RESEARCH   **Open Access**

A scalable global positioning system-free localization scheme for underwater wireless sensor networks

Mohammed Aquil Mirza[1*], Muhammad Zeeshan Shakir[2] and Mohamed-Slim Alouini[3]

Abstract

Seaweb is an acoustic communication technology that enables communication between sensor nodes. Seaweb technology utilizes the commercially available telesonar modems that has developed link and network layer firmware to provide a robust undersea communication capability. Seaweb interconnects the underwater nodes through digital signal processing-based modem by using acoustic links between the neighboring sensors. In this paper, we design and investigate a global positioning system-free passive localization protocol by integrating the innovations of levelling and localization with the Seaweb technology. This protocol uses the range data and planar trigonometry principles to estimate the positions of the underwater sensor nodes. Moreover, for precise localization, we consider more realistic conditions namely, (a) small displacement of sensor nodes due to watch circles and (b) deployment of sensor nodes over non-uniform water surface. Once the nodes are localized, we divide the whole network field into circular levels and sectors to minimize the traffic complexity and thereby increases the lifetime of the sensor nodes in the network field. We then form the mesh network inside each of the sectors that increases the reliability. The algorithm is designed in such a way that it overcomes the ambiguous nodes *errata* and reflected paths and therefore makes the algorithm more robust. The synthetic network geometries are so designed which can evaluate the algorithm in the presence of perfect or imperfect ranges or in case of incomplete data. A comparative study is made with the existing algorithms which proves the efficiency of our newly proposed algorithm.

1 Introduction

Seaweb network is a wide area network that interconnects the static (sensor nodes, buoys, etc.) and mobile wireless nodes (unmanned underwater vehicles submarines), by employing digital signal processing-equipped telesonar modems and through acoustic links, in the undersea applications [1,2]. Seaweb network basically consists of three main components namely, (a) Seaweb server, (b) gateway buoy (GB), and (c) repeater/sensor nodes [1]. The Seaweb server is located either on the ship or ashore which is in turn controlled by the operator to monitor and control the undersea deployed network. The GB acts as a centralized node which communicates with all the sensor nodes and thereby improves the lifetime of the network. The GB is the only mode of direct communication

from the sensor nodes to the server. If there is no direct radio communication between the GB and the nodes, then satellite link acts as an alternative way of communication. The GB is fixed at a suitable position and anchored at the sea/ocean floor such that it has a direct line of sight (LoS) with the Seaweb server. The buoy is attached to the radio/satellite communication equipment and solar panels at the surface. Underwater sensor nodes that are usually anchored/moored at the seabed consist of a clamp weight, acoustic release, telesonar modem, and subsurface float [1,3].

In [1,2,4], inter- and intra-distances between parent (gateway buoy) and child nodes are obtained as a byproduct of telesonar signalling and navigation of mobile nodes such as submarines. Link-layer methods like handshaking, selective repeat request (SRQ), and forward error correction provide the network reliability. A network layer mechanism such as maintaining the routing table, feedback packets, and packet serialization enhances

*Correspondence: aquil.mirza@pilani.bits-pilani.ac.in
[1]Electrical and Electronics Engineering Department, Birla Institute of
Technology and Science (BITS), Pilani, Rajasthan, 333031, India
Full list of author information is available at the end of the article

the quality of service. This paper uses node-to-node ranges, compiled at parent node, to solve the problem of localization. Synthetic data analysis with different network geometries helps in evaluating the protocol with imperfect ranges, or insufficient data.

In [5,6], a technique incorporating the time synchronization between the two nodes has been proposed. The existing algorithms does not consider the effect of realistic condition namely, (a) reflected paths, (b) watch circles, and (c) node displacement, which increases the localization error exponentially. However, the complexity of this scheme is exceptionally high, which may result in poor energy efficiency and reduces the lifetime of the sensor nodes. Moreover, this scheme is very difficult to extend to larger networks. In particular, the number of beacon nodes is fixed and set to three only in the paper. However, an exceptional increase in computational complexity is expected with the addition of even one more beacon node to the network. Moreover, the addition of beacon node change/update in program for both beacon and ordinary nodes is a must. However, we overcome the problem of reflected paths and watch circles in our scheme, and the proposed scheme does not require time synchronization in the network. In [5], the network calculations are implemented by using the ordinary sensor nodes, and we reduce this complexity by incorporating Seaweb/GB communication between sensor nodes and Seaweb server.

In a recent work [6], the authors introduce a localization scheme designed for large network of underwater wireless sensor networks. The proposed localization scheme depends on time-difference of arrival measurements calculated locally at a sensor to measure the range differences from the sensor to three anchors that can overhear each other. However, it suffers from two disadvantages. One of these is the computational overhead; in their proposed scheme, all of the calculations are implemented by ordinary nodes, and since the complexity of this scheme is not low, it may result in poor energy efficiency. Second, this scheme is very difficult to be extended to larger networks and hence is not scalable.

In [7], an application of sensor deployment in the Unet'08 Seatrial is studied. A variable speed of sound profile is considered which increases the localization error. This scheme does not take into consideration the displacement of the nodes over the sea surface. We overcome this problem by using MunkâĂŹs canonical formula [8] and the finite difference linearization and law of cosines. The paper is organized as follows. In Section 2, we give a brief overview of the Seaweb network. We provide more suitable solution and a reliable algorithm by taking into consideration these environmental factors.This section explains the network layout and the type of protocols being used. Section 3 describes our proposed localization algorithm and discusses the types of constraints being

imposed and its implementation. Section 4 describes the simulations and comparisons of the proposed scheme with the traditional algorithms and finally, Section 5 concludes our paper.

2 Seaweb network description

2.1 Network layout

Seaweb network consists of number of sensor/repeater nodes and GBs. The sensor nodes are interconnected through each other, thereby forming a mesh grid. The paths between each of the sensor node are reconfigurable, if there is a route failure due to the death of the particular sensor. The GB is positioned to maximize the lifetime of a particular node, i.e., away from the traffic lanes, out of ship. This GB acts as a parent node and also as a relay between the nodes and the server. The term parent is often used to describe the GB as it is localized at the desirable position or it can be used as point of reference (say (0, 0)). Once the GB is deployed, we deploy the first sensor node such that it acts as a second point of reference for the localization purposes. This reference node is referred to as ref_x , which is deployed along the line of bearing from the parent node. This allows fixing the location of these two nodes in order to start localization process. The remaining interested nodes in the network are deployed randomly or uniformly as per interest of the operator. The ranged data are stored into a time-stamped matrix, where these matrices are complied into time sequenced stacks. This matrix is then statistically analyzed for all the measured ranges between each of the neighboring nodes, thereby eliminating all the ranges that fall outside a confidence interval of 5% from the mean. These filtered ranges are then used to localize each of the nodes into *x-y* position. We then form levelling and a mesh network inside levelling, which allows nodes to have multiple communication paths and in turn increases the reliability. The distance between each of the node is varied from 1 to 5 km depending upon the deployment and the density of the network.

2.2 Network protocol

Seaweb uses a link-layer protocol for communicating from one node to another for addressing, ranging, and power control. A SRQ protocol is used for the delivery and acknowledgment of message units [9]. In this context, the nodes use the request to send (RTS) and clear-to-send command to establish the connection between them before transmitting the data. The same is depicted from Figure 1 where the range between the nodes is found by using the PING and ECHO commands (supported by SRQ), which are generated by using the hyperbolic frequency modulated (HFM) signal [10]. The underwater modulated frequency is ranged from over a frequency of 9–14 kHz. When a node *x* PING to its neighboring node *y*, node y determines the time of arrival and angle of arrival
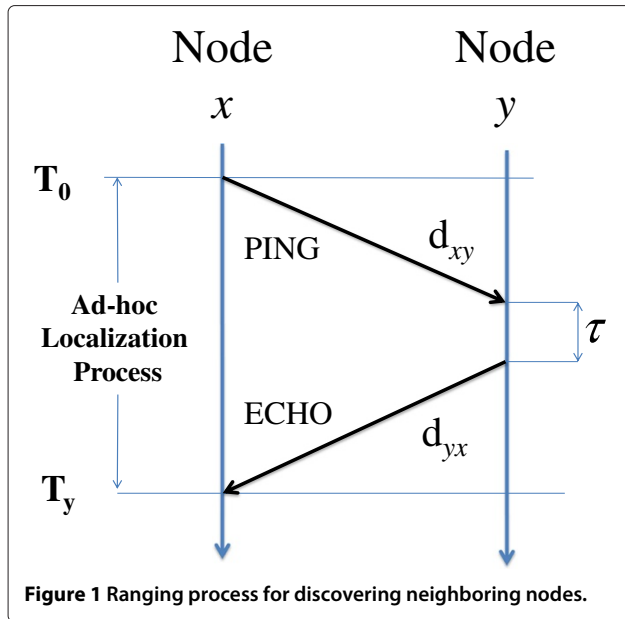
Figure 1 Ranging process for discovering neighboring nodes.

(AoA) by using the peak of the HFM-matched filter. Node y waits for a specified time (say τ), before sending ECHO to node x, as shown in Figure 1. This follows the same path based on reciprocity. Henceforth, the sound propagation delays between these nodes are equal ($d_{xy} = d_{yx}$). Once the ECHO is received by node x, Seaweb modem calculates the range, r_{xy} based on the total time delay as follows

$$r_{xy} = Cd_{xy} \tag{1}$$

where $d_{xy} = (T_y - T_0 - \tau)/2$ and T_0 is the initial time the localization process starts, T_y is the end time of localization between nodes x and y, C is the speed of sound, and d_{xy} is the distance between node x and node y.

As the depth of sea/water increases the speed of sound varies [11]. Since we use the repeater nodes which are anchored at the sea level we consider the speed of sound to be constant at 1,520 m/s. However, we consider the cases where the speed of sound varies due to uneven surfaces at the sea bed.

2.3 *Ad hoc* discovery process
Once the network is deployed, the first task is to perform the node-to-node ranges (using PING and ECHO) starting from the parent node and whereby these ranges are maintained at the Seaweb server. Then the parent node initiates a PING and commands each of its discovered nodes to conduct the broadcast PING in order to discover the remaining nodes. These nodes thereby maintain the mesh or a route to which they can communicate in a single hop.

3 Seaweb network passive localization
Localization in Seaweb server of repeater/sensor nodes is done by using range-based intersecting circles of the known nodes. When an unknown node (non-localized) comes in contact with two known nodes (localized nodes), it is localized in its *x-y* plane by using the range-based algorithm. In [12,13], the weighted averages and center of mass method have been introduced which does not eliminate range errors fully. The finite difference linearization method is proved to be more efficient compared to other approaches [6]. The acoustic localization/communication is affected by many environmental factors like reflected paths, transmission losses and speed of sound. These factors have been studied in much detail in [1,2,4]. Based upon the range measurements between the known and unknown sensor nodes, there are some constraints which we describe in the following section.

3.1 Acoustic localization constraints
The localization depends upon the topology of the deployed network and the acoustic nature of the environment. Depending on these scenarios, we can have the possible three cases namely; no-solution, which occurs when there is only one known node range is available for localization; secondly, many solutions, where one unknown node has two ranges to know known nodes, i.e., there are two ambiguous outcomes that exist when two range circles intersect. However, this can be resolved by using AoA [5] and law of cosines [14]; finally, finite difference linearization is applied, where one unknown node has three ranges from known nodes, such that a single, unambiguous solution exists [14].

Acoustic communication does not preserve the variability of different sources of error. Therefore, we need to consider these errors in order to design a reliable algorithm under non-ideal conditions. Some of the errors are mentioned as below, which has been taken into consideration while designing the algorithm. However, Appendix discusses how these errors are taken into consideration in implementation of algorithm.

1. *Speed of sound.* It usually varies from 1,480 to 1,520 m/s, i.e., +1.3% or − 1.3%. We presumed the speed of sound as 1,520 m/s. We also assume that sound travels in a straight line and hence is linearly related to range. However, the temperature, pressure, and salinity in the water cause some deviation to the speed of sound which results in refraction. Hence, we use a variable sound speed profile by employing Munk's canonical profile [8] which is defined as

$$M(z) = C[1 + \varepsilon(e^{-\vartheta} + \vartheta - 1)]\,\text{m/s} \tag{2}$$

where $\vartheta = [2(z - z_{\text{axis}})]/S$, C is the speed of light, ε is the perturbation coefficient, z_{axis} is the depth of

sound channel, z is the depth of water in meters, and S is the scale depth and ϑ is the dimensionless distance beneath the sound channel axis. The values are as defined, $z_{\text{axis}} = 1,000$ m, $S = 1,000$ m, $\varepsilon = 0.0057$ and $C = 1,520$ m/s.

2. *Transmitted/reflected paths.* Reflected paths result into absorption and scattering of the transmitted signal that reduces the intensity of sound energy. For the surface, the reduction depends on the roughness of sea surface and the spectral frequency of the transmitted signal [2]. Losses at the bottom of sea are of the order 8 dB per interaction [2,4]. However, our algorithm takes into consideration these losses when determining the node-to-node ranges by using peak detector filter which allows selecting the highest peak multi-path arrival.

3. *Node position or depth variance.* It is possible that two nodes are located at uneven surfaces and both of them have a horizontal and a slant range. The horizontal and slant range differences are considered in our algorithm whereby we presume that all nodes cannot be located in same horizontal plane.

4. *Current/waves circles.* It is possible that tides replace the fixed/stationary transducers and thereby causing deviations in the measured ranges and positions of the nodes. These rotations or replacements are called watch circles. Since the transducers are 3 m off the bottom there is a slight variation in the range difference of two nodes [2].

5. *Dilution of precision.* The large uncertainty between the distance of two nodes is referred to as geometric dilution of precision (DOP) or GDOP. It occurs when the distance between the two referenced nodes is very close. However, we tackle this problem by using finite difference linearization.

3.2 Localization algorithm
Localization using the Seaweb server consists of basically five steps namely, server input, data input, error correction, levelling of the network, mesh grids and finally, determination of location of nodes. The pseudocode of our localization algorithm is shown as Algorithm 1.

Step 1. In the first step, the Seaweb server initiates the request to parent node positioned at (0,0) or gateway buoy to collect all the measure range data with respect to the ref_x node at position (500,0). The ref_x node is placed such that it is in the range of gateway buoy so that the server may have the true bearing between these two nodes. The operator (at Seaweb) then maintains a table which specifies all the measured ranges from repeater nodes. This step is carried out only at the initial stage and during the second stage (say after watch circles), the localization

---

Algorithm 1 A passive GPS-free localization algorithm

Input: set of randomly deployed repeater/sensor nodes
Output: localization/mapping of nodes-to-nodes
foreach *task m* **do**
  foreach *sensor i* **do**
    Calculate range using SRQ(i,j);
    if *LoS(i,j) ≠ LoS(j,i)* **then**
      $i = \theta j$;
      $j = \theta i$;
while *non-localized sensor remaining* **do**
  foreach *sensor m* **do**
    Calculate through planar trigonometry, PT(i,m);
    if *PT(i,j) ≤ PT(i,m)* **then**
      $m = \theta j$;
  Mapping(sensor i on sensor j) ;
for *sensor field* **do**
  Levelling in the sensor field;
  if *Packet from parent node* **then**
    Set level, L_1 ;
    Continue until level, L_m
for *sensors inside the levels* **do**
  mesh grids.

---

is carried out based on node-to-node ranges. This resolves the problem of symmetrical ambiguities.

Step 2. Once the table is ready the data is sorted into a 3D matrix which contains all the measured ranges. This matrix consists of number of layers equal to the number of range files uploaded by the operator. Each of these layers is a square matrix, where the nodes addresses are placed in the first row and first column of the matrix. Once all the nodes are stacked into the matrix, it is processed for the range errors.

Step 3. In the third step, we analyze the normalized mean error ranges between each pair of nodes. Here each pair of nodes is statistically evaluated which is explained in Appendix, and the results are stored into a 2D array. We try to put all the ranges of a particular node pair into a single vector, which is obtained from the stacked matrix. Simultaneously, the reciprocal combination (say from node y to node x) is obtained by looking 'up the stack.'

Step 4. The obtained vector is statistically analyzed between the selective nodes thereby eliminating the error ranges that fall outside 5% confidence interval (CI) of the data ranges. Initially, the data that fall outside the 5% CI is noted and its mean is calculated. The datum falling farthest from the mean is eliminated. The mean and CI are then recalculated and the process is repeated until the datum does not fall outside the CI. This final estimate is then stored in a 2D array. This output array is a 2D-square array

which is diagonally symmetrical and with node pairs that lack range estimate which is denoted as no-range or NR.

Step 5. This step divides the entire network into circular levels. This is initiated by the parent node. The parent node sends a packet to all its neighboring nodes which are one hop distance away. The nodes receiving the packet from the parent node set their level ID as L_1. The ref_x node also has an ID of L_1. The L_1 nodes, in turn, increment the packet ID (say L_2) and transmits it to its one hop neighbors and these neighbors set their level as L_2. The process is continued until all the nodes are levelled in the network. The detailing of the process is defined in [15]. After levelling of the network, the nodes try to form the mesh grids/routes from their one hop neighbors. This leads to proactive path formation and thereby decreases the congestion. The Figure 2 shows the whole process of five-step implementation in the simulation setup.

Now, once the data has been filtered, averaged, and organized into a 2D array, we now estimate the location of the nodes. As explained earlier, the parent node is set to be the origin of a horizontal Cartesian grid. The ref_x is placed such that its x-axis is intersecting parent node, coordinates assigned to (ref_x, 0). Once this node is localized, it is considered to be a known node and with the two known nodes the process of localization can be initiated. If

there is only one range to the node, it is left isolated unless it has two or more ranges. If there are two known ranges to an unknown node, there exist two solutions. These two solutions are stacked and compared when a previously unknown node becomes known.

When an unknown node has three known ranges, and then the solution can be found by using [16];

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (x_2 - x_1) & (y_2 - y_1) \\ (x_3 - x_2) & (y_3 - y_2) \end{bmatrix}^{-1} \\ \begin{bmatrix} (r_1^2 - r_2^2) + (x_1^2 - x_2^2) + (y_1^2 - y_2^2) \\ (r_2^2 - r_3^2) + (x_2^2 - x_3^2) + (y_2^2 - y_3^2) \end{bmatrix}, \quad (3)$$

where $[r_i^2] = [(x - x_i)^2 + (y - y_i)^2]$.

If the node is evaluated for the first time then, we store solution in the stack and loop through all nodes calculation to determine if it is involved in any possible ambiguous solution sets. If the node is not localized for the first time, then it check with the previous solution. If their x and y coordinates are greater than some threshold difference, then the new solution is taken into consideration. This process is continued until all the nodes are perfectly localized into the network.

3.3 Routing protocol

In this section, we provide a detailed implementation of routing between the sensor nodes. The basic idea in this algorithm is to divide the field into sectors and route the events by using nodes which can switch between SLEEP and WAKE modes.
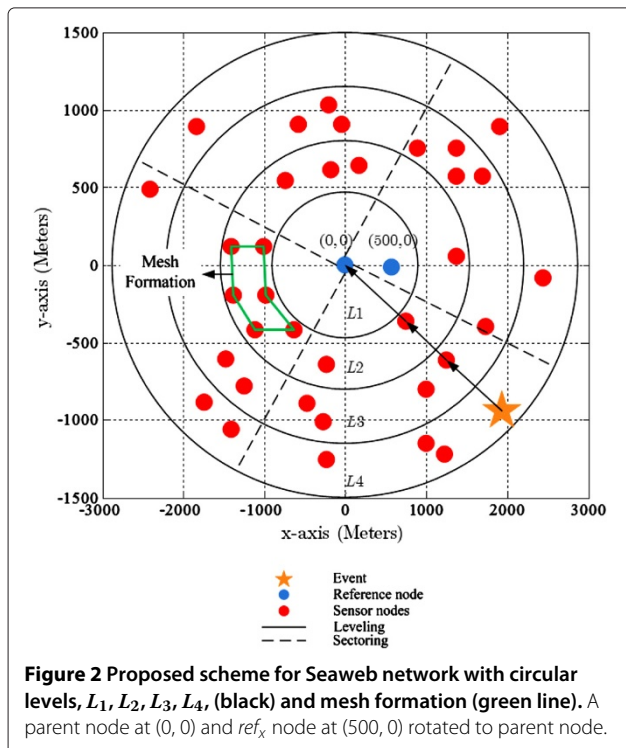
> *Step 1: levelling.* As per our assumptions, we consider a densely deployed sensor field. Initially, the base station (BS) sends signals with a minimum power level and all those sensor nodes that receive this information will set their level to 1. Then the base station will increase its power level and transmit the signal. This time those nodes which receive the signal for the first time set their level to 2. This procedure continues till all the nodes in the network have their level IDs determined. To counter the effects of fading in wireless channels, hop count-based leveling can also be done [3].
> *Step 2 : sectorization.* Using the directional antenna, the BS will send signals with maximum power and divide the sensor field in to equiangular sectors with an angle of θ (consider θ as 45°). Now, every node in the network is aware of its level and sector [8].
> *Step 3 : clustering or forming mesh.* Clusters of sensor nodes are formed based on the signal strength and use these local cluster heads as routers to sink. The optimal number of cluster heads is estimated to be 5% of the total number of nodes. The decision is made by choosing a random numbers between 0 and 1. The node becomes a cluster head for the current

Figure 2 Proposed scheme for Seaweb network with circular levels, L_1, L_2, L_3, L_4, (black) and mesh formation (green line). A parent node at (0, 0) and ref_x node at (500, 0) rotated to parent node.

round if the random number is less than the threshold value $T(n)$ which is defined as

$$T(n) = \begin{cases} \frac{p}{1 - p(r \bmod \frac{1}{p})}, \text{if } n \epsilon G \\ 0, \text{otherwise} \end{cases} \quad (4)$$

where p is the desired percentage of cluster heads (e.g., 0.05), r is the current round, and G is the set of nodes that have not been cluster heads in last $1/p$ rounds [1].

Step 4 : mode setup. The part of a sector which is in a particular level is called sector ID. If an event occurs in the level (L), these nodes flood very small metadata packets that contain the level ID and sector ID of the node where the event has occurred source node). Each node that receives this packet will read the location of source level ID (L), if the level of this level ID is L or $L2$ or $L4$ . . .. Then these nodes will go to SLEEP mode. The level IDs with level L_1 or L_3, etc. and level L go to into SLEEP mode (if and only if there is no transmission in that particular level ID). On completion of this setup, the source node floods the data packets in the direction of BS. The node that receives the packet checks for two conditions: one is for the level ID and the other is for the sector ID. If the data is from higher level, it only checks for sector ID. If the packet is from neighboring sector of higher levels, then the packet is forwarded, and in the other case the packet is discarded.

The proposed algorithm follows flowchart procedure as shown in Figure 3:

1. When an event occurs at a node, the node floods the data packets to every neighbor.
2. Only the nodes which are in WAKE mode will receive the packet, and nodes in SLEEP mode do not receive the packets.
3. Then the nodes that receive data packets check the level ID and sector ID of the packet.
4. If the level ID from the source is lesser than its level ID, the packet is dropped.
5. If the level ID from a source is larger, then the node checks whether the sector ID is from neighboring sectors; i.e., which are at one hop distance. If not, the packet is dropped.

4 Simulation results

In this section, we simulate our scheme and then compare the same with the existing algorithms using MATLAB (Mathworks Inc., Natick, MA, USA). We randomly generate a 70-node layout on a $10,000 \times 10,000$ m grid, placing the parent node as reference frame axis and ref_x node as $(x, 0)$. The true node locations and range-based locations of few selected nodes are given as in Table 1.

For realism, ranges exceeding range of 9,000 m are neglected which, in turn, are calculated through finite linearization method to delimit the limitations of acoustic communications and to maximize the number of ranges evaluated in each pass. The analysis is carried out for ten realizations of the 70 node networks, resulting in 91,466 ranges being evaluated. The percentage errors between the actual ranges and ranges with offset are recorded. The percentage error between the output and actual ranges is then compared.

The mean percentage error for the range that is not run through stacks is 5.6%, and the largest percent error is 5.96%. After the ranges are stacked, the mean error is reduced to 3.5% and the largest percent error is 4.1%. In absolute sense, the accuracy of 10,000 m range estimate is improved by 96.7 m.

Firstly, we perform an error-free testing for the algorithm to verify the ability of the proposed scheme to correctly determine the positions of all the nodes in the network. Several statistics are calculated for different network realizations. The node locations and ranges are shown in Table 1. These node locations are calculated by using the ranging process in order to discover the neighboring nodes. The most important statistic is how well the algorithm is able to localize the nodes when compared to the actual positions. This is done by finding the normalized mean error for each of the particular node. The mean nodal localization error is 4.6 m with a maximum of 225 m. In [7] a fixed speed of sound profile increases the localization error. Node displacement velocity is assumed to be constant, which gives the actual throughput for real time deployment. Whereas authors in [5] uses the time synchronization between the two nodes without considering the notion of reflected paths, which increases the error exponentially. Due to watch circles, the displacement occurs very frequently and the synchronization between the nodes are needed to be carried out at regular interval of times. Figure 4 shows the comparative study made with our proposed scheme. Our scheme performs well and has least localization error as the number of nodes are increased. This is mostly due to the fact that we consider the node displacement at fixed regular intervals. We also frequently update the node and location range estimates as shown in Table 1.

Secondly, we induce the errors using (6) by using the fixed mean and variances. The goal is to study the performance of the network when the errors are induced randomly at random nodes. In Figure 5, as expected, the performance decreases in case of error-induced ranges. The mean number of nodes localized is 61.5 nodes. The number of iterations required for the 70-node network to get localized is shown in Figure 6. The mean number of iterations for error-free ranges is 45.8, whereas the mean number of iterations required for error induced ranges

Figure 3 Flowchart of routing algorithm.
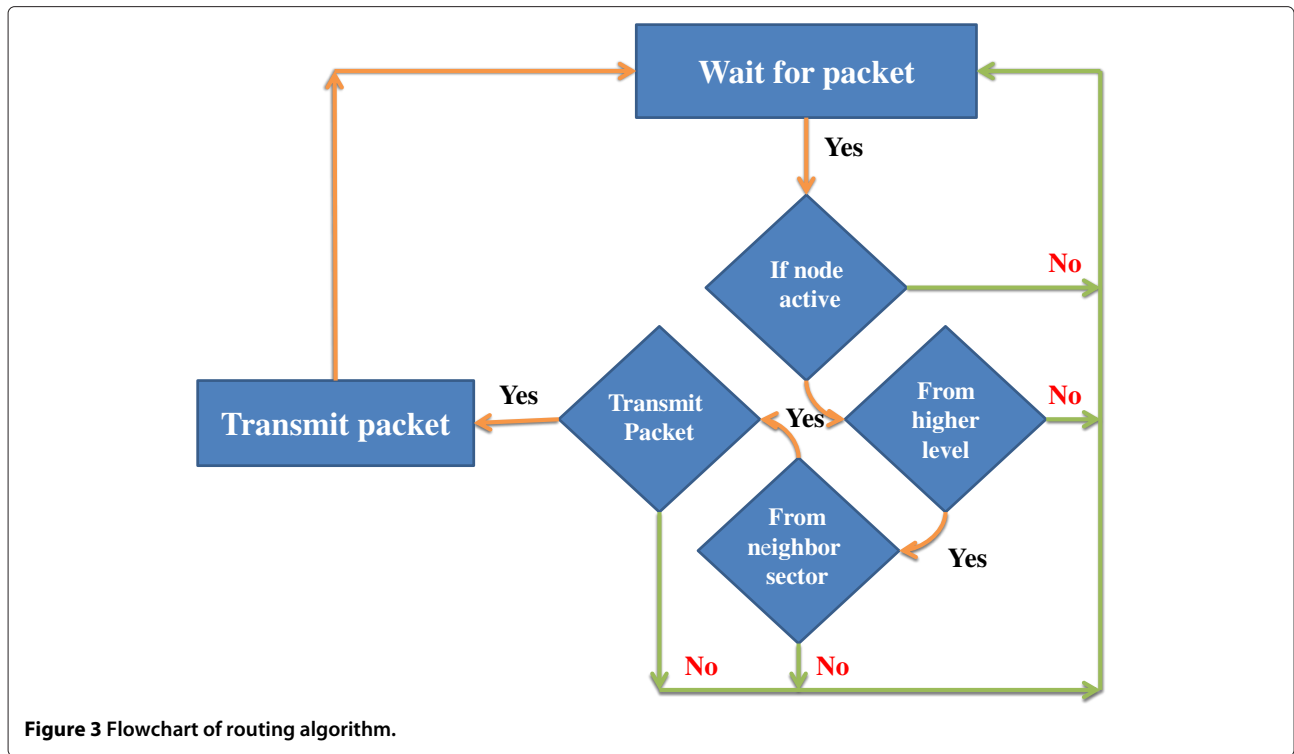
is 43.78 m. The number of iterations required for localization is reduced since we do the localization cluster by cluster or mesh by mesh.

It is observed that using 70-node network gives a localization error of 4 m (with error-free ranges) and an error of 3.5 m with error-induced ranges. It is unclear at this point why range-induced errors improves the error at 70 nodes when the error-free ranges predict a higher error. It

is presumed that the maximum number of allowed iterations and acceptance criteria for the algorithm to not flag an ambiguous solution for another iteration play major roles.

The next figure, Figure 7, shows the number of events detected by the number of the randomly scattered nodes in the network field. An event is any activity detected by the sensor nodes.
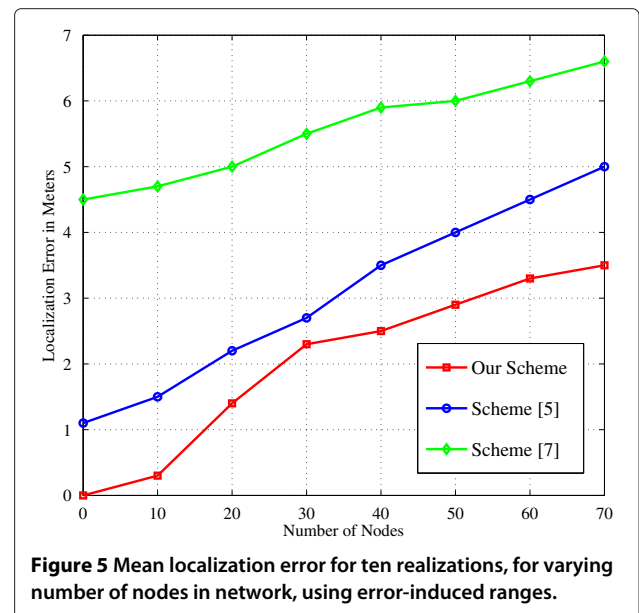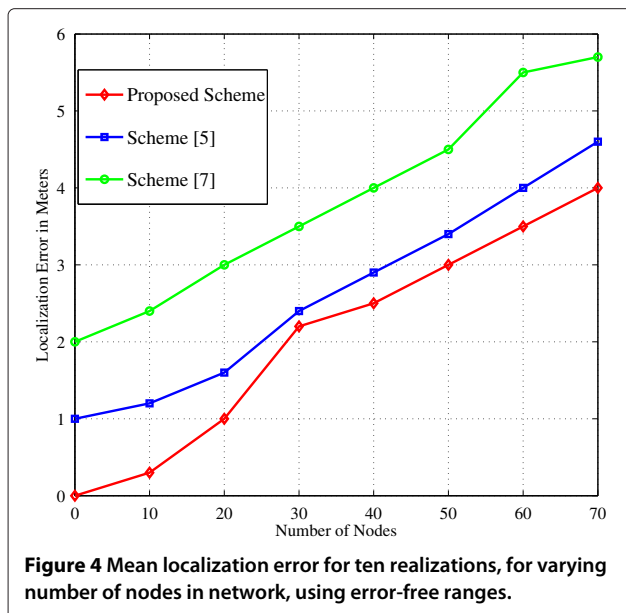

Figure 4 Mean localization error for ten realizations, for varying number of nodes in network, using error-free ranges.

Figure 5 Mean localization error for ten realizations, for varying number of nodes in network, using error-induced ranges.

Table 1 Node locations and range estimates

Node	Locations (in meters)		Node-to-node range (meters)				
	x-position	*y*-position	1	25	30	50	70
1	4,089	4,538	NR	3,578	9,879	3,114	8,117
25	9,877	9,874	3,402	NR	987	7,664	6,774
30	1,393	2,375	5,590	5,687	NR	6,677	9,198
50	7,077	5,550	2,265	2,278	7,311	NR	8,765
70	634	4,567	2,451	886	4,412	3,386	NR

NR, not relevant.

As the network size increases, the numbers of events detected gets reduced. This is due to the fact that as the number of nodes increases, the nodes in the network (or a sector) increases. This leads to transmitting a message packet from one level to another through a number of nodes, thereby depleting energy quickly. In this figure, we make a vivid comparison of our scheme with various techniques and methods. Our scheme proves out to overcome the defect of traditional methods and detects the events more efficiently. We proved that considering the range and location estimates at regular intervals gives a precise localization.

In Figure 8, we see the localization estimation error from different target distances. Our scheme and the scheme in [5] perform the same with little error difference, which is due to the fact that the BS or ref$_x$ is considered to be the centralized node. Hence, this figures concludes that the localization estimation error increases as the target distance increases.
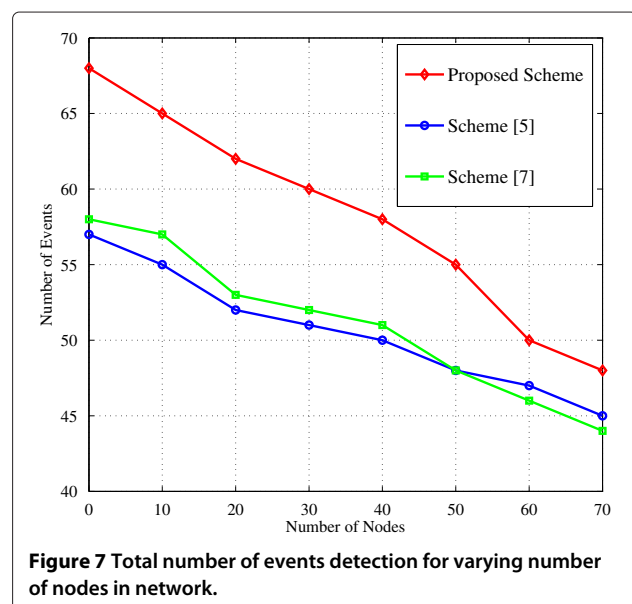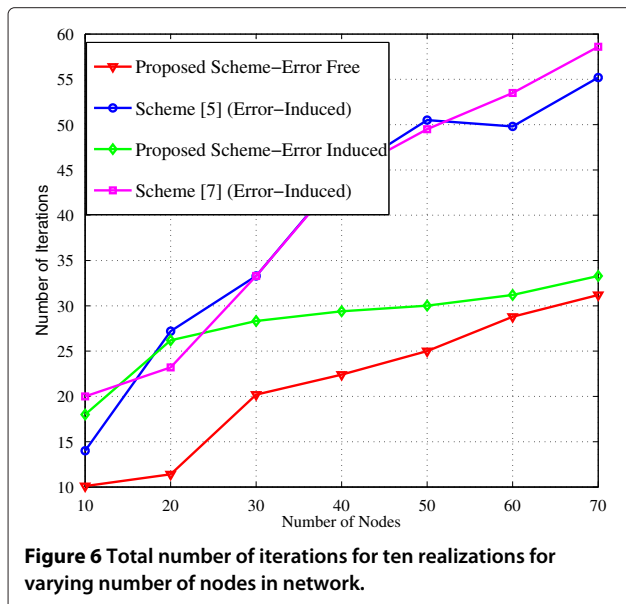
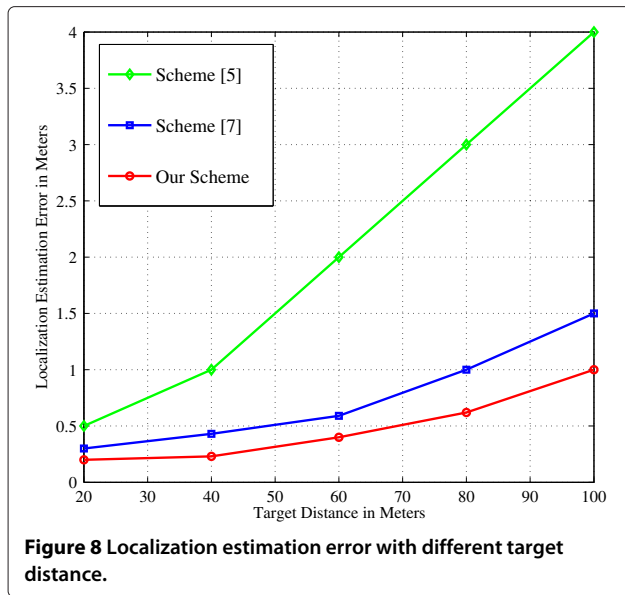Energy consumption becomes very crucial when it comes to underwater, as it becomes difficult to replace/charge the sensor nodes. Therefore, it is important to reduce energy consumption to increase the lifetime of the sensor nodes.

$$E = E_{\text{receiving}} + E_{\text{transmitting}} + k, \quad (5)$$

where $E_{\text{receiving}}$ is the energy dissipated in receiving the message packet and $E_{\text{transmitting}}$ is the energy spent in transmitting a message packet, and k is the energy spent in processing the information. In Figure 9, we make the energy consumption comparison of our proposed scheme with other schemes which proves that our algorithm requires very less energy consumption as compared to others algorithms. This proves that our scheme/algorithm can detect more number of events than any traditional schemes.

Figure 10 compares the lifetime of a network model that employs a combination of levelling on top of which sectoring is done. The results outperform the traditional levelling protocol. The techniques of levelling and flooding will overcome the problems of flooding and gossiping.

Figure 6 Total number of iterations for ten realizations for varying number of nodes in network.

Figure 7 Total number of events detection for varying number of nodes in network.

Figure 8 Localization estimation error with different target distance.


Figure 10 Levelling vs. sectoring.

In flooding, sensor broadcasts packet to all its neighbors until destination is reached; and in gossiping, sensor sends packets to a randomly selected neighbor which does the same. The sectoring on top of levelling will help the packet to reach the destination with much ease and in less number of hops, thereby consuming less energy.

5 Conclusions

In this paper, we have presented a simple acoustic underwater protocol and implement the same for the existing network. To employ the depth information of the sea level and to overcome the problems of watch circles, we employ Munk's formula by averaging the signal. When we applied the criterion of levelling,
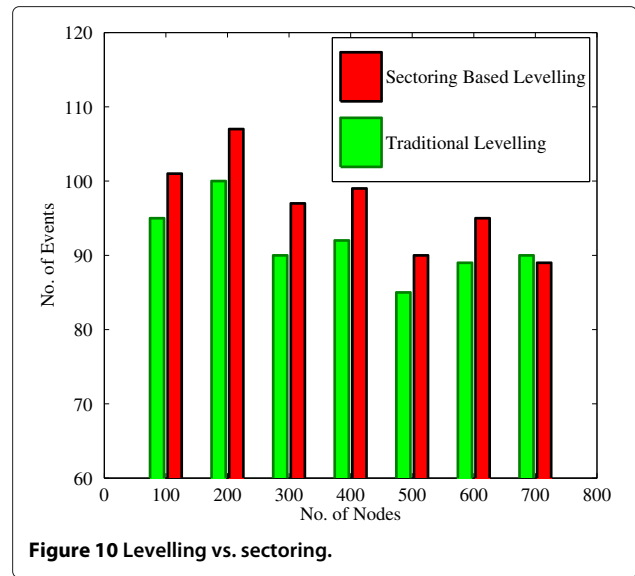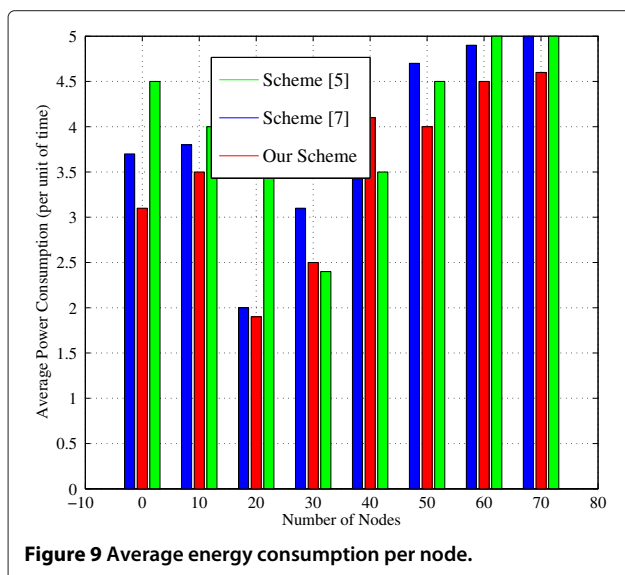
it is observed that the performance evaluation has been increased, and mesh networks typically for the proactive routes avoid the traffic control paths. The comparison of our proposed scheme with that of the traditional approaches proves that our scheme is energy-efficient and improves the localization capabilities. The localized scheme proves out to be robust and has less storage overhead and complexity. Future works include improving the algorithm by overcoming the error uncertainties caused by ambiguous solutions. We also plan to incorporate a partition strategy that reduces the total number of iterations and the accumulative errors.

Appendix

Synthetic data is generated in MATLAB using the built-in command 'unidrn' function that gives uniformly distributed coordinates of the nodes. Error free ranges, r_{true} are calculated by node pairs where as error induced ranges are generated by

$$r_{\text{error-induced}} = \mathcal{N}(\mu, \sigma) \times r_{\text{true}} + 0.02 \times r_{\text{true}} \times \chi^2(\theta) \quad (6)$$

where we multiply normally distributed random variable (rv) by error-free range and then adding 2% of the error-free range multiplied by Chi-squared distributed random variable. The normal distribution accounts for speed of sound errors which are proportional to range of the nodes. The mean, μ, and the variance, σ, of the normally distributed rv is set to 1 and 0.01, respectively, to ensure that the reported ranges are near actual range. Moreover, the Chi-squared distribution, $\chi^2(\theta)$, is used to account for the reflected transmission paths and depth variance errors which are also proportional to range of the nodes where θ is the degree of freedom and the mean and variance are set to 1 and 4, respectively. When Chi-squared

Figure 9 Average energy consumption per node.

distribution is multiplied with 10% of the error-free range, it results in 10% overestimation of the range. Watch circles are addressed by typically averaging out five to seven measurements over a frequency of 9–14 KHz for the course of network operation.

For analysis of the algorithm performance, we incorporate 70 nodes in a network field placed in 4×4-km region, since this is an approximate area of a testing field. For each simulation, the node-to-node range calculations are performed 15 times with different realizations of the range errors. For consistency across all simulation, parent node and ref_x are used for point of reference. During the simulation testing, the program rejects the ranges that fall outside the 5% confidence interval and estimates a good range between them. For analysis, we assumed a 70-node network with 2-m node range. We also performed ten realizations of the 70-node network, resulting in over 5,560 ranges being evaluated. The percentage error is thus calculated between actual ranges and ranges with the offsets. The mean percent range error is found to be 4.7%, and the largest percent error is 4.97%. In absolute terms, the accuracy of 70-node network is improved by 2-m range.

Competing interests
The authors declare no competing interests.

Authors' contributions
MAM has designed and developed a new scheme for GPS-free localization for underwater deployment of sensors. MAM has done an extensive literature survey along with MZS and MSA and come out with major concerns/issues which were not incorporated in the previous traditional algorithms. MZS and MSA did the drafting of the article by revising and by putting some interpretation data and some intellectual content. "All authors read and approved the final manuscript."

Author details
[1]Electrical and Electronics Engineering Department, Birla Institute of Technology and Science (BITS), Pilani, Rajasthan, 333031, India. [2]Electrical and Computer Engineering Department, Texas A&M University at Qatar, Doha, Qatar. [3]Computer, Electrical, and Mathematical Science and Engineering (CEMSE) Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Makkah Province, Kingdom of Saudi Arabia.

References
1. GN Roberts, R Sutton, *Advances in Unmanned Marine Vehicles*. (Institution of Electrical Engineers, Cornwall, 2006)
2. J Rice, in *Proceedings of the International Conference Underwater Acoustic Measurements: Technologies and Results*. Seaweb acoustic communication and navigation networks,Heraklion, Crete, (28 June to 1 July 2005)
3. K Scussel, Acoustic Modems for Underwater Communication (Wiley, Hoboken, 2003). [http://dx.doi.org/10.1002/0471219282.eot161]
4. D Kilfoyle, A Baggeroer, The state of the art in underwater acoustic telemetry Oceanic Eng. IEEE J. **25**, 4–27 (2000)
5. W Cheng, A Thaeler, X Cheng, F Liu, X Lu, Z Lu, in *29th IEEE International Conference on Distributed Computing Systems Workshops,2009 ICDCS Workshops '09*. Time-synchronization free localization in large scale underwater acoustic sensor networks (IEE Press, Piscataway, 2009), pp. 80–87
6. T Steinar, A Schei, finite-difference method for linearization in nonlinear estimation algorithms. Automatica. **33**(11), 2053–2058 ((1997)). http://www.sciencedirect.com/science/article/pii/S0005109897001271
7. D Grimmett, in *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*. Undersea communication network self-localization during the Unet 08 seatrial,Biloxi, (26 October to 29 October 2009), pp. 1–7
8. WH Munk, Sound channel in an exponentially stratified ocean, with application to. SOFAR. J. Acoust Soc. Am. **55**(2), 220–226 (1974). http://link.aip.org/link/?JAS/55/220/1
9. K Ausavapattanakun, A Nosratinia, Analysis of selective-repeat ARQ via matrix signal-flow graphs. Commun, IEEE Trans. **55**, 198–204 (2007)
10. J Yang, T Sarkar, Acceleration-invariance of hyperbolic frequency modulated pulse compression. Digital Signal Process. **18**(2), 228–235 (2008). http://www.sciencedirect.com/science/article/pii/S1051200407000218
11. J Rice, B Creber, C Fletcher, P Baxley, K Rogers, K McDonald, D Rees, M Wolf, S Merriam, R Mehio, J Proakis, K Scussel, D Porta, J Baker, J Hardiman, D Green, in *OCEANS 2000 MTS/IEEE Conference and Exhibition, Volume 3,(2000)*. Evolution of Seaweb underwater acoustic networking (IEEE Press, Piscataway,(2000)), pp. 2007–201
12. D Kilfoyle, A Baggeroer, The state of the art in underwater acoustic telemetry. Oceanic Eng. IEEE J. **25**, 4–27 (2000)
13. SP Ouimet, *Undersea navigation of a glider UUV using an acoustic communications network, M.S.* Naval Postgraduate School, Thesis, 2005)
14. TM Page, Proof of law of cosines. http://www.themathpage.com/aTrig/law-of-cosines.htm
15. A Mirza, R Garimella, in *International Conference on Information Networking, 2009. ICOIN 2009*. PASCAL: power aware sectoring based clustering algorithm for wireless sensor networks (IEEE, Piscataway, 2009, pp. 1–6
16. L Krause, A direct solution to GPS-type navigation equations. Aerosp. Electron. Syst. IEEE Trans. AES-23(2), 225–232 (1987)