

An efficient AUV-aided data collection in underwater sensor networks

Ghoreyshi, Seyed Mohammad; Shahrabi, Alireza; Boutaleb, Tuleen

Published in:

IEEE International Conference on Advanced Information Networking and Applications (AINA 2018)

Publication date:

2018

Document Version

Peer reviewed version

[Link to publication in ResearchOnline](#)

Citation for published version (Harvard):

Ghoreyshi, SM, Shahrabi, A & Boutaleb, T 2018, An efficient AUV-aided data collection in underwater sensor networks. in *IEEE International Conference on Advanced Information Networking and Applications (AINA 2018)*. IEEE.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please view our takedown policy at <https://edshare.gcu.ac.uk/id/eprint/5179> for details of how to contact us.

An Efficient AUV-aided Data Collection in Underwater Sensor Networks

Syed Mohammad Ghoreyshi, Alireza Shahrabi, and Tuleen Boutaleb
 School of Engineering and Built Environment,
 Glasgow Caledonian University, Glasgow, UK
 {Seyed.MohammadGhoreyshi, A.Shahrabi, T.Boutaleb}@gcu.ac.uk

Abstract—From the view of routing protocols in Underwater Sensor Networks (UWSNs), mobile data-gathering mechanisms using Autonomous Underwater Vehicle (AUV) have received significant attention because of data collection capability via short-range communications. In this paper, a new Cluster-based AUV-aided Data Collection scheme (CADC) for large-scale UWSNs is proposed to make a trade-off between energy saving and data gathering latency. Our scheme consists of three phases: discovery phase, clustering phase, and data gathering phase. Neighbouring information is exchanged and then collected by AUV during the discovery phase. The collected information is used in the clustering phase in order to determine the cluster heads and members. Then, the AUV tour is planned such that all cluster heads are visited while shortening the tour length of the AUV. To cluster the sensors and cover their heads with the shortest possible tour, we first propose an optimal algorithm to find the global optimal solution, and then propose an efficient algorithm to obtain the near-optimal solution in the less computational time. CADC is scalable and also applicable in both connected and disconnected networks. In terms of energy-latency trade-off, CADC can effectively keep the tour length short while prolonging the network lifetime compared to those of mobile data-gathering approaches. The effectiveness of CADC is validated through an extensive simulation study which reveals the performance improvement in the packet delivery ratio, energy saving, and data gathering latency.

I. INTRODUCTION

Underwater sensor networks have obtained a tremendous interest in a wide range of aquatic applications, such as exploration of ocean resource, environmental monitoring, battlefield surveillance, disaster prevention, pollution monitoring, etc [1]–[3]. Underwater sensors are usually distributed in a large-scale marine environment to collect data and transfer them to a destination, which may be a static sink, or a mobile sink (e.g. AUV) [4]–[6].

In a static sink model, underwater sensors close to the sink consume much more energy than other sensors because of relaying more data packets. Thus, these nodes may fail sooner affecting the network connectivity. This problem can be exacerbated in large-scale UWSNs. However, data-gathering using AUV is more suitable for large-scale networks due to reducing the number of transmissions and balancing the energy consumption [7].

AUV is a mobile sink equipped with a powerful transceiver, moving through the underwater area to continuously collect data packets from sensors [4]. The data-gathering tour is periodically initiated from a static base station, followed

by collecting data packets from sensors, and completed by transferring data packets to the static base station [8]. Using a mobile sink contributes to prolong the lifetime of sensors since any packet relay is bounded within a given number of hops.

In AUV tour planning, there is a trade-off between energy saving and data gathering latency [9]. If AUV traverses within the transmission range of each sensor directly to collect data without any data packet relay, the maximum energy saving can be achieved for sensors. However, data gathering latency is increased due to increase in the tour length. Thus, it is more appropriate to decrease the data gathering latency by performing the local aggregation in a subset of sensors as cluster heads and transferring the aggregated data to AUV. It should be noted that the local data aggregation should be bounded to a few number of transmission hops to adequately handle the increased energy consumption and packet loss.

In this paper, a new Cluster-based AUV-aided Data Collection scheme (CADC) is presented to minimise tour length, packet loss, energy consumption, and latency. In CADC, a subset of underwater sensors is selected as cluster heads to collect data from affiliated sensors and transfer the data to AUV when it arrives. Furthermore, by affiliating sensors with cluster heads, any packet relay is bounded within a few number of hops which decreases the chance of collisions and packet loss. Limiting multi-hop relay to a certain level also reduces the energy consumption at sensors. To this end, we first propose a modified branch-and-bound algorithm to find the global optimal solution, and then propose a heuristic algorithm to obtain the near-optimal solution. In CADC, we take advantage of randomness and greediness to create some local solutions, and to find an acceptable solution among them using less computational time. The effectiveness of CADC is verified by comparing with other existing mobile data-gathering approaches in UWSNs.

The rest of this paper is organized as follows. In Section II, we review the related work on mobile data gathering in UWSNs. In Section III, we provide a detailed description of the system model. In Section IV, CADC is presented in details. Section V gives simulation results and some discussions. In Section VI, we conclude the paper.

II. RELATED WORK

In this section, we briefly review some of underwater mobile data-gathering protocols.

Mobicast [10] is a mobile data-gathering protocol in which AUV traverses a predetermined circle path to collect data packets from sensors in different geographic regions called 3-D zone of references (3-D ZOR). In Mobicast, there is no use of clustering mechanism and sensors should relay the data packets to AUV in a single hop or multi-hop transmissions. To deal with the presence of various water currents and void areas, a larger covering area surrounding the 3-D ZOR is considered to find alternative paths to deliver the packet to all sensors. This covering area involves more sensors resulting in more energy consumption. Furthermore, collecting data from all sensors in the sensing field is not possible because only sensors within 3-D ZORs can be investigated.

In AEERP (AUV aided Energy Efficient Routing Protocol) [11], AUV traverses a predetermined elliptical trajectory in each cycle. The sensors are divided into two categories: gateways and members. The gateway sensors only can communicate with an AUV, and they are selected based on the nearness to AUV trajectory and their remaining energy. Members are then allocated to the gateways sensors using a Shortest Path Tree (SPT). However, there is no bound on the hop distance from members to a gateway sensor which causes an increase in energy consumption in a broader network.

In AURP (AUV-aided underwater routing protocol) [12], multiple AUVs are used as a relay to collect data packets from gateway nodes and then forward to a sink. Each gateway is responsible for gathering data from underwater sensors. Thus, the sink and gateways should periodically broadcast their interest in receiving data to be used by sensors for choosing the next hop such that the path length is minimised. In AURP, it is mentioned that gateways and the trajectory of AUVs can be determined dynamically or before deployment; however, this procedure and the resulting overhead has not been investigated thoroughly. Moreover, an AUV trajectory is a fixed elliptical path which reduces the flexibility in confront of different sensor deployment strategies.

AUV_PN [13] is a mobile data-gathering protocol in which underwater sensors are clustered around several CHs while an AUV is employed to visit some identified locations to collect the aggregated data. AUV partitions the network using the Voronoi criteria and then travels the network field in a predefined lawn-mower pattern to broadcast the information. By receiving this information, sensors elects a CH using the cluster set-up phase of the LEACH protocol. Each CH also partitions its cluster into several sub-clusters and selects a Path Node (PN) for each sub-cluster with the responsibility to collect data from its members. During data-gathering phase, AUV travels to the nearest CH to obtain the list of PNs and then visits each PN to collect data. After visiting all PNs in a cluster, AUV travels to the next nearest CH and repeat the same procedure until all CHs are visited. However, there are some constraints which can confine the AUV_PN performance. The AUV tour is not optimal, and it crosses over itself. By broadening the network size, sensors should transmit the data packets with higher power because they are placed farther away from PNs. The network-partitioning has a complicated procedure, and it is performed with substantial overhead in energy and communication.

III. SYSTEM MODEL

In this section, the network architecture and acoustic propagation model are described in details.

A. Network architecture

A typical mobile data-gathering UWSN has a 3D network topology consisted of underwater sensors, a single data sink, and a mobile data sink (AUV) [13]. Underwater sensors are distributed in a two-dimensional plane with a fixed depth. Each sensor can control its depth using a pressure gauge and fish-like bladder apparatus [14], [15]. It is assumed that sensors are static or anchored to the bottom of the ocean. They are homogeneous regarding transmission range and power.

The static sink is located on the water surface, which can communicate with AUV and monitoring centre using an acoustic and radio modem, respectively. The AUV operates at a fixed depth above the underwater sensors, and it has the freedom to move in all directions. Underwater sensors are not required to know their full geographical coordinates; however, AUV can obtain sensors coordinates by marking the locations where it receives data from them [16]. It is also assumed that AUV is supplied with unlimited energy and memory.

B. Acoustic propagation model

The Thorp model [17] is used for describing the underwater acoustic channel model. The path loss or acoustic channel attenuation over distance d can be represented as [17]:

$$A(d, f) = A_0 d^k \alpha(f)^d \quad (1)$$

where f is the signal frequency and $\alpha(f)$ is the absorption coefficient which is determined by the Thorp model. Furthermore, A_0 denotes a unit-normalizing constant, and k is the geometric spreading factor which is set to 1.5 for practical scenarios. The underwater noises are dominant in the different frequency regions and are composed of four main components of turbulence $PN_t(f)$, shipping $PN_s(f)$, waves $PN_w(f)$ and thermal energy $PN_{th}(f)$ which can be represented as [18]:

$$PN(f) = PN_t(f) + PN_s(f) + PN_w(f) + PN_{th}(f) \quad (2)$$

The signal-to-noise ratio (SNR) over distance d with the signal frequency f can be expressed as [17]:

$$SNR(d, f) = \frac{PR(f)}{A(d, f)PN(f)} \quad (3)$$

where $PR(f)$ denotes the transmission power with frequency f at the forwarding node. To receive the data packet without any error, SNR at the receiver should be greater than the detection threshold. The bit error probability over distance d can be computed by [14]:

$$P_e(d) = \frac{1}{2} \left(1 - \sqrt{\frac{SNR_{avg}(d, f)}{1 + SNR_{avg}(d, f)}} \right) \quad (4)$$

where $SNR_{avg}(d, f)$ is the average signal-to-noise ratio over distance d with frequency f which can be calculated using Eq.

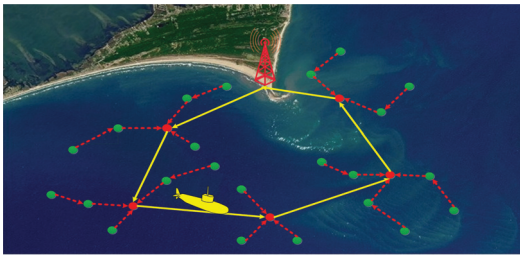


Fig. 1: Mobile data gathering within two hops

3. Accordingly, the delivery probability of a data packet with size n bits over distance d can be expressed as follows [14]:

$$P = (1 - P_e(d))^n \quad (5)$$

IV. CADC DETAILS

In this section, our proposed scheme (CADC) is presented.

A. Overview

The primary goal of a tour-planning routing protocol is to find an efficient way of regularly collecting data from sensors. To this end, cluster-based tour-planning approaches have attracted much attention. In CADC, as a cluster-based scheme, our primary goal is to find a subset of sensors as Cluster Heads (CHs) to cover affiliated sensors within limited number of hops and then to find a short tour for AUV to visit each CH in a certain order. Each cluster head can buffer all collected data from its cluster members and upload them to AUV when it arrived within a single-hop vicinity of the CH. In order to perform this task, our scheme is divided into three phases: discovery phase, clustering phase, and data gathering phase. Fig. 1 shows an example of mobile data gathering technique in which sensors are clustered with a maximum of two hops distance from a CH.

During the initial phases of discovery and clustering, underwater sensors should discover neighbouring nodes to form the required clusters. The number of transmission hops that connects nodes to the cluster heads has a crucial role in the energy consumption.

In UWSNs, relay hop count should be bounded due to a number of reasons. First, underwater environment is very noisy which can increase the chance of packet failure by forwarding a packet over several hops [1]. Second, energy efficiency can be achieved by limiting the number of packet transmissions. Third, there is a limitation on the sensor buffering capacity. Thus, it is not practical to allocate a high number of sensors to a CH for local data aggregation.

The relay hop bound, d , is a system parameter which can be set based on the application priorities on the energy saving and delay. For delay-tolerant applications, d is set to a small value to save more energy at sensors. Mobile data-gathering protocols are usually suitable for the applications which are almost delay-insensitive [13].

For the clustering and tour planning, we first solve the problem using an optimal algorithm, and then an efficient

algorithm is presented in order to solve the problem in a less computational time. In our heuristic scheme, CHs are selected using a greedy iterative search manner.

The data-gathering phase has the largest share of the network energy consumption. However, the performance of this phase is mostly dependent on the efficiency of clustering phase. During the data-gathering phase, AUV continuously starts its tour from a static sink, which can be placed anywhere on the surface, collects data packets from CHs and then returns to the static sink to forward all gathered data.

B. Discovery phase

In discovery phase, neighbouring information should be exchanged between sensors to be used during clustering phase. Each sensor is required to obtain and maintain neighbouring information. To this end, each sensor should have a neighbouring table to maintain the IDs of neighbouring sensors within its transmission range.

In the discovery phase, each underwater sensor broadcasts a control packet including the packet type and sensor ID. Upon receiving a control packet, each receiving sensor updates its neighbouring table based on the newly discovered sensor. Each receiving sensor also measures its relative distance to the sending sensor via the difference between the initial and received signals strengths and keeps it in the neighbouring table. The initial signal strength is known to each node as all the nodes are homogeneous in terms of the transmission power.

Broadcasting a control packet at the same time by underwater sensors may result in collisions in the network. Thus, the transmission time of each sensor should randomly be selected from a predefined interval [18]. After the information exchange, AUV needs to travel the entire sensing field to collect sensors information and marks their locations. After the exploring of all sensors, AUV returns to the static sink to upload the sensor list.

C. Optimal Clustering Algorithm

In this section, we propose a modified branch-and-bound technique to find an optimal solution for the mobile data-gathering problem. In CADC, by having one-hop adjacency information, the d -hop degree of each sensor can be calculated. To perform the branch-and-bound algorithm, having only one-hop adjacency information of all sensors is sufficient. We also need the sensors geographical coordinates to calculate the distances. These information are gathered during the discovery phase.

In the next step, the main objective is to find a subset of sensors as CHs in which the shortest path passes through them while other sensors are covered within the relay hop bound of those CHs. This problem belongs to the class of NP-hard problems. It can be proved by the fact that, by reducing the transmission range of sensors below a certain level in a way that all sensors become unreachable from each other, the travelling salesman problem (TSP) which is NP-hard can be reduced to a special case of our problem in polynomial time.

Algorithm 1 Optimal Algorithm

```

1: procedure Backtrack( $S, T, l, \text{lengthSoFar}, \text{minCost}$ )
2: if  $S$  is empty then
3:    $\text{tourLength} = \text{lengthSoFar} + \text{Distance}(T(l), T(0))$ 
4:   if  $\text{tourLength} < \text{minCost}$  then
5:      $\text{minCost} = \text{tourLength}$ 
6:     for each  $j > l$  do
7:        $T^*(j) \leftarrow \emptyset$ 
8:     end for
9:   end if
10: end if
11: if  $S$  is not empty then
12:   for each  $s_i \in S$  do
13:     Remove  $s_i$  from  $S$ , and  $T(l) \leftarrow s_i$ 
14:     Remove  $d\_neigh(s_i)$  from  $S$ 
15:      $\text{newLength} = \text{lengthSoFar} + \text{Distance}(T(l), T(l-1))$ 
16:     if  $\text{newLength} < \text{minCost}$  then
17:        $L = \text{Backtrack}(S, T, l+1, \text{lengthSoFar}, \text{minCost})$ 
18:       if  $L < \text{minCost}$  then
19:          $\text{minCost} = L$ 
20:          $T^*(l) \leftarrow T(l)$ 
21:       end if
22:     end if
23:     Reinsert  $s_i$  and  $d\_neigh(s_i)$  in  $S$ 
24:      $T(l) \leftarrow \emptyset$ 
25:   end for
26: end if
27: return  $\text{minCost}$ 
28: end procedure

```

A branch-and-bound technique to obtain the global optimal solution for this problem is presented in Algorithm 1. This recursive algorithm tries to look at all possible tours while it skips the recursive calls for those that will never find an optimal solution.

Let $S = \{s_1, s_2, \dots, s_n\}$ be a set of sensors and T be the sequence of sensors already visited (an ordered set). In the initial call, S contains all sensors and T contains only sink ($T(0) = \text{Sink_ID}$). The variable lengthSoFar is used to maintain the length of the partially constructed tour so far and minCost gives the length of the best tour found so far. In the initial call, we set ∞ for minCost and 1 for the index l .

In lines 11-12, all possible choices that can extend the partial solution constructed so far are generated. T maintains the partial solution constructed so far and S contains the list of sensors that still need to be visited. To try a new choice, an unvisited sensor from S is removed and appended to the T , and its d -hop neighbouring sensors in S are removed as well (Lines 13-14). In lines 15-16, we compute a lower bound on the length of any solutions that can be found in that branch which is called newLength . The lower bound is compared with the best solution found so far, and if this solution cannot lead to an optimal solution, we prune the branch. Otherwise, we explore the branch to look for a better solution (line 17).

The minCost is an upper bound on the length of the best tour, and it is improved every time by finding a shorter tour

(lines 18-19). Whenever a shorter tour is found, the current sensor (as a CH) is maintained in T^* which finally gives us the optimal tour (line 20). In line 23-24, we undo the selection and reinsert sensor i and its members in S , and remove sensor i from T .

When T is a complete solution and S is empty, we compute the length of the tour represented by T and compare it with minCost (lines 2-4). If T is a shorter tour, the minCost is updated, and then the existing T^* members with an index more than l are removed (lines 5-7). This is because the number of visiting points in each solution can be different so the visiting points from the previous solution should be removed (line 7) or updated (line 20).

At the end, minCost returns the length of the optimal tour, and T^* contains the optimal tour itself. After finding the CHs, each remaining sensor is able to find at least one CH in its d -hop range. Finally, each non-CH sensor is assigned to a CH with the closest distance.

If the number of sensors is n , the worst case complexity of this algorithm remains the same as that of the Brute Force which is given as $O(n!)$. The branch-and-bound technique makes the algorithm much faster; however, the time-saving depends on the order of generated tours. As the complexity of this algorithm is very high, it is obvious that this algorithm is inefficient for a high-density network. Therefore, an approximation algorithm, which do not always find an optimal solution, is required, but can obtain near-optimal solutions with less complexity.

D. Heuristic Clustering Algorithm

In this section, we propose a new heuristic algorithm to solve the problem approximately. Our proposed algorithm is a greedy randomized adaptive search which can obtain an acceptable solution in a limited amount of time. The proposed algorithm is performed iteratively, while taking advantage of randomness and greediness to obtain an acceptable solution among the local optimal solutions. The greediness provides faster convergence to a local minimum while decreasing the search time in each iteration [19]. The randomisation contributes in achieving different solutions in each local search.

Before explaining the algorithm, it should be mentioned that which criteria are used to select a solution among all the local solutions found so far. Some major features can be used to obtain a shorter tour. First, the number of CHs should preferably be the smallest under the constraint of the relay hop bound. Second, the selected CHs should be distributed close to the sink. A heuristic algorithm using these criteria is proposed in Algorithm 2.

Let $S = \{s_1, s_2, \dots, s_n\}$ be a set of sensors and Φ be a temporary set to maintain the selected CHs at each iteration. The variable numCH shows the number of CHs of the best solution found so far, and totalDist shows the total distance of all these CHs to the sink. In the beginning, their initial value is set to infinite (lines 2-3).

In our algorithm, at each iterative, a randomised and greedily biased solution is generated. The maxitr can determine the maximum number of iterations (line 4). Increasing the

Algorithm 2 CADC Algorithm

```

1: procedure CADC( $S, maxitr, \Phi$ )
2:  $numCH = \infty$ 
3:  $totalDist = \infty$ 
4: for  $k = 1$  to  $maxitr$  do
5:   Select  $\lambda$  randomly from interval  $[0, 1]$ 
6:    $Clustering(S, \lambda, \Phi)$ 
7:   for each  $s_i \in \Phi$  do
8:      $dist = dist + Distance(s_i, Sink)$ 
9:   end for
10:  if  $|\Phi| < numCH$  and  $dist < totalDist$  then
11:     $numCH = |\Phi|$ 
12:     $totalDist = dist$ 
13:     $\Phi^* \leftarrow \Phi$ 
14:  end if
15: end for
16: end procedure

```

number of iterations can lead to exploring a better solution. λ is a real parameter which receives a random value from interval $[0, 1]$ (line 5). Then, the clustering function is called to generate a randomised and greedily biased solution in which some sensors are selected as the CHs while the rest can be assigned to them as members subject to the relay hop bound constraint (line 6).

After finding any solution, the total distances of all CHs to the sink is calculated, and its value is stored in the variable $dist$ (lines 7-9). If the number of CHs of a solution is less than the $numCH$ and $dist$ is less than the $totalDist$, this solution is maintained as a better solution in Φ^* , and the values of $numCH$ and $totalDist$ are updated accordingly (lines 10-13).

In the clustering procedure, at each step, a sensor is randomly selected from the candidate sensors with a high d -hop degree value, and other sensors that are in d -hop adjacency to the selected sensor, are removed from the candidate list. The clustering procedure is shown in Algorithm 3. Let Φ be the list of CHs which is initially set to empty and C be the set of candidate sensors which contains all sensors at the beginning (lines 2-3). The variable l is used to maintain and calculate the number of CHs which is initially set to 0 (line 4).

The clustering procedure continues as long as the C is not empty (line 5). At each step, a sub-graph induced by C is constructed and maintained in $G(C)$ to be used to recalculate d -hop degree of all candidates (lines 6-9). Let d_{min} and d_{max} be the minimum and maximum d -hop degree of $G(C)$, respectively (lines 10-11). All candidate sensors with higher d -hop value than $d_{min} + \lambda(d_{max} - d_{min})$, are placed in a Restricted Candidate Set (RCS). One sensor is randomly selected from RCS and added to Φ . The selected sensor and its d -hop neighbours are also removed from C (lines 12-16). The procedure ends when C is empty, and it gives us Φ as a solution which its optimality is compared in the main function (Algorithm 2). At the end, each non-CH sensor should be assigned to a CH with the closest distance.

It should be noted that Φ^* only provides us with the list of selected CHs of an optimal local solution and not the order of

Algorithm 3 Clustering Algorithm

```

1: procedure Clustering( $S, \lambda, \Phi$ )
2:  $\Phi \leftarrow \emptyset$ 
3:  $C \leftarrow S$ 
4:  $l = 0$ 
5: while  $|C| > 0$  do
6:    $G(C) \leftarrow$  Sub-graph induced by  $C$ 
7:   for each  $s_i \in C$  do
8:     Calculate  $d\_deg(s_i)$  with respect to  $G(C)$ 
9:   end for
10:   $d_{min} = \min\{d\_deg(s_i) \mid s_i \in C\}$ 
11:   $d_{max} = \max\{d\_deg(s_i) \mid s_i \in C\}$ 
12:   $RCS = \{s_i \in C \mid d\_deg(s_i) \geq d_{min} + \lambda(d_{max} - d_{min})\}$ 
13:  Select  $s_i$  at random from the  $RCS$ 
14:  Remove  $s_i$  and  $d\_neigh(s_i)$  from  $C$ 
15:   $l = l + 1$ 
16:   $\Phi(l) \leftarrow s_i$ 
17: end while
18: end procedure

```

visiting them during the data-gathering phase.

E. Data gathering phase

After the heuristic clustering phase, the main objective is to find a short path passes through all CHs. Finding such a path is considered as a Travelling Salesman Problem (TSP) which belongs to the class of NP-complete problems [20]. We therefore use a greedy heuristic approach offering comparably fast running time and still yielding near-optimal solutions [20].

Following this approach, if the number of CHs is N , a tour is gradually constructed by repeatedly adding the shortest edge to the tour as long as there is no cycle with less than N edges, or no CH with a degree more than 2. The greedy algorithm can be summarised as follows:

1. All edges should be sorted in increasing order of length.
2. The shortest edge is included in the tour if
 - No early cycle is formed, and
 - No vertices has a degree of 3
3. Does tour include N edges? If no, step 2 is repeated.

The complexity of this greedy algorithm to create a path is given as $O(n^2 \log_2(n))$ [20].

Once a tour is generated by the greedy algorithm, it can be optimised using some heuristic techniques. We use 2-opt algorithm which is a basic local search algorithm to take a route that crosses over itself and convert it to a tour without any crossed line [21]. The main idea is to incrementally improve an initial tour by removing two edges from the tour, reconnecting the two paths created, and replacing the current tour with new tour if it decreases the length of the tour. This procedure is continued by swapping all possible pairs of edges in the tour until no 2-opt improvements can be found.

After the tour planning, AUV initiates the data gathering phase. It should visit each CH based on the planned tour to collect the data and return to the static sink to upload the aggregated data. Meanwhile, each sensor monitors the environment and sends data packets to its CH with a fixed data

rate. Transmitting data by sensors to their CHs is independent from AUV activity. In this way, data packets are buffered in CHs and are transferred to AUV when it arrives.

When AUV becomes close to a CH, it announces its arrival using a control packet. Then, CH starts transmitting the collected data to AUV. After collecting data from a CH, AUV travels to the next CH and repeats the same procedure until it returns to the static sink. The next round of data gathering is then initiated in a similar way.

V. EXPERIMENTAL RESULTS

In this section, the details of our simulation study and also the performance results are presented. We first compare the performance of the proposed algorithm with that of the optimal solution in a small network. Finally, we conduct extensive simulations in large networks to evaluate the performance of CADC against other existing mobile data gathering schemes, *AUV_PN* [13], and *AEERP* [11].

A. Performance Metrics

We use the following metrics for the performance evaluation of proposed algorithms.

Tour length: It is defined as the total travelled distance by AUV in each data gathering round.

Number of cluster heads: It indicates the number of CHs which should be visited by AUV during the data gathering phase.

End-to-end delay: It is defined as the average delay time taken from the moment of the creation of packets at sensors until successfully being delivered to the static sink.

Energy tax: The energy tax shows the average energy consumed per message to deliver a packet to the static sink successfully.

Relay hop count: It shows the average number of relay hop counts from sensors to their CHs.

Packet Delivery Ratio (PDR): It is defined as the ratio of the number of packets successfully received by the sink node to the number of packets sent by the sensors.

B. Performance comparison with optimal solution

The branch-and-bound algorithm becomes infeasible to obtain the optimal solution in large UWSNs because the problem is NP-hard. Therefore, we have managed to compare the optimal solution with the results of the proposed scheme for small UWSNs.

1) *Simulation setup:* The number of sensors varies from 10 to 40. Sensors are randomly deployed in a $500m \times 500m$ field with a fixed depth. The static sink is placed at point (0, 250, 0). The transmission range is set to 100 meters for each sensor. We consider d equal to 2 for the optimal solution and the proposed scheme. All the results are averaged over 50 runs for randomly generated topologies.

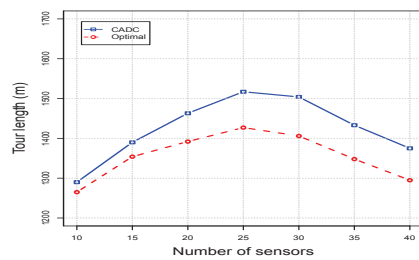


Fig. 2: Tour Length vs node density

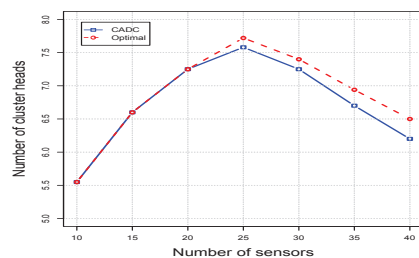


Fig. 3: Number of cluster heads vs node density

2) *Results:* We can observe in Fig. 2 that the tour length increases when the network is sparse (10 to 25 sensors) because the CHs selection is confined to the small number of sensors and consequently CHs are placed far from each other and sink. However, in a dense network (25 to 40 sensors), the tour length reduces because there is a higher chance to find the CHs with less dispersion from each other, and also more proximity to the sink. It is apparent that the length of CADC tour is very close to the optimal solution.

As can be seen in Fig. 3, the number of CHs also increases by increasing the number of sensors from 10 to 25 because the network is still sparse and more sensors have a chance to become a CH. However, by further increasing the number of sensors, network become dense and consequently fewer CHs are required to cover all sensors all over the network field. The CADC has the lower number of CHs because its main criteria to select a group of CHs is based on the minimum number of CHs.

Fig. 4 shows the average relay hop count increases when the number of sensors increases. It is because by deploying more sensors in the field, more sensors can be placed in the d -hop distance of CHs. The lower average relay hop count contributes to more energy saving.

C. CADC performance

In this section, we assess the performance of our proposed algorithm against other existing data-gathering schemes, *AUV_PN* and *AEERP*, in terms of the tour length, end-to-end delay, packet delivery ratio, and energy tax.

1) *Simulation setup:* The underwater acoustic communication channel described in Section III is used in our simulation. We deploy the sensors (ranging from 100 to 500) randomly in a two-dimensional plane $1000m \times 1000m$ at a depth of 300 m of a 3D underwater environment. The transmission

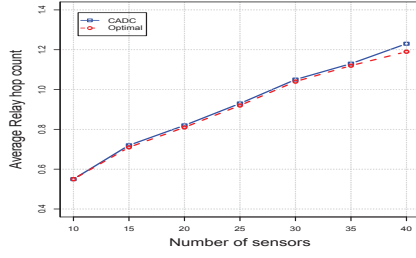


Fig. 4: Average relay hop count vs node density

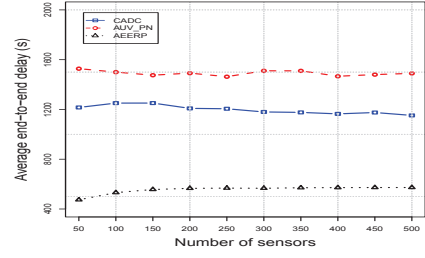


Fig. 6: Average end-to-end delay vs sensor density

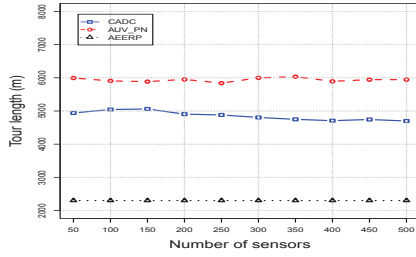


Fig. 5: Tour Length vs sensor density

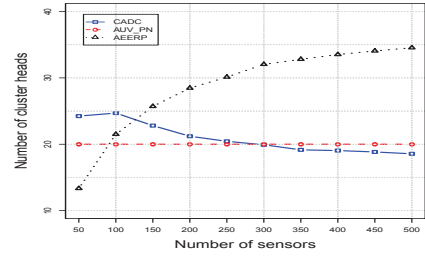


Fig. 7: Number of cluster heads vs sensor density

power and the power threshold for receiving a packet are set to $105 \text{ dB re } \mu \text{ Pa}$ and $10 \text{ dB re } \mu \text{ Pa}$, respectively. Each sensor consumes 50 W and 0.158 W energy for sending and receiving a packet, respectively, while the idle power consumption is 0.008 W . The signal frequency f_{data} is set to 20 kHz and the acoustic signal propagation speed is 1500 m/s . The transmission range of each sensor is considered as 100 meters.

Each sensor generates a data packet every 100 seconds. The bit rate is set to 10 kbps , and the data packet size is fixed at 1024 bits. The static sink is placed at the corner of the network topology with $(0, 500, 0)$ coordinates. The speed and depth of AUV are set to 4 m/s and 250 m , respectively. We consider d equal to 2 for CADC. All the results are averaged over 50 runs for randomly generated topologies while the simulation time for each run is set to 12 hours.

2) *The impact of sensor density:* In this set of simulations, the impact of sensor density on performance metrics are examined. We change the number of sensors from 50 to 500. The results for the tour length, end-to-end delay, number of cluster heads, packet delivery ratio, and energy tax are plotted in Figs. 5, 6, 7, 8, and 9, respectively.

From Fig. 5 and 6, we can see that the tour length and the average end-to-end delay of AUV_PN is higher than CADC. This is because, in AUV_PN, AUV starts its tour from the static sink and travels to the closest CH to obtain the list of Path Nodes (PNs) and then visits them to collect data packets; however, it is not much focused on shortening the length of the tour. The constructed tour by AUV_PN is a tour with so many crossed lines which increases the tour length. However, in CADC, the tour is constructed using a greedy approach and optimised by a 2-opt algorithm before the AUV travelling. The AEERP tour is an elliptical path which is short and fixed; however, it has the lowest performance in terms of packet delivery ratio and energy efficiency as can be seen in Figs. 8,

and 9.

Fig. 7 shows the number of CHs for each approach with different sensor density. Moving from 50 to 100 sensors, we can see an increase in the number of CHs for CADC because the network is not yet fully-connected. With more than 100 sensors, the network becomes dense and well-connected, and it can be covered by the lower number of CHs. However, in AEERP, the number of CHs is increased when the network becomes dense because more sensors can be distributed around the elliptical path and become a CH. Finally, it can be observed that AUV_PN always has the fixed number of CHs (including PNs).

From Fig. 8, we can see that CADC can obtain a higher packet delivery ratio compared to other approaches. This is because all the packet transmissions are bounded to $d = 2$ hops. However, in AUV_PN, there is no bound for the distance between the members and PNs. Therefore, the packet failure is higher because of the path loss or acoustic channel attenuation over a longer distance. In AEERP, there is also no hop bound limitation from the members to the CHs, and it is obvious that the packet failure probability is high when the number of relays increases. Another issue is that the void area can occur between some members and CHs when the network is sparse. Thus, the packet delivery ratio in AEERP is very low when the network is sparse.

Fig. 9 shows the average energy consumed per message in each protocol. It is observed that the energy consumption of CADC is considerably less than AEERP. It is because the number of transmissions is significantly reduced by bounding the relay hop to lower values. In AEERP, when the network is sparse, the energy is wasted due to the void problem. When the network is high density, the void problem mostly disappears, but the boundless hop distance between the members and CHs still keeps its energy consumption higher than other approaches.

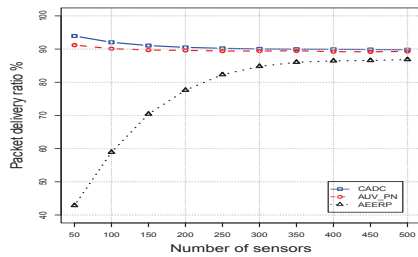


Fig. 8: Packet delivery ratio vs sensor density

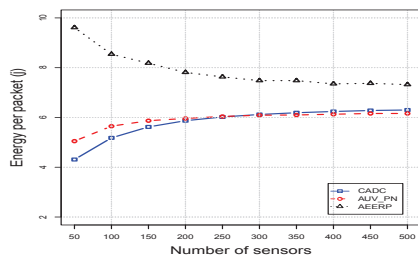


Fig. 9: Energy consumption per packet vs sensor density

In addition to having a shorter tour, CADC consumes less energy than AUV_PN, when the number of sensors varies from 50 to 300. This is because, in AUV_PN with less than 300 sensors, the number of cluster heads is less than CADC. Thus, sensors are placed in a longer distance to CHs which require them to send their packets with a higher power. The larger number of CHs results in a shorter distance between sensors and CHs, and therefore saving more energy. With more than 300 sensors, CADC consumes a little bit more energy than AUV_PN to trade off energy against lower latency as shown in Fig. 5. Moreover, in AUV_PN, some CHs do not participate in data collecting, and AUV only visits them to obtain the list of PNs. Thus, the number of CHs which participate in local data collecting is less than the actual value.

VI. CONCLUSION

In large-scale UWSNs, where multi-hop routing is quite a challenging task because of adverse channel conditions and the presence of void areas, an AUV-aided mobile data collection mechanism seems to be necessary to maintain the network performance. In this paper, we proposed a mobile data-gathering scheme for UWSNs by exploiting a trade-off between energy and data gathering latency. In the proposed scheme, called CADC, a group of sensors are selected as CHs to collect data locally from their members. A near-optimal tour is then planned by AUV to visit all those CHs to gather data packets and upload them to a static sink on the surface. CADC is highly scalable and also applicable in both connected and disconnected networks. The simulation results illustrated that CADC can shorten the tour length while maintaining the relay hop bound, resulting in a better trade-off between energy and data-gathering latency.

REFERENCES

- [1] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, "Void-handling techniques for routing protocols in underwater sensor networks: Survey and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 800–827, 2017.
- [2] F. Alfouzan, A. Shahrabi, S. M. Ghoreyshi, and T. Boutaleb, "Efficient depth-based scheduling mac protocol for underwater sensor networks," in *Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference on*, pp. 827–832, IEEE, 2017.
- [3] F. Alfouzan, A. Shahrabi, S. M. Ghoreyshi, and T. Boutaleb, "Performance comparison of sender-based and receiver-based scheduling mac protocols for underwater sensor networks," in *Network-Based Information Systems (NBIS), 2016 19th International Conference on*, pp. 99–106, IEEE, 2016.
- [4] S. Yoon and C. Qiao, "Cooperative search and survey using autonomous underwater vehicles (auvs)," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 364–379, 2011.
- [5] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, "An underwater routing protocol with void detection and bypassing capability," in *31st International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan*, pp. 530–537, IEEE, 2017.
- [6] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, "An inherently void avoidance routing protocol for underwater sensor networks," in *Twelfth International Symposium on Wireless Communication Systems, Brussels, Belgium*, pp. 1–6, IEEE, 2016.
- [7] J. U. Khan and H.-S. Cho, "Data-gathering scheme using auvs in large-scale underwater sensor networks: A multihop approach," *Sensors*, vol. 16, no. 10, p. 1626, 2016.
- [8] N. Javaid, N. Ilyas, A. Ahmad, N. Alrajeh, U. Qasim, Z. A. Khan, T. Liaqat, and M. I. Khan, "An efficient data-gathering routing protocol for underwater wireless sensor networks," *Sensors*, vol. 15, no. 11, pp. 29149–29181, 2015.
- [9] M. Akbar, N. Javaid, A. H. Khan, M. Imran, M. Shoaib, and A. Vasylakos, "Efficient data gathering in 3d linear underwater wireless sensor networks using sink mobility," *Sensors*, vol. 16, no. 3, p. 404, 2016.
- [10] Y.-S. Chen and Y.-W. Lin, "Mobicast routing protocol for underwater sensor networks," *IEEE Sensors Journal*, vol. 13, no. 2, pp. 737–749, 2013.
- [11] A. Ahmad, A. Wahid, and D. Kim, "Aeerp: Auv aided energy efficient routing protocol for underwater acoustic sensor network," in *Proceedings of the 8th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, pp. 53–60, ACM, 2013.
- [12] S. Yoon, A. K. Azad, H. Oh, and S. Kim, "Aurp: An auv-aided underwater routing protocol for underwater acoustic sensor networks," *Sensors*, vol. 12, no. 2, pp. 1827–1845, 2012.
- [13] J. U. Khan and H.-S. Cho, "A distributed data-gathering protocol using auv in underwater sensor networks," *Sensors*, vol. 15, no. 8, pp. 19331–19350, 2015.
- [14] U. Lee, P. Wang, Y. Noh, F. Vieira, M. Gerla, and J.-H. Cui, "Pressure routing for underwater sensor networks," in *29th conference on Information communications (INFOCOM), San Diego, California, USA*, pp. 1676–1684, 2010.
- [15] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, "A novel cooperative opportunistic routing scheme for underwater sensor networks," *Sensors*, vol. 16, no. 3, p. 297, 2016.
- [16] V. Chandrasekhar, W. K. Seah, Y. S. Choo, and H. V. Ee, "Localization in underwater sensor networks: survey and challenges," in *Proceedings of the 1st ACM international workshop on Underwater networks, Los Angeles, USA*, pp. 33–40, ACM, 2006.
- [17] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 34–43, 2007.
- [18] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, "An opportunistic void avoidance routing protocol for underwater sensor networks," in *30th International Conference on Advanced Information Networking and Applications, Crans-Montana, Switzerland*, pp. 316–323, IEEE, 2016.
- [19] S. Babadi, R. Ramirez-Iniguez, T. Boutaleb, and T. Mallick, "Novel non-imaging optic design for uniform illumination," in *Complex Light and Optical Forces X*, vol. 9764, p. 976416, International Society for Optics and Photonics, 2016.
- [20] J. J. Bentley, "Fast algorithms for geometric traveling salesman problems," *ORSA Journal on computing*, vol. 4, no. 4, pp. 387–411, 1992.
- [21] M. Bellmore and G. L. Nemhauser, "The traveling salesman problem: a survey," *Operations Research*, vol. 16, no. 3, pp. 538–558, 1968.