know its own location before sending data to its neighbor. The need for location arises because the number of nodes is very large and it is not possible for the *base station* to find the nodes' positions, so the individual node is required to send location information along with the observed data to provide exact location to the user, which means the node must localize itself.

Communication and collaboration among nodes are essential to assist node self-localization. In most localization algorithms, nodes collaborate with each other by considering several aspects like limited energy resources, number and density of nodes, and existence of obstacles. Compared to terrestrial sensor networks, underwater sensor networks face new communication challenges: (1) underwater communication systems today mostly use acoustic technology because electromagnetic waves cannot propagate over long distances in water; acoustic communications offer longer ranges, but with large latency, limited bandwidth, and time-varying multipath propagation, and (2) underwater sensor networks are dynamic networks and sensor locations change continuously because of ocean currents. However, observed data is typically interpreted with reference to the sensor's position. In other words, sensor locations are considered as data locations. In such environments, localization schemes designed for static networks need to run periodically to update sensor locations, resulting in communication overheads and sensor power consumption. In a harsh aqueous environment, sensor mobility makes underwater sensor networks the most challenging of all network scenarios.

Although the network conditions in underwater environments make localization difficult, a useful property we found is that in many applications, such as environmental monitoring, sensors periodically report their observed data to the base station. For these applications, location information is only useful at those discrete time points. Therefore, a heuristic approach to reduce communication and energy costs while maintaining the localization accuracy is to set

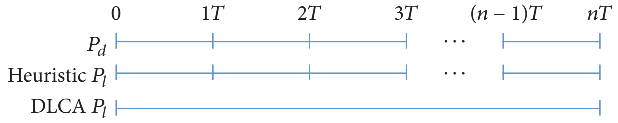


Figure 1: Comparison of period length.

the *sensor self-localized period* (P_l) equal to the *data observed period* (P_d). The top line and middle line in Figure 1 denote the *data observed period* (P_d) and the *sensor self-localized period* (P_l) of the heuristic approach, respectively. However, the downside of the heuristic approach is that when P_d is small, it results in relatively high communication costs being generated. Due to the limited bandwidth of acoustic channels used by underwater sensor networks, this will be a large burden on the network.

Based on this motivation, we aim to design a hybrid localization approach for underwater sensor networks, called *DLCA*, which takes full advantage of computing the data-location at a base station when data packets are received, instead of relying on sensors continuously self-localizing. In other words, DLCA extends P_l to multiples of P_d, as shown by the bottom line in Figure 1. The whole localization process of DLCA is divided into two parts: *node localization* and *observed data-location correction*. Node localization is run by the nodes themselves, and observed data-location correction is run by the base station. Simulation results show that DLCA can greatly reduce localization communication costs while maintaining relative localization accuracy.

This work makes two contributions. First, our work is unique in aiming at localization of location-aware data instead of sensor localization in underwater sensor networks and provides a practical system with hybrid localization. The second contribution is the design of post facto location correction on the base station that improves localization accuracy without additional communication overheads and power consumption on sensor nodes.

The remainder of this paper is organized as follows. In Section 2, we provide background information, including several distance acoustic measurement models, time synchronization, and communication schemes for UWSNs. In Section 3, we review related works. The design of DLCA, including its network architecture, data structure, and fundamental algorithms are presented in Section 4, the simulation results are presented in Section 5, and Section 6 contains our conclusions.

2. Background

Underwater sensor networks comprise a large number of small devices deployed in a physical underwater environment. Each node has special capabilities, such as wireless communication with its neighbors, sensing data, data storage, and processing. Today there are more than 50 localization algorithms in existence [5–7]. These can be classified on the basis of different aspects.

Centralized Localization versus Distributed Localization. In centralized localization, one central base station computes the locations of unknown nodes, while in distributed localization, computation is done by the sensor itself, and nodes communicate with each other to obtain their position in the network. DLCA uses *hybrid* localization. The locations of sensors are computed by the sensors themselves to support real-time monitoring, but the data locations are corrected by a base station to reduce nodes power consumption.

Anchor Based versus Anchorless. The anchor-based algorithms use anchor nodes (which know their own position from prior GPS data) as reference nodes for localization. The more anchor nodes, the higher localization accuracy, but the cost also increases as anchor nodes are equipped with extra resources. The anchorless schemes measure the distance between nodes for creating a local map of the nodes. However, the distance measurement techniques to date have not been accurate, so global coordinates are preferred over local coordinates for most applications. Therefore, DLCA utilizes the anchor-based localization.

Range-Based versus Range-Free. Range-based systems use techniques such as ultrasound to measure the distance between nodes and then triangulation to compute the positions of nodes. Range-free techniques use implicit information provided by anchor nodes to obtain positions of nodes, such as number of hops between devices or radio coverage membership [8–10]. Although range-free protocols do not need additional hardware for distance measurements, they can only provide rough positional estimates. DLCA is more interested in accurate localization and thus DLCA adopts range-based schemes.

Several distance measuring techniques, such as received signal strength indication (RSSI), time of arrival (ToA), and time difference of arrival (TDoA), are widely used in UWSNs for localization. Among them, ToA, in particular, is the most commonly used for UWSNs as it is more accurate [1–4]. It is not affected by channel fading but has an issue in achieving synchronization between nodes. It sends a single packet from one node to another containing the time of its transmission, assuming perfect clock synchronization between nodes. The receiving node knows when the packet arrived and, if it is synchronized with the sender node, the distance travelled can be calculated using the following equation:

$$d_{\text{ToA}} = c * \Delta t_{\text{ToA}}, \tag{1}$$

where c is the speed of sound in seawater ($c = 1500$ m/s) and Δt_{ToA} is the time of arrival (Δt_{ToA} = receiver time − sender time).

However, sound waves can have transmission delays and varying transmission rates. The delays make the ToA distance longer than the actual distance, and the longer the real distance, the more the error deviation of the ToA distance [11–13]. Equations (2) and (3) simulate the difference between actual distance and ToA distance

$$d_{\text{ToA}} = d_{\text{Real}} + \varepsilon, \tag{2}$$

$$\varepsilon = d_{\text{Real}} \times N\left(\mu, \sigma^2\right), \tag{3}$$

where d_{Real} is the real distance between two nodes and ToA distance, d_{ToA}, is the real distance with an error ε simulated by Gaussian distribution calculated in (3).

Synchronized time is the prerequisite of using ToA to estimate distance and therefore many localization algorithms rely on the time synchronization services. A synchronization algorithm for UWSNs must consider additional factors, such as long propagation delays from the use of acoustic communication and sensor node mobility. Many time synchronization approaches have been proposed for different network topologies [14–17]. The approach in [17] is designed for the same network topology as ours. Their approach solves localization and time synchronization jointly to save energy and to improve the accuracy of both services by utilizing the message exchanges among the nodes. We will apply and evaluate the joint scheme, performing the node localization and time synchronization simultaneously at every P_l so that the clock drift can be limited to a tolerable range by setting the length of P_l to similar network scenarios. The results will be reported in a future paper. In this paper, we assume that all nodes are time synchronized.

Unlike terrestrial networks, which mainly rely on radio waves for communication, UWSNs utilize acoustic waves for information exchange. To provide high throughput in an energy-efficient way, it is important for UWSNs to have an efficient Medium Access Control (MAC) protocol that allows the nodes to share a common broadcast channel and to prevent simultaneous transmissions or resolve collisions of data packets, while providing energy efficiency, low channel access delays, and fairness among the nodes. However, the underwater acoustic environment poses difficulties for MAC protocol design, for example, high and variable propagation delay, limited bandwidth and data rate, noise, and energy consumption. In addition, network topology and deployment highly affect the performance of the MAC protocol. Contention-free MAC protocols are not good solutions for our network topology [18] because they are either dedicated to source-to-destination packet exchange without supporting broadcasting or may not yield better performance than random access approaches owing to the long and varying propagation latency of the underwater acoustic channel. In addition, our approach, DLCA, can tolerate packet collision. Although the higher the node density, the higher the probability of collision, as long as there are neighbors, our data-location correction approach is applicable and therefore the more the neighbor nodes, the higher the collision tolerance. In this paper, we assume that the contention-based MAC protocol (e.g., random access) is adopted for communications. In [19], a collision-tolerant packet scheduling was proposed for underwater acoustic localization. They assumed that, during a localization period, the reference nodes transmit randomly, for example, according to a Poisson distribution with an average transmission rate of λ packets per second. We will apply and evaluate the collision-tolerant algorithm described in [19] with our scheme and report the results in a future paper.

3. Related Work

A tremendous amount of localization schemes has been reported for terrestrial WSNs [5–7]. However, the unique characteristics of the underwater sensor network environment, for example, sensor node mobility, long propagation delay, and high power consumption in transmit and receive modes, make the existing WSN algorithms inefficient for UWSNs. To overcome these issues, modifications to WSN localization schemes or different alternatives have been proposed [1–4].

According to [4], UWSN localization techniques can be divided into two categories: centralized and distributed techniques. The main advantage of the centralized technique is reducing the computational burden of the underwater sensor nodes. The major drawback of the scheme is not supporting real-time location information and requiring high communication overhead and high energy consumption in sending localization related messages to an access point, which is underwater or on the surface. In contrast, the main advantage of the distributed technique is to support applications that need real-time location information, for example, online monitoring and coordinated motion. The major drawbacks of the scheme are high energy consumption and the communication and computational burdens on the sensor nodes.

The design of the localization techniques highly depends on the network topology and applications. We describe here the most recent localization techniques with similar network topologies as ours.

3.1. Distributed Localization Techniques. In distributed localization techniques, the sensor nodes need to collect anchor positions, compute the distance to anchors or neighbors, and run location estimation algorithms for self-positioning. LDB [9, 10], AAL [20, 21], DNRL [22, 23], and MobiL [24] focus on how anchors provide their positions to the sensor nodes and assume that the anchors are mobile and equipped with GPS to obtain their locations. Their disadvantages include high localization delays and high cost as the accuracy of the location relies on a large number of mobile beacons, especially for large-scale UWSNs.

Large-Scale Hierarchical Localization (LSHL) explores the localization problem in large-scale UWSNs and provides a hierarchical approach to divide the localization process into anchor and ordinary node localizations [25, 26]. It assumes that an anchor can get perfect location estimation and focus on ordinary node localization. The drawback of LSHL is having a high energy consumption and communication overhead due to beacon exchanges. Many researches [27–30] are proposed to improve LSHL, including SLMP and our work, based on the same network topology and assumptions.

SLMP is a prediction-based localization scheme [27]. The anchor locations are estimated by either using trilateration with surface buoy coordinates or running mobility prediction algorithms. The ordinary nodes use the mobility pattern to predict their locations, and the pattern is assumed valid until an update from an anchor node is received. The approaches in

[28–30] are designed to improve the accuracy of mobility prediction algorithms. However, the communication overhead and energy consumption depend on the mobility pattern for the ordinary node.

3.2. Centralized Localization Techniques. Centralized techniques calculate the location of each sensor node in a command center or sink, and the sensor nodes do not know their locations unless the sink node explicitly sends their information. Therefore, they are not convenient for applications that require accurate and real-time location information. In addition, most centralized techniques are not applicable to large-scale UWSNs; for example, SLMP [27] assumes that data are sent via wired communications.

Collaborative Localization (CL) [31] is anchor-free, and the nodes collaborate to determine their positioning autonomously without using surface buoys or ships. The sensor nodes are categorized as profilers or followers. A profiler travels to a depth first and its trajectory is used as a prediction of the future location of the followers. However, it initially assumes that all nodes are localized by GPS and the coverage and accuracy of localization depend on the trajectories of the followers that have to be the same as the profilers, which are not applicable to large-scale UWSNs.

Furthermore, [32, 33] target applications where the relation between the data and location is resolved at the postprocessing stage by a central station. The sensor node collects distance estimates between itself and its neighbors and then all distance estimations are sent to a central station and processed offline. An iteration algorithm is used to obtain the positions. At each iteration, the algorithm refines the positions of the sensor nodes. The drawbacks are long localization time and huge energy of the sink. In this paper, we propose a vector-based approach to improve these.

In this paper, we propose a hybrid localization approach to support real-time location information when it is needed and to reduce the computational burden on sensor nodes by post facto correcting the location of observed data. The communication overhead and energy consumption depend on the accuracy and promptness of location demanded by the applications. For an acceptable location error tolerance, our approach, DLCA, decreases the number of localization updates and consequently its communication overhead and energy consumption is low.

4. Design of DLCA

In this section, we describe the design of DLCA, a data-location correction approach, for underwater sensor networks. We first present the network architecture and then the algorithms governing the model, followed by an example.

4.1. Network Topology. The DLCA network architecture comprises four types of nodes, as shown in Figure 2: surface buoys, anchor nodes, sensor nodes, and a base station. *Surface buoys* are nodes deployed on the sea surface and equipped with GPS to obtain their absolute location from GPS. The *anchor nodes* communicate with surface buoys and obtain their locations through the GPS intelligent buoy system [34, 35]. Therefore, we assume that the locations of surface buoys and anchor nodes are sufficiently accurate in this paper. Neither surface buoys nor anchor nodes are equipped with sensors, and their main functions are assisting the localizing of sensor nodes and transferring data from sensor nodes. The *sensor nodes* are those nodes that cannot communicate directly with the surface buoys because of distance or other constraints but can communicate with the anchor nodes to estimate their own locations. Through message passing, localized sensor nodes can assist other nonlocalized sensor nodes to estimate their locations [8, 36]. The *base station* is the node to centralize and to integrate all the data transferred by surface buoys.

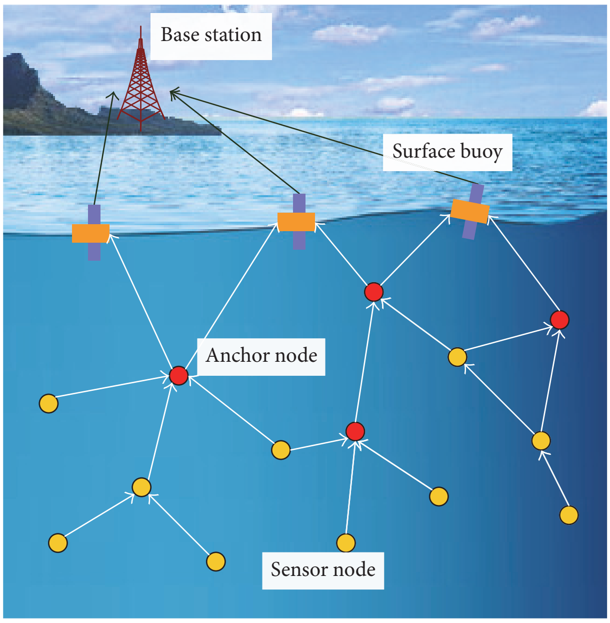


Figure 2: Network architecture of underwater sensor networks.

4.2. Overview of DLCA. Here are the assumptions made in the design of the DLCA.

(1) All nodes are time synchronized initially. Then at every P_l, the time synchronization is jointed with node localization by adopting [17] and therefore we assume that all nodes are time synchronized.

(2) To avoid packet collision, all nodes transmit separately, for example, according to a Poisson distribution with an average transmission rate of λ packets per second. Collision-tolerant packet scheduling [19] can be exploited to achieve a desired probability of successful self-localization for a given number N of anchors, while λ and the minimum localization time can be determined.

(3) To avoid broadcast storming, the sensor node only broadcasts the packet with its observed data but

forwards the received data packets to its reference nodes. The reference nodes are updated every P_l.

(4) Initially, only surface buoys and anchor nodes can fix their locations using GPS, a GPS intelligent buoy system, or other means.

(5) All the sensor nodes are functionally identical.

4.2.1. Node Localization. Again, DLCA is a hybrid approach comprising two parts: *node localization*, which is run in the nodes, and *data-location correction*, which is run at the base station. The sensor nodes initialize and periodically update their locations as follows.

To estimate their locations, the sensor nodes stamp the sending time T_1 and then immediately broadcast a *Req* message to their neighboring reference nodes. The sensor nodes then listen to the localization messages from any other localized nodes.

Upon receiving the *Req* message, each reference node immediately marks its local time as t_2 and then after a time interval t_r (avoiding collision), each of the reference node sends back a *Res* message containing its location information, t_2, and sending time t_3. When receiving the *Res* message, the ordinary node marks its receiving time T_4.

In each period P_l, the unlocalized or location-expired sensor nodes that received more than three location messages perform self-localization by multilateration [24, 25, 27] and time synchronization by Mobi-Sync [14]. The successfully localized and time synchronized sensor nodes start to observe data and then transmit data packets with their location information.

4.2.2. Data Packet Format for Data-Location Correction. DLCA relies on the information embedded in data packets to do data-location correction. To achieve a good balance between packet length and sufficient localization information, the data packet format is only slightly changed, as shown in Figure 3.

(i) *Observer ID* stores the ID of the sensor that observes the data and broadcasts this data packet.

(ii) *Observed data* stores the data content observed by the sensor node at each P_d. We assume here that sensors observe data periodically.

(iii) *Location* stores the location of the observer node.

(iv) *Observed time* stores the time when the data is observed. We assume here that the sensor broadcasts the data packet right after the data is observed.

The former four fields are essential for all UWSN applications. Two extra fields are added to support data-location correction: "receiver ID" and "received time."

(i) *Receiver ID* stores the ID of the node that is the first receiver of this data packet.

(ii) *Received time* stores the time when the first receiver receives this data packet.

Observer ID	Observed data	Location	Observed time	Receiver ID	Received time

Figure 3: Data packet format.

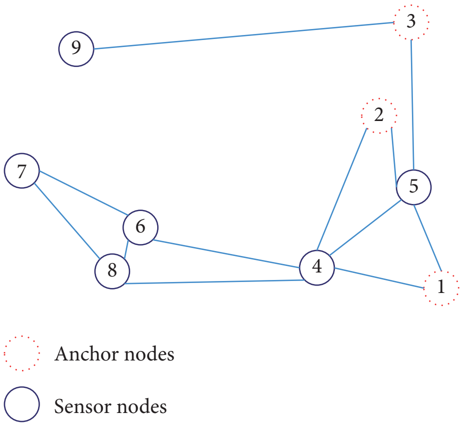


Figure 4: An example of DLCA graph.

Table 1: Data packets grouped with identical send time which is equal to 3.

Observer ID	Observed data	Location	Observed time	Receiver ID	Received time
4	...	(52, 29)	3	1	3.00657
4	...	(52, 29)	3	2	3.00712
4	...	(52, 29)	3	5	3.00780
4	...	(52, 29)	3	6	3.00555
4	...	(52, 29)	3	8	3.00227
5	...	(56, 35)	3	1	3.00717
5	...	(56, 35)	3	2	3.00225
5	...	(56, 35)	3	3	3.00597
6	...	(44, 31)	3	7	3.00352
7	...	(39, 36)	3	8	3.00655
8	...	(43, 29)	3	6	3.00386
9	...	(41, 45)	3	3	3.00757

After several transmissions, data packets will be transmitted to the base station. The base station then classifies received data packets according to their observed time. Packets with the same or close observed time are grouped together. Table 1 illustrates an example of grouping received data packets with identical observed time = 3. When the number of data packets in a group is sufficient or a predetermined correct time is achieved [32], the base station starts the data-location correction procedure.

4.2.3. Data Structure of DLCA. Data packets within the same group can be illustrated schematically on a *DLCA graph*. For example, Figure 4 shows the DLCA graph of Table 1. In the DLCA graph, there are sensor nodes and anchor nodes within the same packet table. For observer node p and receiver node

Table 2: DLCA table.

n_{id}	Location	$\text{cv}_{\text{loc}}(n_{\text{id}})$	Neighbor list			
			nb_{id}	$d_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$	$\text{cv}_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$	$d_{\text{Euc}}(n_{\text{id}}, nb_{\text{id}})$

q in a data packet, we denote the two nodes p and q as *neighbor* nodes to each other and use an undirected edge $e(p, q)$ to show their relationship in the DLCA graph.

A *DLCA table* is used to maintain information of the DLCA graph, as shown by Table 2. The fields of the DLCA table, from left to right, are node ID, data packet location, accuracy of the data packet location, information of their neighbor nodes including the neighbor node's ID, ToA distance between the node and its neighbor, accuracy of ToA distance, and Euclidean distance between the node and its neighbor. We assume that the base station has all the information of anchor nodes, including the locations and accuracies.

4.2.4. DLCA Table Initialization. Each node on the DLCA graph is an entry in the DLCA table. Initially, the data packet location is set as the location after node self-localization every P_l. The accuracy of every P_d, which is the confidence value of the data packet location at every P_d, is given by $1/(1+n)$, where n ranges from 0 to $(P_l/P_d - 1)$ as the accuracy of the location will decrease with time, and P_l can be a multiple of P_d. $d_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$ represents the ToA distance between n_{id} and its neighbor nb_{id}. The length is computed based on (1). $\text{cv}_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$ represents the confidence value of the ToA distance. The confidence value of each ToA distance is set based on the following equation:

$$\text{cv}_{\text{ToA}}\left(n_{\text{id}}, nb_{\text{id}}\right) = \begin{cases} 0, & d_{\text{ToA}}\left(\cdots\right) > R \\ 1 - \dfrac{d_{\text{ToA}}\left(n_{\text{id}}, nb_{\text{id}}\right)}{R}, & \text{otherwise}, \end{cases} \tag{4}$$

where R is the communication radius of the sensor node. The confidence value of the ToA distance is set to zero when the distance is larger than R, as it is unreasonable that the measured distance is larger than the communication radius.

The Euclidean distance $d_{\text{Euc}}(n_{\text{id}}, nb_{\text{id}})$ between n_{id} and nb_{id} is computed based on the two nodes' locations and follows (5). The $d_{\text{Euc}}(n_{\text{id}}, nb_{\text{id}})$ is updated when the location of the data packet is updated.

$$\begin{aligned} d\left(p, q\right) &= d\left(q, p\right) \\ &= \sqrt{\left(q_1 - p_1\right)^2 + \left(q_2 - p_2\right)^2 + \left(q_3 - p_3\right)^2}. \end{aligned} \tag{5}$$

Here we use the same example to illustrate the DLCA table initialization. In Table 3, nodes with number (1, 2, 3) are anchor nodes. Their confidence values are 1 because they are assumed to be accurate. Their neighbor lists are null because they do not need other nodes to correct their locations. The remaining nodes in the DLCA table are sensor nodes. In this example, P_d is set as 1. The confidence value of the data packet location is 0.25 because the last sensor self-localized time is 0, and the data packet observed time is 3. The neighbor nodes of node 4 include nodes 1, 2, 5, 6, and 8. The ToA distance between nodes 1 and 4 is 9.86 because the data packet is sent at time = 3 and the data packet was first received by node 1 at 3.00659 and the speed of sound in seawater is 1500 m/s. The communication radius is 12 so the confidence value of the ToA distance is 0.18. The Euclidean distance between node 4 and node 1 is the distance between (52, 29) and (57.24, 27.68).

4.3. Recursively Correcting Data Packet Locations

4.3.1. Find Victim Node, s_v. Entries with a confidence value less than a designated threshold, θ_{victim}, are considered as correctable candidates. The base station will continually choose one of the candidates to see if it is qualified to be the *victim node*, s_v in short.

From the example of Figure 4, the base station first selects node 4, which has sufficient neighbors. For each neighbor node of node 4, the base station computes its reliable value. The *reliable value*, computed by (6), is used to quickly choose the four most reliable neighbor nodes as reference nodes, r_{id}.

$$\text{reliablevalue} = \text{cv}_{\text{loc}}\left(nb_{\text{id}}\right) * \text{cv}_{\text{ToA}}\left(n_{\text{id}}, nb_{\text{id}}\right). \tag{6}$$

To see if the four reference nodes are reliable enough to correct the data packet location, the base station then computes the new confidence value of node 4 after being corrected by reference nodes based on

$$\begin{aligned} &\text{cv}_{\text{loc}}\left(s_v\right) \\ &\quad = \frac{1}{4}\sum\left[\text{cv}_{\text{loc}}\left(r_{\text{id}}\right) * (1-\alpha) + \text{cv}_{\text{ToA}}\left(n_{\text{id}}, r_{\text{id}}\right) * \alpha\right], \end{aligned} \tag{7}$$

$$\alpha = \frac{\text{cv}_{\text{ToA}}\left(n_{\text{id}}, r_{\text{id}}\right)}{\text{cv}_{\text{loc}}\left(r_{\text{id}}\right) + \text{cv}_{\text{ToA}}\left(n_{\text{id}}, r_{\text{id}}\right)}. \tag{8}$$

DLCA adopts the average confidence values of the four references nodes as the confidence value of s_v, as the location of s_v will be computed according to the four reference nodes. α, which is called *adjustment parameter*, is used to adjust the weight of the ToA distance confidence value. When the ToA distance is more accurate than the locations of neighbor nodes, α will be greater than 0.5.

If $\text{cv}_{\text{loc}}(s_v)$ is higher than $\text{cv}_{\text{loc}}(n_{\text{id}})$, the node is classified as a victim node s_v and its location is computed by the method described in the next section. Otherwise, the base station

Table 3: Initialization of DLCA table.

n_{id}	Location	$\text{cv}_{\text{loc}}(n_{\text{id}})$	Neighbor list			
			nb_{id}	$d_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$	$\text{cv}_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$	$d_{\text{Euc}}(n_{\text{id}}, nb_{\text{id}})$
1	(57.24, 27.68)	1	N/A			
2	(54.47, 40.40)	1	N/A			
3	(55.83, 47.11)	1	N/A			
4	(52, 29)	0.25	1	9.86	0.18	5.4
			2	10.68	0.11	11.66
			5	11.7	0.02	7.21
			6	8.32	0.31	8.25
			8	3.41	0.72	9
5	(56, 35)	0.25	1	10.76	0.1	7.42
			2	3.37	0.72	5.61
			3	8.96	0.25	12.11
			4	11.7	0.02	7.21
6	(44, 31)	0.25	4	8.32	0.31	8.25
			7	5.28	0.56	7.07
			8	5.79	0.52	2.24
7	(39, 36)	0.25	6	5.28	0.56	7.07
			8	9.82	0.18	8.06
8	(43, 29)	0.25	4	3.41	0.72	9
			6	5.79	0.52	2.24
			7	9.82	0.18	8.06
9	(41, 45)	0.25	3	11.36	0.05	14.98

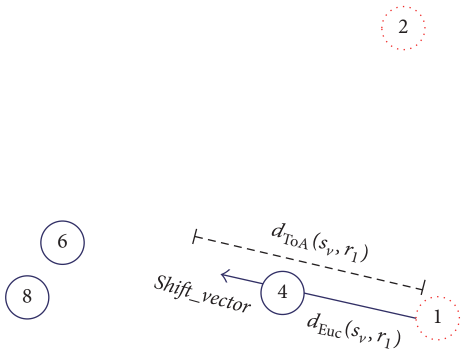


Figure 5: $d_{\text{ToA}}(s_v, r_{\text{id}}) > d_{\text{Euc}}(s_v, r_{\text{id}})$.

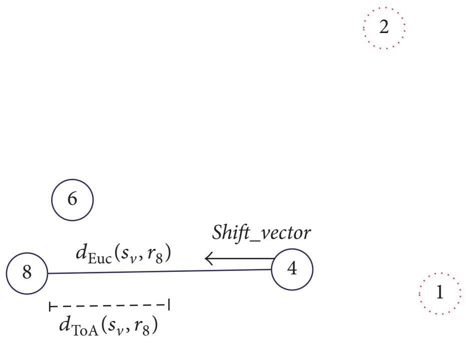


Figure 6: $d_{\text{ToA}}(s_v, r_{\text{id}}) < d_{\text{Euc}}(s_v, r_{\text{id}})$.

skips this node and selects another candidate to repeat the procedure of selecting the candidate s_v. The recursive victim node selection stops when there are no candidates left. Table 4 shows the result of victim selection.

4.3.2. Compute Location of Victim Node by Shift Vectors. DLCA uses the ToA distance and the Euclidean distance to decide the *shift vector* of the victim node. However, there are two conditions of the *shift vector*.

Condition 1. $d_{\text{ToA}}(s_v, r_{\text{id}}) > d_{\text{Euc}}(s_v, r_{\text{id}})$ is shown in Figure 5. In this condition, the victim node s_v (node 4) will be adjusted far from its reference node 1.

Condition 2. $d_{\text{ToA}}(s_v, r_{\text{id}}) < d_{\text{Euc}}(s_v, r_{\text{id}})$ is shown in Figure 6. In this condition, the victim node s_v (node 4) will be adjusted closer to its reference node 8.

Equation (9) calculates the *shift vector*:

$$\vec{v_i} = \alpha * \left(d_{\text{ToA}}\left(s_v, r_{\text{id}}\right) - d_{\text{Euc}}\left(s_v, r_{\text{id}}\right)\right) * \frac{\overrightarrow{\left(r_{\text{id}}, s_v\right)}}{\left|\overrightarrow{\left(r_{\text{id}}, s_v\right)}\right|}, \quad (9)$$

where subscript i is the reference node id and α is the adjustment parameter which is defined by (8), to control the length to adjust s_v.

Each reference node of s_v generates one shift vector as shown in Figure 7.

Table 4: The result of victim selection.

s_v	nb_{id}	$d_{\text{ToA}}(n_v, nb_{\text{id}})$	$\text{cv}_{\text{ToA}}(n_v, nb_{\text{id}})$	$d_{\text{Euc}}(n_v, nb_{\text{id}})$	Reliable_value	Reference node
4	1	9.86	0.18	5.4	0.18	v
	2	10.68	0.11	11.66	0.11	v
	5	11.7	0.02	7.21	0.01	
	6	8.32	0.31	8.25	0.08	v
	8	3.41	0.72	9	0.18	v

Table 5: DLCA table after DLCA.

n_{id}	Location	$\text{cv}_{\text{loc}}(n_{\text{id}})$	Neighbor list			
			nb_{id}	$d_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$	$\text{cv}_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$	$d_{\text{Euc}}(n_{\text{id}}, nb_{\text{id}})$
1	(57.24, 27.68)	1	N/A			
2	(54.47, 40.40)	1	N/A			
3	(55.83, 47.11)	1	N/A			
4	(47.28, 29.16)	0.67	1	9.86	0.18	10.07
			2	10.68	0.11	13.35
			5	11.7	0.02	11.45
			6	8.32	0.31	3.76
			8	3.41	0.72	4.28
5	(55.73, 36.88)	0.82	1	10.76	0.1	9.32
			2	3.37	0.72	3.74
			3	8.96	0.25	10.23
			4	11.7	0.02	11.45
6	(44, 31)	0.25	4	8.32	0.31	3.76
			7	5.28	0.56	7.07
			8	5.79	0.52	2.24
7	(39, 36)	0.25	6	5.28	0.56	7.07
			8	9.82	0.18	8.06
8	(43, 29)	0.25	4	3.41	0.72	4.28
			6	5.79	0.52	2.24
			7	9.82	0.18	8.06
9	(41, 45)	0.25	3	11.36	0.05	14.98

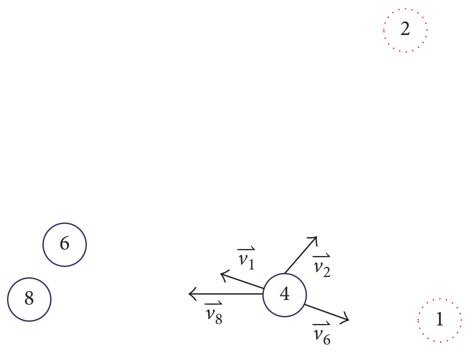


Figure 7: Shift vectors produced by four reference nodes of s_v.

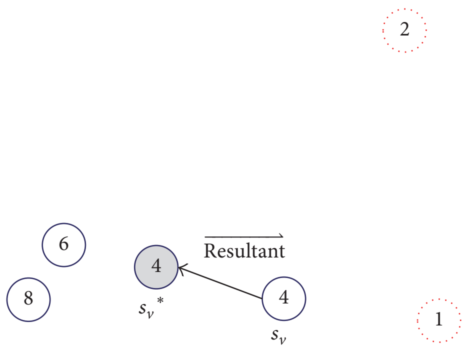


Figure 8: Corrected location of s_v.

The corrected location of s_v, which is denoted as s_v^*, is decided by the resultant of the four shift vectors generated by the four reference nodes as calculated in (10) and Figure 8.

$$\overrightarrow{s_v^*} = \overrightarrow{s_v} + \sum \overrightarrow{v_i}. \tag{10}$$

4.3.3. Recursively Correct Data Locations. When there are no candidates left, the mean value of total $\text{cv}_{\text{loc}}(n_{\text{id}})$ is computed. When the variance of the mean value of $\text{cv}_{\text{loc}}(n_{\text{id}})$ is less than a threshold θ_{cv}, the DLCA process is complete. Table 5 presents the final result of DLCA.

Table 6: Simulation settings.

Settings	Values
Number of nodes	500
Anchor ratio	10%
Region	100 m $*$ 100 m $*$ 100 m
Communication radius, R	25 m
Data observed period, P_d	1 s
DLCA discontinue threshold, θ_{cv}	0.001

5. Simulation Results

In this section, we evaluate the performance of DLCA using simulations.

5.1. Simulation Settings. In our simulations, unless otherwise specified, we use the settings shown in Table 6. Five hundred nodes, comprising 50 anchor nodes and 450 sensor nodes, are randomly distributed in a 100 m × 100 m × 100 m region, with a communication radius of 25 m. P_d is set to 1 s, and θ_{cv} is set to 0.001. All the results are the mean value of 100 simulations. The simulation was run on a personal computer: Intel core i7-3770, 3.4 GHz with 16 GB RAM, and 64 bit Windows 7. The computing time is approximately 2.5 ms for 500 nodes.

Table 7 presents the memory needed to maintain a DLCA table in a timing group. For each node, we use a 4-byte integer to hold the node ID. For location information, three 4-byte floats are used due to the three dimensions. The confidence value is also held by a 4-byte float. All nodes, excluding the anchors, need to maintain a list of neighbors. The information of a neighbor is a node ID, which can be held by a pointer, and the other three fields can each be held by a 4-byte float. Therefore, the total memory usage is 20 bytes for a node and 12 bytes for a neighbor's information. Taking our simulation as an example, it needs (500 nodes × 20 bytes) + (12 bytes × average degree × 450 sensors), which is approximately 64 K bytes of memory.

As for the node mobility pattern, two models are adopted. One is the kinematic model in [37]. The current field is assumed to be a superposition of a tidal and a residual current field. The tidal field is assumed to be a spatially uniform oscillating current in one direction, and the residual current field is assumed to be an infinite sequence of clockwise and anticlockwise rotating eddies. The dimensionless velocity field in the kinematical model can be approximated by

$$V_x = k_1 \lambda v \sin(k_2 x) \cos(k_3 y) + k_1 \lambda \cos(2k_1 t) + k_4,$$
$$V_y = -\lambda v \cos(k_2 x) \sin(k_3 y) + k_5, \tag{11}$$

where V_x is the speed in the x-axis, V_y is the speed in the y-axis, and k_1, k_2, k_3, λ, and v are variables which are closely related to environmental factors, such as tides and bathymetry. These parameters will change in different environments.

We use *node density* as the node degree, which is the expected number of nodes in a node's neighborhood. Node density impact on *localization coverage* is defined as the ratio of localization nodes to the total nodes. In Figure 9, the node density is varied by changing the communication range of all the nodes from 5 m to 35 m. Figure 9 shows that, for localization methods based on trilateral positioning, the minimum node density to achieve 0.8 localization coverage is 10.

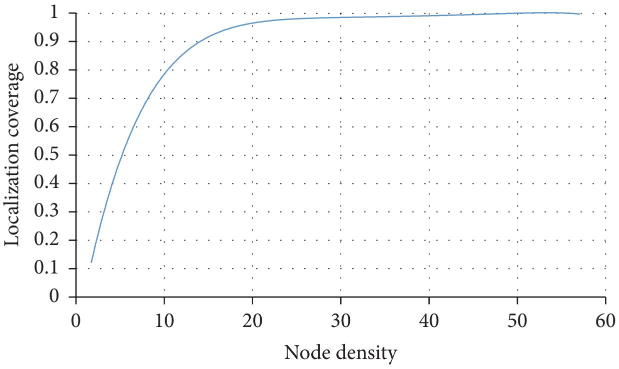


Figure 9: Impact of node density on localization coverage.

By simulating the node mobility based on the kinematic model, we found that the node density decreases quickly as time increases, especially when the speed of ocean current is high. To keep the area of deployment the same and maintain high localization coverage (>0.8), in this paper we simulate the ocean current by assuming that all the parameters are random variables subject to a normal distribution with the settings presented in Table 8.

ToA distance measurements between nodes are assumed to follow (2) and (3), which are subject to the normal distributions with 0.01 as mean values and 0.05 as the standard deviations. This is a reasonable assumption and can be easily justified by existing underwater distance measurement technologies [11–13].

In addition to the kinematic model [37], the meandering model [38] is also considered in this paper. The nondimensional form of the meandering jet model is

$$\begin{aligned}
\psi(x, y, t) &= -\tanh\left[\frac{y - B(t)\sin(k(x - ct))}{\sqrt{1 + k^2 B^2(t)\cos^2(k(x - ct))}}\right], \\
u &= -\frac{\partial \psi}{\partial y}; \\
v &= -\frac{\partial \psi}{\partial x}, \\
\dot{x} &= -\partial_y \psi(x, y, t), \\
\dot{y} &= -\partial_x \psi(x, y, t),
\end{aligned} \tag{12}$$

where $B(t) = A + \epsilon\cos(\omega t)$. The parameter k sets the number of meanders in the unit length. c is the phase speed with which they shift downstream. The time-dependent function B modulates the width of the meanders: A determines the average meander width, ϵ is the amplitude of the modulation,

Table 7: Memory usage of a DLCA table.

	n_{id}	Location	$\text{cv}_{\text{loc}}(n_{\text{id}})$	Neighbor list			
				nb_{id}	$d_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$	$\text{cv}_{\text{ToA}}(n_{\text{id}}, nb_{\text{id}})$	$d_{\text{Euc}}(n_{\text{id}}, nb_{\text{id}})$
Bytes	4	4 ∗ 3 dimensions	4	Pointer	4	4	4

Table 8: Ocean current parameter settings.

Parameters	Mean	std
k_1, k_2	π	$0.1\,\pi$
k_3	2π	$0.2\,\pi$
k_4, k_5	1	0.1
v	1	0.1
λ	0.5~3.0	0.05~0.3

Table 9: Meandering model parameter settings.

Parameters	Values
Average meander width, A	1.2
Phase speed, c	0.12
Number of meanders in the unit length, k	$2\,\pi/7.5$
Frequency, ω	0.4
Amplitude of the modulation, ϵ	2, 3, 4, 5

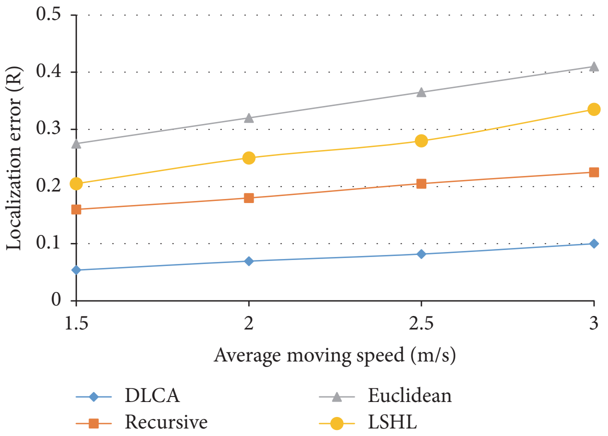


Figure 10: DLCA performance on localization error compared to other schemes.

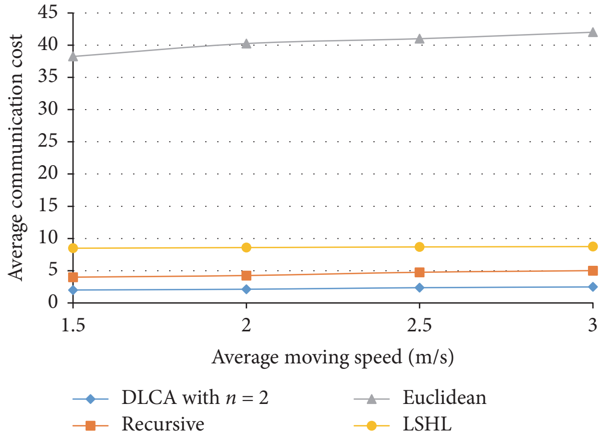


Figure 11: DLCA performance on average communication cost compared to other schemes.

and ω is its frequency. We use the meandering model to simulate the ocean current based on the settings in [38], presented in Table 9.

We consider two performance metrics: *localization error* and *average communication cost*. Localization error is the average distance between the estimated positions and the real positions of all nodes. As in [25, 27], for our simulations, we normalize this absolute localization error to the node communication range R. Average communication cost is defined as the overall messages exchanged in the network divided by the number of localized nodes, which is normalized to the size of the beacon message (16 bytes in our simulations).

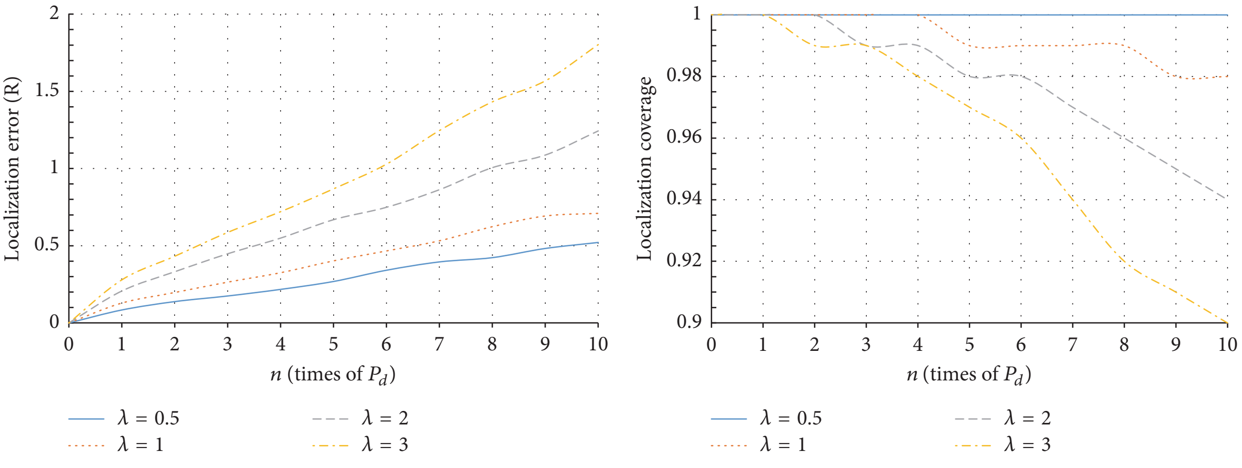
5.2. Results and Analysis

5.2.1. Performance with Varying Average Moving Speed. In this simulation, we compare our scheme to three schemes: LSHL scheme [25, 26], Euclidean scheme [39], and recursive scheme [40]. P_l is set as $2P_d$. Figures 10 and 11 clearly show the effect of node mobility on the localization performance. We can see that DLCA successfully corrects data packet locations and decreases the communication cost. This is because DLCA is executed in the base station; therefore, the period of sending localization message can be extended to decrease the communication costs. The localization error of all the schemes increases with the speed at which the node moves. This is mainly because the distance measurement error increases with the average moving speed. Correspondingly, the final localization error will increase as well.

5.2.2. Performance with Varying Times of Pd. In this set of simulations, we set the length of P_l to 11 multiples of P_d, and change n from 0 to 10. For kinematic model, we consider different marine environments by setting λ to be 0.5, 1, 2, and 3, and the corresponding speeds of ocean current are 2.54, 3.95, 6.44, and 8.86 (m/s), respectively. For meandering model, we set ϵ to be 2, 3, 4, 5, and the corresponding speeds of ocean current are 0.86, 1.61, 2.05, and 3.25 (m/s), respectively.

It is shown in [25] that range-based localization schemes place large requirements on the node density of networks. Figure 9 also shows that the node density needs to be at least 10 in order to localize 80% of the nodes.

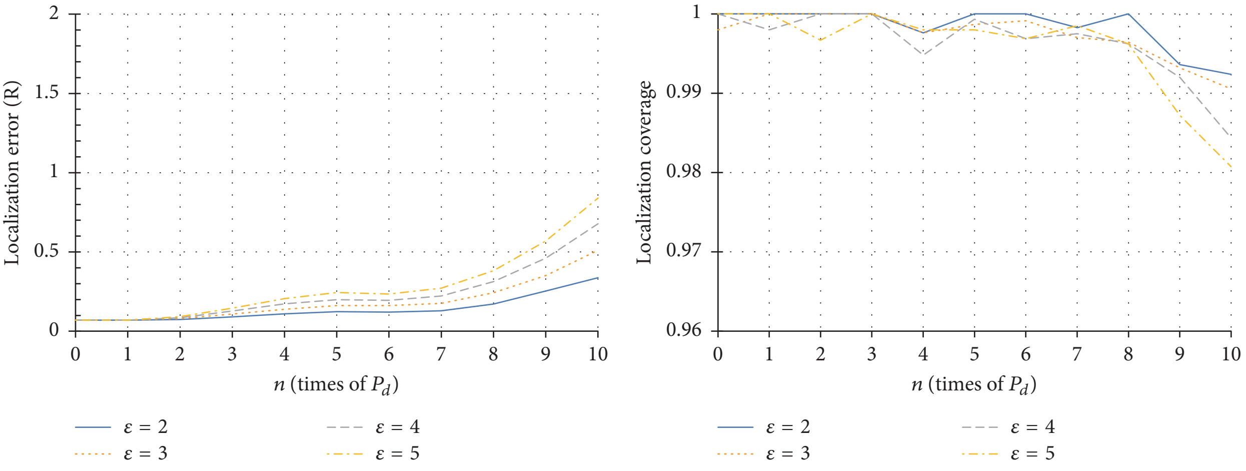
Figures 12 and 13 provide references for UWSN applications to decide a suitable n to fit their performance requirement in different marine environments based on kinematic model and meandering model, respectively. For example, the maximum n is 10 to localize 99% nodes with less than $0.5R$



(a)

(b)

Figure 12: DLCA performance on (a) localization error and (b) coverage with varying λ and n (times of P_d) based on kinematic model.



(a)

(b)

Figure 13: DLCA performance on (a) localization error and (b) coverage with varying ε and n (times of P_d) based on meandering current mobility model.

localization error, when 10% anchor nodes and less than 3 m/s speed are present in the network.

5.2.3. Performance with Varying Anchor Percentage. In the simulation of Figure 14, λ is set to 0.5 and ε is set to 5. We can see that the more the anchors, the lower the localization error. For example, if the anchor percentage is 5%, the slope is almost 0.6, but if the anchor percentage is 20%, the slope can reach 0.45. This suggests that, in sparse networks, we can increase the number of anchor nodes to achieve higher localization accuracy. It should be noted that, for all range-based localization methods, their localization accuracy increases with the anchor percentage.

6. Summary

In summary, the main contributions of this paper are as follows: (1) we are the first to propose a hybrid localization approach which, post facto, corrects data locations at a base station to improve the overall network communication cost and sensor power. With our hybrid localization approach, the period of sensor self-localization can be extended and thus decrease the computational overheads and high energy requirements of sensors; (2) DLCA is easily implemented and is cost-efficient in both computing time and memory space, and (3) we analyze DLCA performance under different marine environments by simulating ocean-current speeds based on kinematic models and also compare the results to several range-based localization approaches. The

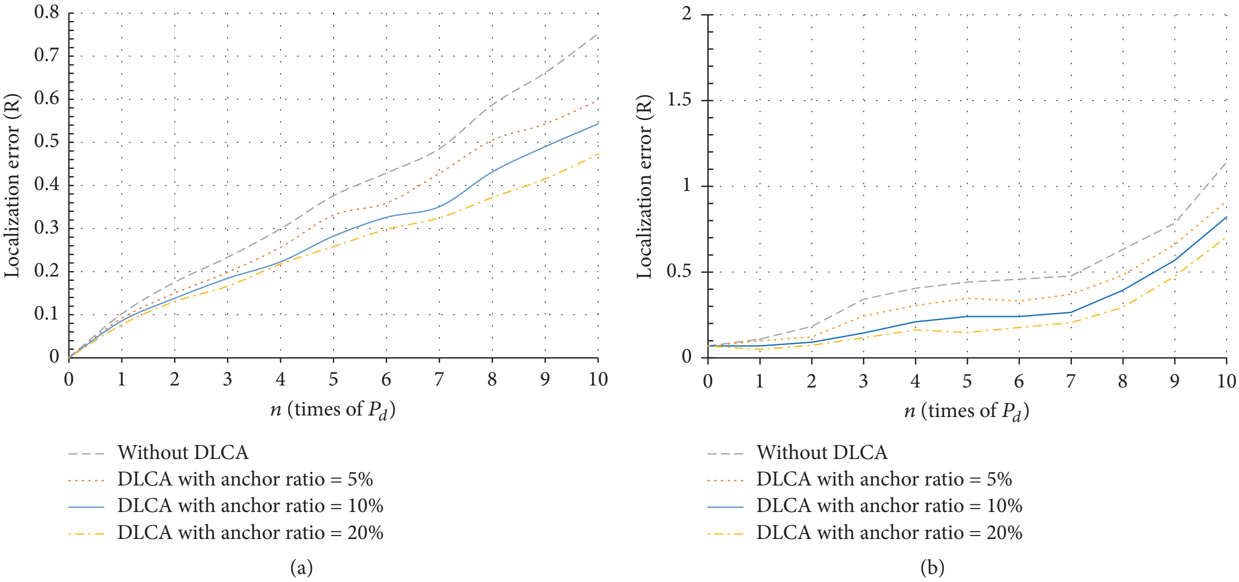


Figure 14: Performance on localization error with varying anchor ratio and n (times of P_d) based on (a) kinematic model and (b) meandering current mobility model.

results indicate satisfactory performance of our proposed scheme.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] H.-P. Tan, R. Diamant, W. K. G. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization," *Ocean Engineering*, vol. 38, no. 14-15, pp. 1663–1676, 2011.

[2] M. Beniwal and R. Singh, "Localization techniques and their challenges in underwater wireless sensor networks," *International Journal of Computer Science and Information Technologies*, vol. 5, pp. 4706–4710, 2014.

[3] G. Han, J. Jiang, L. Shu, Y. Xu, and F. Wang, "Localization algorithms of underwater wireless sensor networks: a survey," *Sensors*, vol. 12, no. 2, pp. 2026–2061, 2012.

[4] M. Erol-Kantarci, H. T. Mouftah, and S. Oktug, "A survey of architectures and localization techniques for underwater acoustic sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 487–502, 2011.

[5] V. Garg and M. Jhamb, "A review of wireless sensor network on localization techniques," *International Journal of Engineering Trends and Technology*, vol. 4, pp. 1049–1053, 2013.

[6] T. J. S. Chowdhury, C. Elkin, V. Devabhaktuni, D. B. Rawat, and J. Oluoch, "Advances on localization techniques for wireless sensor networks: a survey," *Computer Networks*, vol. 110, pp. 284–305, 2016.

[7] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: a survey," *Telecommunication Systems*, vol. 52, pp. 1–18, 2011.

[8] S. Lee and K. Kim, "Localization with a mobile beacon in underwater sensor networks," in *Proceedings of the IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing, EUC '10*, pp. 316–319, December 2010.

[9] H. Luo, Z. Guo, W. Dong, F. Hong, and Y. Zhao, "LDB: localization with directional beacons for sparse 3D underwater acoustic sensor networks," *Journal of Networks*, vol. 5, no. 1, pp. 28–38, 2010.

[10] H. Luo, Y. Zhao, Z. Guo, S. Liu, P. Chen, and L. M. Ni, "UDB: using directional beacons for localization in underwater sensor networks," in *Proceedings of 14th IEEE International Conference on Parallel and Distributed Systems, ICPADS '08*, pp. 551–558, December 2008.

[11] S. A. Golden and S. S. Bateman, "Sensor measurements for Wi-Fi location with emphasis on time-of-arrival ranging," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1185–1198, 2007.

[12] H. Xiong, Z. Chen, W. An, and B. Yang, "Robust TDOA localization algorithm for asynchronous wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 598747, 10 pages, 2015.

[13] K. K. Chintalapudi, A. Dhariwal, R. Govindan, and G. Sukhatme, "Ad-hoc localization using ranging and sectoring," in *Proceedings of the IEEE INFOCOM 2004 - Conference on Computer Communications - Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 2662–2672, March 2004.

[14] J. Liu, Z. Zhou, Z. Peng, J.-H. Cui, M. Zuba, and L. Fiondella, "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.

[15] J. Liu, Z. Wang, M. Zuba, Z. Peng, J.-H. Cui, and S. Zhou, "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.

[16] Z. Li, Z. Guo, F. Hong, and L. Hong, "E^2DTS: an energy efficiency distributed time synchronization algorithm for underwater acoustic mobile sensor networks," *Ad Hoc Networks*, vol. 11, no. 4, pp. 1372–1380, 2013.

[17] J. Liu, Z. Wang, J.-H. Cui, S. Zhou, and B. Yang, "A joint time synchronization and localization design for mobile underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 530–543, 2016.

[18] K. Chen, M. Ma, E. Cheng, F. Yuan, and W. Su, "A survey on MACprotocols for underwater wireless sensor networks," *IEEE Communications Surveys Tutoials*, vol. 16, pp. 1433–1447, 2014.

[19] H. Ramezani, F. Fazel, M. Stojanovic, and G. Leus, "Collision tolerant and collision free packet scheduling for underwater acoustic localization," *IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 2584–2595, 2015.

[20] M. Erol, L. F. M. Vieira, and M. Gerla, "AUV-aided localization for underwater sensor networks," in *Proceedings of the Proceeding of the 2nd Annual International Conference on Wireless Algorithms, Systems, and Applications (WASA '07)*, pp. 44–54, Chicago, Ill, USA, August 2007.

[21] M. Waldmeyer, H.-P. Tan, and W. K. G. Seah, "Multi-stage AUV-aided localization for underwater wireless sensor networks," in *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA '11)*, pp. 908–913, March 2011.

[22] M. Erol, L. F. M. Vieira, and M. Gerla, "Localization with Dive'N'Rise (DNR) beacons for underwater acoustic sensor networks," in *Proceedings of the 2007 International Conference on Mobile Computing and Networking, MobiCom '07 - Second Workshop on Underwater Networks, WUWNet'07*, pp. 97–100, September 2007.

[23] M. Erol, L. F. M. Vieira, A. Caruso, F. Paparella, M. Gerla, and S. Oktug, "Multi stage underwater sensor localization using mobile beacons," in *Proceedings of the 2nd International Conference on Sensor Technologies and Applications*, pp. 710–714, Cap Esterel, France, August 2008.

[24] T. Ojha and S. Misra, "MobiL: a 3-dimensional localization scheme for Mobile Underwater Sensor Networks," in *Proceedings of 2013 National Conference on Communications, NCC '13*, February 2013.

[25] Z. Zhou, J.-H. Cui, and S. Zhou, "Efficient localization for large-scale underwater sensor networks," *Ad Hoc Networks*, vol. 8, no. 3, pp. 267–279, 2010.

[26] G. Han, A. Qian, C. Zhang, Y. Wang, and J. J. P. C. Rodrigues, "Localization algorithms in large-scale underwater acoustic sensor networks: a quantitative comparison," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 379382, 11 pages, 2014.

[27] Z. Zhou, Z. Peng, J.-H. Cui, Z. Shi, and A. Bagtzoglou, "Scalable localization with mobility prediction for underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 3, pp. 335–348, 2011.

[28] Y. Zhang, J. Liang, S. Jiang, and W. Chen, "A localization method for underwater wireless sensor networks based on mobility prediction and particle swarm optimization algorithms," *Sensors*, vol. 16, no. 2, 2016.

[29] M. Liu, X. Guo, and S. Zhang, "Localization based on best spatial correlation distance mobility prediction for underwater wireless sensor networks," in *Proceedings of the 34th Chinese Control Conference, CCC '15*, pp. 7827–7832, July 2015.

[30] G. Zhu, R. Jiang, L. Xie, and Y. Chen, "A distributed localization scheme based on mobility prediction for underwater wireless sensor networks," in *Proceedings of the 26th Chinese Control and Decision Conference, CCDC '14*, pp. 4863–4867, June 2014.

[31] D. Mirza and C. Schurgers, "Collaborative localization for fleets of underwater drifters," in *Proceedings of the OCEANS*, pp. 1–6, Vancouver, Canada, October 2007.

[32] D. Mirza and C. Schurgers, "Motion-aware self-localization for underwater networks," in *Proceedings of the 3rd ACM International Workshop on Underwater Networks*, pp. 51–58, 2008.

[33] D. Mirza and C. Schurgers, "Energy-efficient ranging for post-facto self-localization in mobile underwater networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1697–1707, 2008.

[34] C. Bechaz and H. Thomas, "GIB System: the underwater GPS solution," *Proceedings of the 5th Europe Conference on Underwater Acoustics*, 2000.

[35] T. C. Austin, R. P. Stokey, and K. M. Sharp, "PARADIGM: A buoy-based system for AUV navigation and tracking," *Oceans Conference Record (IEEE)*, vol. 2, pp. 935–938, 2000.

[36] J. Callmer, M. Skoglund, and F. Gustafsson, "Silent localization of underwater sensors using magnetometers," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, Article ID 709318, 8 pages, 2010.

[37] S. P. Beerens, H. Ridderinkhof, and J. T. F. Zimmerman, "An analytical study of chaotic stirring in tidal areas," *Chaos, Solitons & Fractals*, vol. 4, no. 6, pp. 1011–1029, 1994.

[38] A. Caruso, F. Paparella, L. F. M. Vieira, M. Erol, and M. Gerla, "The meandering current mobility model and its impact on underwater mobile sensor networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 221–225, Phoenix, Ariz, USA, April 2008.

[39] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1734–1743, San Francisco, Calif, USA, April 2003.

[40] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *Proceedings of the 2001 International Conference on Network Protocols ICNP*, pp. 35–41, November 2001.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration