Comparative node selection-based localization technique for wireless sensor networks: A bilateration approach

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Comparative Node Selection Based Localization Technique for Wireless Sensor Networks: A Bilateration Approach

Nikhath Tabassum[1] | Geetha D. Devanagavi[1] | Rajashekhar C. Biradar[1] | Mihai T. Lazarescu[2]

[1]School of Electronics and Communication Engineering, REVA University, Karnataka, India

[2]Department of Electronics and Communication, Politecnico di Torino, Torino, Italy

Correspondence
Nikhath Tabassum Email: nikhath005@gmail.com

Summary

Wireless sensor networks find extensive applications, such as environmental and smart city monitoring, structural health, and target location. To be useful, most sensor data must be localized. We propose a node localization technique based on bilateration comparison (BACL) for dense networks, which considers two reference nodes to determine the unknown position of a third node. The mirror positions resulted from bilateration are resolved by comparing their coordinates with the coordinates of the reference nodes. Additionally, we use network clustering to further refine the location of the nodes. We show that BACL has several advantages over Energy Aware Co-operative Localization (EACL) and Underwater Recursive Position Estimation (URPE): (1) BACL uses bilateration (needs only two reference nodes) instead of trilateration (that needs three reference nodes), (2) BACL needs reference (anchor) nodes only on the field periphery, and (3) BACL needs substantially less communication and computation. Through simulation we show that BACL localization accuracy, as root mean square error, improves by 53% that of URPE and by 40% that of EACL. We also explore the BACL localization error when the anchor nodes are placed on one or multiple sides of a rectangular field, as a trade-off between localization accuracy and network deployment effort. Best accuracy is achieved using anchors on all field sides, but we show that localization refinement using node clustering and anchor nodes only on one side of the field has comparable localization accuracy with anchor nodes on two sides but without clustering.

Keywords: Bilateration Localization, Trilateration, Wireless sensor network, Received signal strength indicator, Outdoor localization, Cluster formation

1 | INTRODUCTION

Wireless sensor networks (WSNs) connect many sensor nodes that are spatially distributed in the field of interest for various applications, such as environmental monitoring [1], target tracking [2], event detection [3], structural health monitoring [4] and solid waste management [5]. Many WSN applications rely on the location of the sensor nodes to make sense of the collected data. Global Positioning System (GPS) can be used for node localization during deployment [6], yet its accuracy can be limited by several factors, e.g., tree canopy, canyons, or tunnels and indoor environments. Moreover, GPS energy consumption is often

too high for the energy reserves on-board the WSN nodes to be used during WSN normal operation, and the relatively high cost of GPS devices can be another limiting factor.

Several other techniques can be used for WSN node localization that provide adequate accuracy with lower cost and energy consumption. The nodes whose locations are known beforehand are called "anchor nodes". The sensor nodes whose locations are unknown are referred to as "unknown sensor nodes" or just "sensor nodes" in this article. The unknown sensor nodes can be localized by stationary anchor nodes or moving anchor nodes. In former method, the positions of the unknown sensor nodes can be found by measuring the distances to the anchor nodes [7, 8, 9, 10, 11, 12] or by counting the number of nodes between the anchor nodes and the unknown sensor nodes [13, 14, 15, 16]. In the latter method, the anchor nodes move between the unknown sensor nodes along a fixed trajectory [17, 18, 19] for the localization process.

To localize an unknown node using measurements of distances to anchors, trilateration needs three non-collinear anchor nodes. The major concern is that the location of the unknown sensor node is not consistent, if the group of reference anchor nodes varies. So, finding a suitable group of anchor nodes and using them for localizing most network nodes may require significant battery energy from these reference nodes. In the proposed method, BACL, the anchor nodes are placed at an edge of the network. These anchor nodes become reference nodes to localize the unknown sensor nodes during bilateration process. Localization starts from the anchors on one edge of the network and progresses until it reaches the opposite edge of the network, in one direction only. The mirror nodes obtained during bilateration process are filtered by comparing their coordinates with those of the participating reference nodes. The newly localized nodes then become reference nodes for further localization of unknown nodes. This recursive process continues until all nodes are localized.

The major contributions of the proposed method are:

- The proposed method considers only two nodes for the localization process instead of three. The error variation in choosing three reference nodes is larger, compared to two reference nodes, as we discuss in Section 5.
- We eliminate the bilateration mirror nodes by incrementally localizing the nodes in the field and comparing the coordinates of the localized node with those of the reference nodes.
- Using bilateration for one-hop neighbors, we significantly reduce the number of beacon messages, hence the communication overhead.
- We make use of the underlying node clustering (where available, at protocol or application level) to reduce error accumulation and refine the localization of the whole field, thus requiring less anchor nodes.
- We localize all nodes in two passes, one for coarse localization and one for refinement, which can be faster and less resource demanding than recursive methods that need several localization iterations.
- We assess the performance of the proposed method by varying the topology of anchor nodes.

The rest of the article is organized as follows. Section 2 discusses most relevant related work. Section 3 introduces the localization problem. Section 4 presents the proposed localization method. Section 5 presents and discusses the experimental results. Section 6 concludes the article.

2 | RELATED WORKS

The localization methods can be broadly classified based on the position of the anchor nodes. The anchor nodes can be stationary or mobile.

The methods that have stationary anchor nodes, can be further classified into:

A. Range based methods, wherein the unknown sensor nodes calculate the Euclidean distance from anchor nodes.

B. Range free methods, estimate the distance by counting the number of nodes between the unknown sensor node and anchor nodes.

The Range free methods can be further classified based on:

- Received Signal Strength Indicator (RSSI): The distance to the anchor node is estimated by the signal strength received at the sensor node location.

- Time of Arrival (ToA): distance is determined by measuring the time taken by the signal to travel between the anchor node and the unknown sensor node
- Time Distance of Arrival (TDoA):uses the difference between the time of arrival of ultrasonic signals and radio frequency signals at the receiver.
- Angle of Arrival (AoA):determines the node position from the angle of arrival of the signal

In the first method in Range based algorithms is RSSI based localization technique, once a sensor node estimates the distance to the anchor nodes, it takes three anchor nodes to locate a sensor node by trilateration [7, 20]. However, bilateration method can be employed to locate the sensor nodes by having only two reference anchor nodes as in the proposed method. Some of the most relevant works to the proposed method are discussed in RSSI based methods. In the Underwater Recursive Position Estimation (URPE) method [20], the anchor nodes broadcast their positions to all neighboring sensor nodes. Once the neighboring nodes estimate their positions from these broadcasts, they become anchor nodes and broadcast their positions. But, as more nodes estimate their positions and start broadcasting, the broadcast messages increase the communication overhead. So, not all the nodes are allowed to become reference nodes. To curtail the reference nodes, a confidence value is calculated based on position error, and a node can become reference only if its confidence is above a threshold. A high threshold may prevent the localization of all sensor nodes, because there may be too few reference nodes, while a low threshold increases the position error in the network. So, there is a trade-off between accuracy, localization coverage, and communication overhead. If accuracy increases, the communication overhead also increases, which drains the batteries and reduces the localization coverage.

Another method, Energy Aware Co-operative Localization (EACL) [21] has a separate refinement phase for the poorly localized nodes, with high position error and low confidence. It also takes into account the energy spent by the sensor nodes to calculate the confidence threshold to uniform energy consumption among nodes. But nearby reference nodes connected to many other nodes may not be always available. Unknown sensor nodes may need to look farther for nodes for localization, which may increase localization error.

A variation of this is in the ERL strategy [22]. Instead of considering the energy of the nodes to compute the confidence value, it simply limits the reference nodes depending on the distance between them. The spacing between the reference nodes is maintained at $r/2$ and $r/3$, where r is the cell radius, so that at least one reference node is present per cell. Hence, there is a trade off between energy consumption, availability of nodes, and error propagation.

Another method which uses the shortest path between the anchor and unknown sensor nodes is RSPA [12]. It also uses trilateration to localize the sensor nodes, but takes the average of the shortest paths (minimum number of connecting nodes) to the anchor nodes to reduce the communication overhead. This method works best in a uniformly distributed network, but the accuracy decreases in irregular networks.

All above methods rely on a trade-off between the selection of the anchor (reference) nodes, communication overhead, and localization errors. Another problem these methods face is the accumulation of errors. When a node has high localization error, it propagates to other nodes and starts accumulating. To reduce this error, the reference nodes should be selected to have the least error. But again, node selection has factors to be considered, as listed above.

The second method is TOA. The OToA [23] is a centralized algorithm with asynchronous network, the packets from the nodes are time stamped and passed on to the centralized node, that calculates the relative distances. The technique is simple but the accuracy is not very high as this method is dependent on precise time synchronization. The third method of TDoA, utilizes the difference of time of arrival of ultrasonic and radio frequency signal at the receiver to localize the unknown sensor nodes. Once the initial estimates are obtained, they are refined using Semi Definite Programming (SDP) [24]. Though it does not depend on precise time synchronization, every node should have additional circuitry to generate and detect two different signals. The fourth method is Angle of arrival, WCSL [25] uses the angle of arrival information at 2 node array of sensors to estimate the distance between nodes, but this method requires a highly accurate directional antenna with a narrow beam.

The second type of algorithms use Range Free techniques. As in KRR ML [13], a ‘small hop count matrix’ among anchor nodes is determined in the offline phase and in online phase, the sensor node locations are mapped. This method is suitable for larger networks but the accumulative error is high. For a medium network, the HDV HOP [14], anchor nodes are present at the perimeter of the network. Through anchor node broadcast the neighboring sensor nodes know their locations. The range free algorithms can also be centralized. In RARL [15], there is a hierarchy. The first level consists of anchor nodes, the second level consists of multipoint relays(MPRS) and the next level onwards unknown sensor nodes are present. 3 anchors nodes which are one hop away are considered for calculating the location of sensor nodes. These anchor nodes can be one-hop neighbor or

TABLE 1 Description of notations

Notation	Description
N	Total number of sensor nodes in the network
(x_i, y_i)	x Coordinate and y coordinate of sensor node
V_i	Vertex identifier with the two attributes x_i and y_i
E_{ij}	Edge between two sensor nodes V_i and V_j
r_i	Radio range of sensor node with vertex V_i
T_p	token at level p, $p = 1 \ldots l$
ϵ_{rmse}	Root mean square error
δr_i	RSSI measurement error
ϵ	Coordinate measurement error
$M_{algorithm}$	Communication overhead of the algorithm
M_d	Measurement packet
M_{cal}	Resultant coordinate packet
M_{dtoken}	Token packet
M_t	Packet containing threshold value
M_{anch}	Packet from anchor node
p	General representation of a packet
$I_{algorithm}$	Computation overhead of the algorithm
I_x	Computation of x coordinate
I_y	Computation of y coordinate
I_{avg}	Average calculation
I_{th}	Threshold calculation
c	General representation unit of a computation

multi-hop neighbor. An improvement to this is the SDVH [16], wherein, the best of 3 anchors are selected to calculate the hop distance. The range free techniques though are simple but the accuracy is lesser than the range based methods.

The above methods have stationary anchor nodes. The next set of methods have mobile anchor nodes that move along a predetermined trajectory to map the unknown nodes. In LMADA [17], an anchor node moving along the network has a compass and 4 directional antennas. 2 directional antennas pointing horizontally and 2 pointing vertically. The positions of the unknown sensor nodes at the periphery has higher error. Instead of having one anchor node, there are 3 anchor nodes which form a triangle and move along the network as in GSCAN [18]. More the number of moving anchor nodes the better the accuracy. The coverage is better in IMAPP [19], as the anchor nodes move in hexagonal shape around the cluster of nodes which have highest connectivity. At each vertex of the hexagon, a beacon message is generated which helps in locating the unknown sensor nodes. In these methods the requirement of anchor nodes is less, but the corner nodes and the peripheral nodes have less accuracy.

3 | SYSTEM MODEL AND PRELIMINARIES

We discuss and compare several methods that can be used to determine the unknown position of a node (localize a node) using the known positions of other nodes and their distances to the node to be localized. Specifically, we discuss the *unilateration*, *bilateration*, and *trilateration* methods.

We assume that every node is equipped with an omnidirectional antenna and the radio frequency propagation in the environment is isotropic. We define *one-hop neighbors* as those nodes that are in radio range of each other and can communicate directly. Moreover, we assume that each node can measure the intensity of the radio communication (RSSI) it receives from its neighboring nodes, and that there is a direct relationship between the RSSI and the Euclidean distance between the communicating nodes. The main notations that we use in the rest of the article are summarized in Table 1.

In unilateration, we consider an unknown node V with coordinates (x, y), which is one-hop neighbor of a reference node, with coordinates (x_1, y_1) and radio range r as in Fig. 1.

This unilateration solution is given by

$$(x - x_1)^2 + (y - y_1)^2 = r^2, \tag{1}$$

and can be anywhere on the circumference of the circle with radius r, as shown in Fig. 1. Hence, unilateration cannot uniquely define the position of a node and is poorly suited for localization.

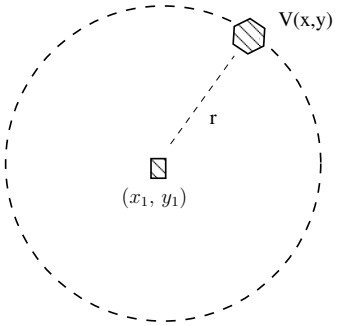


FIGURE 1 Unilateration of node V position. (x_1, y_1) are the coordinates of the known position (anchor) node and r is the measured distance between nodes

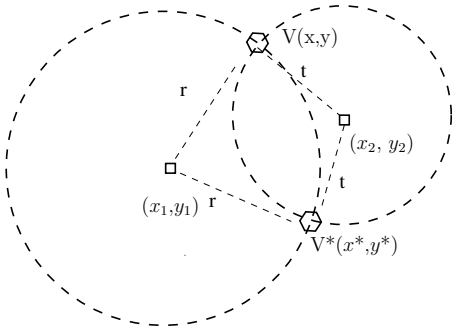


FIGURE 2 Bilateration of node V position, with (x_1, y_1) and (x_2, y_2) the coordinates of the known position (anchor) nodes, and r and t are the respectively measured radii and V^* is the equally likely mirror node position

Bilateration, shown in Fig. 2, refines the unknown position of the sensor node V using an additional reference node, with coordinates (x_2, y_2), measuring a distance t from V.

The possible positions of the node to be localized, V and V^*, are obtained by solving

$$(x - x_1)^2 + (y - y_1)^2 = r^2, \tag{2}$$

$$(x - x_2)^2 + (y - y_2)^2 = t^2, \tag{3}$$

where V^* is the mirror location (flip ambiguity).

V^* can be eliminated by comparing the reference node coordinates, as mentioned in Section 4, or by trilateration (shown in Fig. 3), which uses a third non-collinear reference node. If we assume that it has coordinates (e, f) and measures a distance h to node V, it is described by

$$(x - x_1)^2 + (y - y_1)^2 = r^2, \tag{4}$$

$$(x - x_2)^2 + (y - y_2)^2 = t^2, \tag{5}$$

$$(x - x_3)^2 + (y - y_3)^2 = h^2. \tag{6}$$

Subrata et al. [26] solved (4), (5), and (6) using the notations

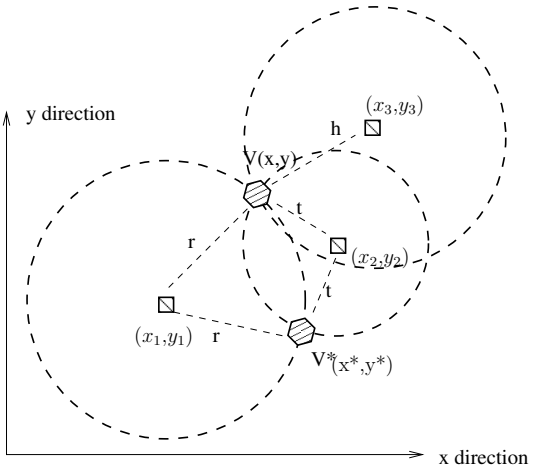


FIGURE 3 T

rilateration of node V position. (x_1, y_1), (x_2, y_2), and $(x_3, y_3)$are the coordinates of the known position (anchor) nodes, and r, t, and h are the respectively measured radii. V^* is the position of one mirror, discarded by the algorithm

$$x_1^2 + y_1^2 - r^2 = g^2, \tag{7}$$

$$x_2^2 + y_2^2 - t^2 = k^2, \tag{8}$$

$$x_3^2 + y_3^2 - h^2 = j^2, \tag{9}$$

and solving to obtain the coordinates (x, y) of node V

$$x = \frac{\left(g^2 - j^2\right)\left(y_2 - y_1\right) - \left(g^2 - k^2\right)\left(y_3 - y_1\right)}{2\left[\left(x_2 - x_1\right)\left(y_3 - y_1\right) - \left(x_3 - x_1\right)\left(y_2 - y_1\right)\right]}, \tag{10}$$

$$y = \frac{\left(g^2 - j^2\right)\left(x_2 - x_1\right) - \left(g^2 - k^2\right)\left(x_3 - x_1\right)}{2\left[\left(y_2 - y_1\right)\left(x_3 - x_1\right) - \left(x_2 - x_1\right)\left(y_3 - y_1\right)\right]}. \tag{11}$$

If the coordinates of the reference nodes are afflicted by errors, then the location of the unknown node will be also afflicted by errors [27], which would result in an accumulation of localization errors across the field. Moreover, there can be flip ambiguity when the mirror coordinates may be selected instead of the real coordinates [28]. The flip ambiguity is overcome in the proposed method as described in Section 4.

Due to the ease of selection of the participating nodes and the reduced complexity of the bilateration process, we have chosen bilateration process over trilateration.

4 | PROPOSED BILATERATION AND COMPARISON LOCALIZATION (BACL) METHOD

We consider a network of N sensor nodes, represented by a graph $G = (V, E)$. The set of vertices V represents the location of the sensor nodes in the network, as shown in Fig. 4. A sensor node with vertex $V_i \in V$, $i = 1, 2, \ldots, N$, has coordinates (x_i, y_i). E represents the set of edges between the sensor nodes. An edge E_{ij} exists between sensor node V_{i} and V_{j} if they are within radio range of each other, hence one-hop neighbors.

The proposed algorithm can be divided in two steps: (1) localization by bilateration process and (2) refinement by cluster formation. In Fig. 4, we can see how the localization by bilateration starts using the anchor nodes present at the edge of the network. We have anchor nodes V_1, V_2, and V_3 lined up along the x axis at known locations. We use them in pairs as anchor nodes to localize an unknown node that they can detect within one-hop radio range. This way, they localize nodes V_4, V_5, and V_6, but for each we obtain also mirror nodes, V_4^*, V_5^*, and V_6^* (see Fig. 4). We filter the mirror nodes as follows. When reference nodes

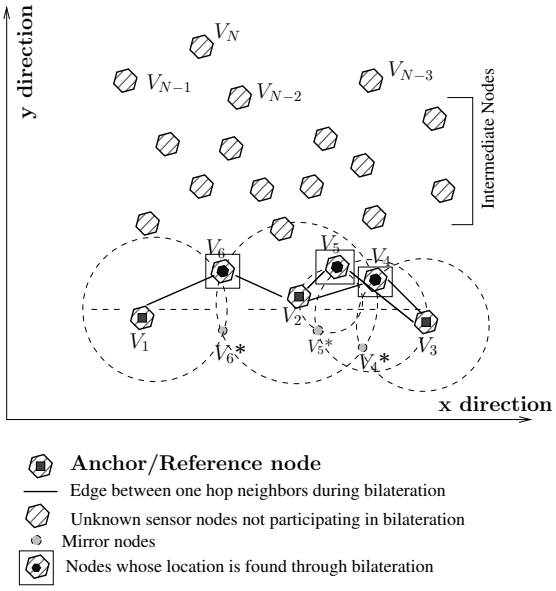


FIGURE 4 Network model of BACL with anchor nodes at the edge

V_1 and V_2 participate in bilateration to localize V_6, we select the true location between V_6 and V_6^* as the one whose y coordinate is greater than the y coordinates of the reference nodes (which, at the beginning, are at the edge of the field). Similarly, reference nodes V_2 and V_3 will localize V_4 and V_5, respectively.

We use tokens to differentiate the reference nodes from the nodes to be localized. Initially, the anchor nodes at the edge of the network generate their own tokens with a value of zero. When the anchor nodes participate in bilateration, they pass an incremented token to the newly localized node. Hence, the first layer of nodes that are localized directly by the anchors (located at the edge of the network) will have tokens with values equal to one. Similarly, when the nodes on this layer start localizing by bilateration, they will pass to the localized nodes their token incremented by one (hence with a value of two). Since only the nodes that have a token can participate in localizations, we enforce this way a layered localization progression, from anchor nodes towards the rest of the field.

We can see this process unfolding in Fig. 5. Nodes V_1, V_4, V_5, and V_6 become the reference nodes to localize V_7, V_8, and V_9, discarding mirror nodes based on their y coordinates. V_7, V_8, and V_9 will receive a token with value of two. Note that V_1, an anchor, passes to V_9 a token with value one, but V_6 passes a token with value two. V_9 selects the highest token value, two. V_8 is one hop neighbor of V_5, V_2 and V_6. It chooses V_6 and V_5 for bilateration because the token values of V_5 and V_6 are higher than V_2. The process continues along the Y axis until all nodes in the field are localized and have a token.

4.1 | Localization by Bilateration

To show the operation of the bilateration process, we consider two reference nodes, V_i with coordinates (x_i, y_i) and V_j with coordinates (x_j, y_j), and a node of unknown position, V_k, with coordinates (x_k, y_k) to be determined. We assume that V_k is within one-hop radio range to reference nodes V_i and V_j, which sense V_k at radio ranges r_i and r_j, respectively (see Fig. 6). The coordinates of the unknown node V_k are obtained from the intersection of V_i and V_j nodes

$$(x_k - x_i)^2 + (y_k - y_i)^2 = {r_i}^2, \tag{12}$$

$$(x_k - x_j)^2 + (y_k - y_j)^2 = {r_j}^2. \tag{13}$$

Subtracting (13) from (12) we obtain

$$(x_k - x_i)^2 + (y_k - y_i)^2 - {r_i}^2 - [(x_k - x_j)^2 + (y_k - y_j)^2 - {r_j}^2] = 0, \tag{14}$$

$$(x_k - x_i)^2 + (y_k - y_i)^2 - r_i^2 - (x_k - x_j)^2 - (y_k - y_j)^2 + r_j^2 = 0. \tag{15}$$

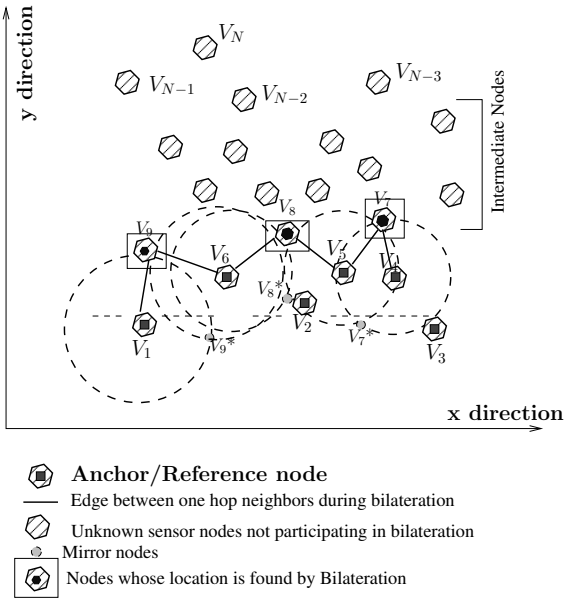


FIGURE 5 Network model of BACL with reference nodes

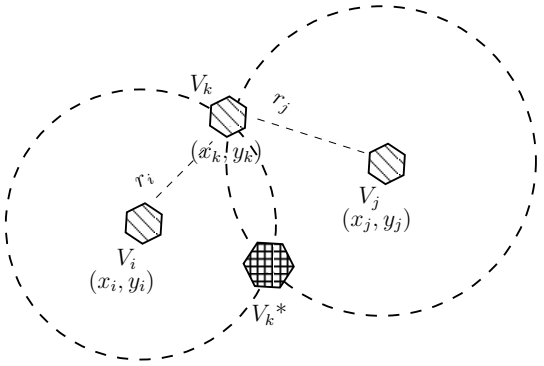


FIGURE 6 Bilateration process to find node V_k

Grouping the x_k and y_k terms together we obtain

$$2x_k(x_j - x_i) + 2y_k(y_j - y_i) + {r_j}^2 - {r_i}^2 + x_i^2 - x_j^2 + y_i^2 - y_j^2 = 0, \tag{16}$$

$$x_k = \frac{{x_j}^2 - {x_i}^2 + {y_j}^2 - {y_i}^2 + {r_i}^2 - {r_j}^2}{2(x_j - x_i)} + y_k \frac{y_i - y_j}{x_j - x_i}. \tag{17}$$

With

$$d_{\text{ij}} = \frac{y_i - y_j}{x_j - x_i} \tag{18}$$

and

$$e_{\text{ij}} = \frac{{x_j}^2 - {x_i}^2 + {y_j}^2 - {y_i}^2 + {r_i}^2 - {r_j}^2}{2(x_j - x_i)}, \tag{19}$$

we get

$$x_k = y_k d_{\text{ij}} + e_{\text{ij}}. \tag{20}$$

Substituting (20) in (12), we obtain

$$(y_k - y_i)^2 + (y_k d_{\text{ij}} + e_{\text{ij}} - x_i)^2 = {r_i}^2. \tag{21}$$

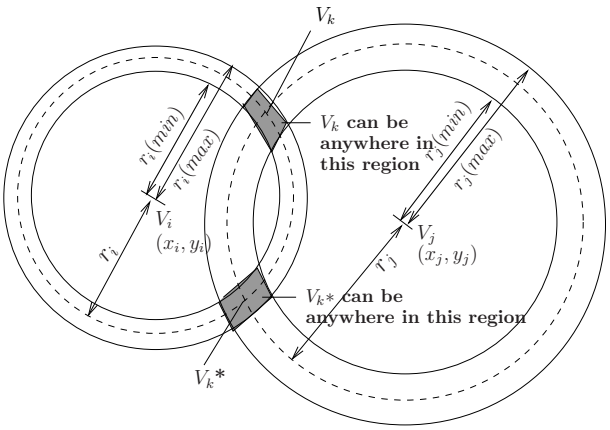


FIGURE 7 Error due to RSSI measurement uncertainties

We replace $(e_{ij} - x_i)$ with c_{ij}

$$(y_k - y_i)^2 + (y_k d_{ij} + c_{ij})^2 = {r_i}^2, \tag{22}$$

$$y_k^2 - 2y_k y_i + y_i^2 + y_k^2 d_{ij}^2 + 2y_k d_{ij} c_{ij} + c_{ij}^2 - r_i^2 = 0, \tag{23}$$

$$y_k^2(1 + d_{ij}^2) + y_k(2d_{ij}c_{ij} - 2y_i) + (c_{ij}^2 - r_i^2 + y_i^2) = 0. \tag{24}$$

To solve for y_k, let

$$(1 + d_{ij}^2) = p^2 \tag{25}$$

and

$$y_k = \frac{-(d_{ij}c_{ij} - y_i) \pm \sqrt{(d_{ij}c_{ij} - y_i)^2 - (p^2)({y_i}^2 + {c_{ij}}^2 - {r_i}^2)}}{p^2}. \tag{26}$$

From (26), we observe that there are two values of y_k with different magnitudes. Among these two coordinates, one is the real node and the other is a mirror node. To eliminate the mirror node, the y coordinate of V_k is compared with the y coordinates of V_i and V_j, i.e., y_k is compared with y_i and y_j. The position with the largest y_k compared to y_i and y_j is selected for the node coordinate, as shown in Fig. 6. The other value is the mirror node, which is discarded. We use a token to ensure that the localization happens only in the positive y direction as the algorithm proceeds. The token T_p is passed by the reference nodes to the newly mapped sensor node. The sensor node that has acquired this token can proceed to localize its one-hop neighbors and act as reference node. By means of the token, the localization progresses monotonically along the Y axis.

For a relatively dense network, we can assume that all nodes are connected. New nodes entering the network will be surrounded by nodes already localized. In this case, any two neighboring nodes that are one hop away with the highest token numbers are chosen as reference nodes for the bilateration process.

4.2 | Refinement Phase of Proposed Method (BACL)

The localization error propagates through the network, as the localization errors of the reference nodes compound with the localizations of new nodes. BACL refinement phase aims to reduce the accumulated error. The error is due to RSSI measurement uncertainties, as shown in Fig. 7. The measured radii r_i and r_j may vary between $[r_i(\text{min}), r_i(\text{max})]$ and $[r_j(\text{min}), r_j(\text{max})]$, respectively. The actual position of V_k may differ and lie in the region of the intersection between $[r_i(\text{min}), r_i(\text{max})]$ and $[r_j(\text{min}), r_j(\text{max})]$. Due to this there can be variation in the position calculated using bilateration.

This error can be minimized by applying multiple bilaterations and averaging the results. To have multiple bilaterations for a single node, we use clusters and use the positions of the cluster member nodes to refine the position of the cluster head, which was initially obtained by BACL (see Section 4.1). We use a clustering method based on the network topology [29, 30] which uses the degree of connectivity introduced in the HEED algorithm [31].

Step 1: *The cluster is formed and a cluster head is elected.* We assume that the sensor nodes have preassigned unique identification numbers (IDs) and we denote them by $V_p^{(n)}$, with $n = 1, 2 \ldots N$. Consider the network shown in Fig. 8, wherein

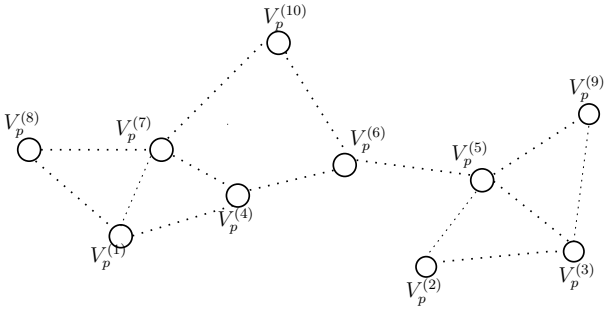


FIGURE 8 Nodes in a network with their one-hop neighbors

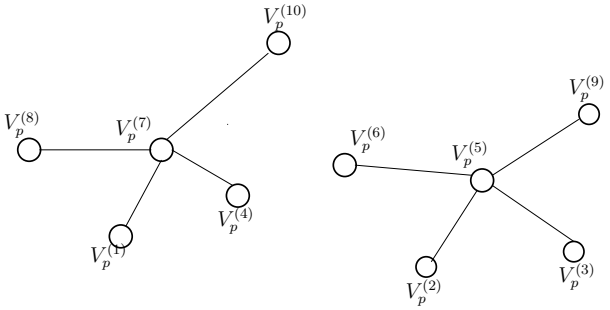


FIGURE 9 Cluster formation with one-hop neighbors

each node broadcasts its ID and also receives the IDs of the nodes that are one hop away. By receiving the IDs from its neighbors, each node can calculate the number of the one-hop neighbors it is connected to. The one-hop neighbor count is again broadcast and then the node which has got the highest number of nodes connected to it is elected cluster head and provides its ID to its one-hop neighbors. In Fig. 9, $V_p^{(7)}$ and $V_p^{(5)}$ become cluster heads because they have the highest node connectivity. Every node chooses the neighbor with the highest one-hop connectivity as cluster head. If two or more neighbors are cluster head candidates with the same connectivity, the node with the highest ID becomes cluster head.

Step 2: *The participating nodes form pairs, to refine the cluster head location.* Cluster members contact their one-hop neighbors that belong to the same cluster (identified by the unique ID of the cluster head) and localize the cluster head again through bilateration using (17) and (26).

Step 3: *The refined cluster head location is used to refine the locations of the participating nodes.* The cluster members refine their locations from the newly calculated cluster head from (12) and (13), where (x_i, y_i) denotes the older cluster head coordinates before refinement, and (x_j, y_j) denotes the newly calculated coordinates of V_c after refinement.

To show the improvement in localization error, consider the arrangement in Fig. 10, wherein the cluster head is denoted as V_c and the cluster members are $V_p^{(1)}, V_p^{(2)}, V_p^{(3)}$ and $V_p^{(4)}$. Although they are equally separated from the cluster head with radii r_i, they have RSSI measurement errors denoted by $\delta r_1, \delta r_2, \delta r_3$ and δr_4. We consider the cluster members in pairs to refine the location of V_c, i.e., $V_p^{(1)}$ and $V_p^{(4)}$, $V_p^{(4)}$ and $V_p^{(3)}$, $V_p^{(3)}$ and $V_p^{(2)}$, and $V_p^{(1)}$ and $V_p^{(2)}$. Initially, the pair $V_p^{(1)}$ and $V_p^{(4)}$ with coordinates (x_i, y_i) and $(-x_i, y_i)$ respectively participates in Bilateration. These coordinates are substituted in (2) and (3)

$$(x - x_i)^2 + (y - y_i)^2 = (r_i + \delta r_1)^2, \tag{27}$$

$$(x + x_i)^2 + (y - y_i)^2 = (r_i + \delta r_4)^2. \tag{28}$$

Since the error δr_1 is an independent variable and the correlation between r_i and δr_1 is zero, we consider

$$(r_i + \delta r_1)^2 = r_i^2 + (\delta r_1)^2. \tag{29}$$

Similarly,

$$(r_i + \delta r_4)^2 = r_i^2 + (\delta r_4)^2. \tag{30}$$

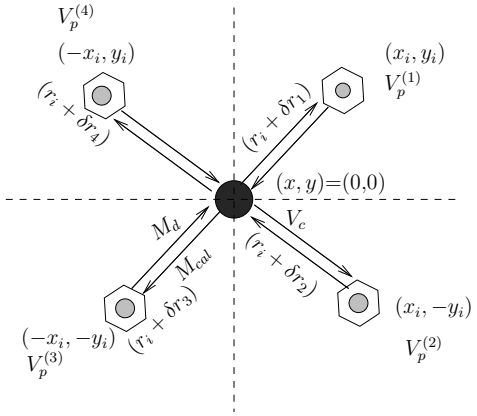


FIGURE 10 Refining cluster head location V_c by cluster members with RSSI measurement errors

Subtracting (28) from (27) and solving for x, we get

$$-4x_i x = (\delta r1)^2 - (\delta r_4)^2. \tag{31}$$

From (31), we get the value of x coordinate and denote it as $x_{1,4}$

$$x_{1,4} = \frac{(\delta r1)^2 - (\delta r_4)^2}{-4x_i}. \tag{32}$$

Consider the pair $V_p^{(2)}$ and $V_p^{(3)}$ to get cluster head coordinates by Bilateration. The coordinates of the participating nodes, $(x_i, -y_i)$ and $(-x_i, -y_i)$ are substituted in (2) and (3)

$$(x - x_i)^2 + (y + y_i)^2 = (r_i + \delta r_2)^2, \tag{33}$$

$$(x + x_i)^2 + (y + y_i)^2 = (r_i + \delta r_3)^2. \tag{34}$$

Subtracting (34) from (33) we get x coordinate and denote it by $x_{2,3}$

$$x_{2,3} = \frac{(\delta r_2)^2 - (\delta r_3)^2}{-4x_i}. \tag{35}$$

Similarly, for the node pair $V_p^{(1)}$ and $V_p^{(2)}$ with coordinates (x_i, y_i) and $(x_i, -y_i)$ respectively, we get

$$(x - x_i)^2 + (y - y_i)^2 = (r_i + \delta r_1)^2, \tag{36}$$

$$(x - x_i)^2 + (y + y_i)^2 = (r_i + \delta r_2)^2. \tag{37}$$

Solving (36) and (37), we get (38)

$$y_{1,2} = \frac{(\delta r_1)^2 - (\delta r_2)^2}{-4y_i}. \tag{38}$$

For the pair $V_p^{(4)}$ and $V_p^{(3)}$ with coordinates $(-x_i, y_i)$ and $(-x_i, -y_i)$, we get y coordinate of V_c as in (39)

$$y_{4,3} = \frac{(\delta r_4)^2 - (\delta r_3)^2}{-4y_i}. \tag{39}$$

Then we calculate the x coordinate of cluster head by taking average of $x_{1,4}$ and $x_{2,3}$ as in (40)

$$x = \frac{x_{1,4} + x_{2,3}}{2}, \tag{40}$$

$$x = \frac{x_{1,4}}{2} + \frac{x_{2,3}}{2}. \tag{41}$$

The y coordinate of the cluster head is obtained by the average of $y_{1,2}$ and $y_{4,3}$

$$y = \frac{y_{1,2} + y_{4,3}}{2}, \tag{42}$$

$$y = \frac{y_{1,2}}{2} + \frac{y_{4,3}}{2}. \tag{43}$$

From (41) and (43), it can be seen that by taking average, the cluster head coordinates are divided by a factor of 2. Therefore the cluster head position error will reduce as the number of participating nodes in the cluster increase. Since the x and y coordinates considered in the example are zero, (40) and (42) will go to zero if the numerator is zero, i.e., if the RSSI measurement error is zero. Let $x_{1,4}$ and $x_{3,2}$ be represented as ϵ_1 and ϵ_2, respectively. For 2 participating pairs to refine the location, we get

$$\epsilon = \frac{\epsilon_1 + \epsilon_2}{2}. \tag{44}$$

For n participating pairs in refinement of cluster head

$$\epsilon = \frac{\sum_{p=1}^{n} \epsilon_p}{n}, \tag{45}$$

and taking limit on both sides

$$\lim_{n\to\infty} \epsilon = \lim_{n\to\infty} \frac{\sum_{p=1}^{n} \epsilon_p}{n} = 0. \tag{46}$$

From (46) it is seen that the error is bounded and convergent. Hence, clustering and averaging the coordinate values reduces the RSSI measurement error and improves the location accuracy.

5 | EXPERIMENTAL RESULTS

The following parameters are used to assess the performance of the proposed BACL method of bilateration with the trilateration methods of EACL and URPE:

- *Computation overhead*: the calculations done by a sensor node for its localization in a network.
- *Communication overhead*: the messages exchanged for the localization of all network nodes.
- *Root Mean Square Error (RMSE)*: the error between the calculated node positions (x_j, y_j) and the actual positions (x_i, y_i)

$$\epsilon_{\text{rmse}} = \sqrt{\frac{1}{N}\sum_{\text{i=1}}^{N}\left[\left(x_i - x_j\right)^2 - \left(y_i - y_j\right)^2\right]}. \tag{47}$$

We report also the minimum, maximum, average, and standard deviation of localization errors.

5.1 | Localization Accuracy

The proposed method BACL was implemented in MATLAB for a field with 100 nodes. The nodes are randomly placed in an area of 120 m×120 m, as shown in Fig. 11. The reference nodes for BACL are ten edge nodes enclosed in rectangles in Fig. 11. For EACL and URPE, the anchor nodes are placed randomly in the network as indicated by nodes enclosed in circles. In BACL, the nodes in the vicinity of any two anchor nodes are localized through bilateration. In EACL and URPE, the nodes that get beacon messages from 3 anchor nodes, localize using trilateration. Trilateration reference nodes must not be collinear, while bilateration does not have restrictions for the relative position of the reference nodes.

The simulation is done by adding 10% of the real distance as random errors. The statistical results are reported by performing 50 independent simulations. The simulation starts from the edge of the network and proceeds to the opposite edge in the positive Y direction. The simulation errors are shown in Fig. 12. Then we simulate 20% and 30% random errors of the real distances, and the results are shown in Fig. 13 and Fig. 14, respectively. From Fig. 12, Fig. 13, and Fig. 14, we can see that for all algorithms the localization error increases with the distance error. On the other hand, it is small for less than 20 nodes in the network, but they are very likely to be either in direct range or at very few hops from the anchor nodes.

As the network grows, the localization errors diverge for EACL, URPE, and BACL because the accumulated errors increase. At some point, in EACL and URPE fewer nodes can become anchors (reference nodes) because they may have excessive localization errors. Hence, the networks may have too few anchors for good accuracy localization as they grow. In BACL, all nodes act as reference nodes as soon as they are localized. Moreover, we refine the localization (reduce the accumulated error) using the clustering technique at the cost of higher communication and computation overhead. We have compared the localization error for EACL, URPE and the proposed method BACL with and without the clustering method in Table 2, Table 3, and Table 4. It

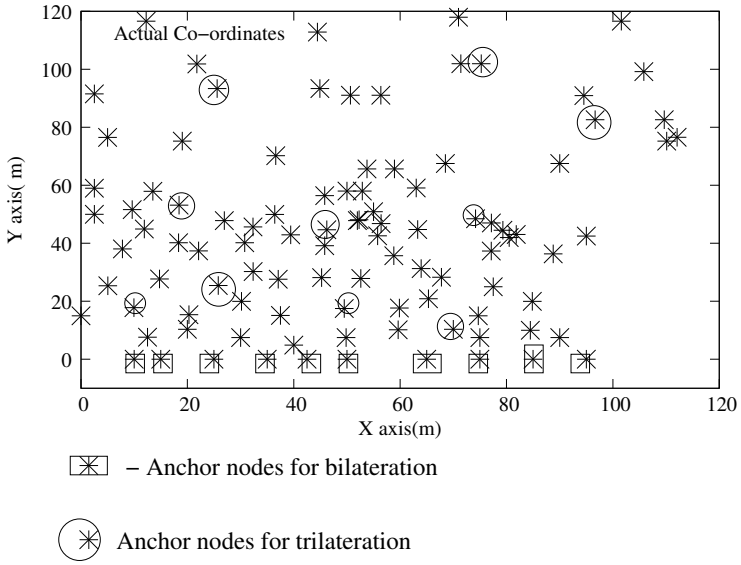


FIGURE 11 Placement of the 100 nodes in the network

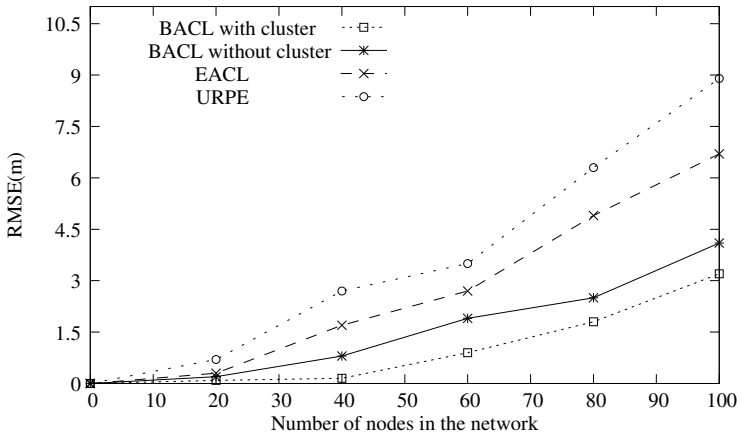


FIGURE 12 Root mean square error (RMSE) for BACL, EACL, and URPE for 10% distance measurement error

can be observed that the proposed method BACL has the least error compared to EACL and URPE when clustering is employed to refine the node location.

BACL performance for different anchor distributions in the network is shown in Table 5 and Fig. 15. We see that distributing the anchors on more sides of the network decreases the localization error. Also, the localization error for the network with anchors on all four edges is similar to that with anchors placed on only one edge but that uses clustering refinement.

We explore the scalability of the localization algorithm, with and without refinement, by increasing the number of nodes in the network from 100 to 200. We test the localization accuracy in the worst case, when the anchor nodes are placed only on one edge of the field, and report the results in Table 6. The localization error increases with the number of nodes, but localization refinement using the clustering technique generally reduces the error.

5.2 | Communication Energy

To calculate the communication energy for localization, we consider the setting in Fig. 10, with a node V_c and four neighboring reference nodes $V_p^{(1)}$, $V_p^{(2)}$, $V_p^{(3)}$, and $V_p^{(4)}$. Network nodes can have different number of neighbors, but we use four neighbors for our calculations to have a uniform reference for comparison between different methods. We denote a single communication packet by p. With M_d we denote an RSSI measurement packet, and with M_{cal} a packet containing the calculated coordinates. To

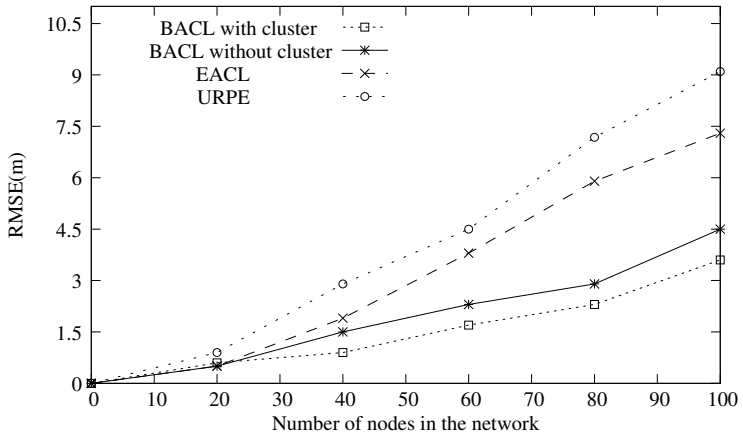


FIGURE 13 Root mean square error (RMSE) for BACL, EACL, and URPE for 20% distance measurement error

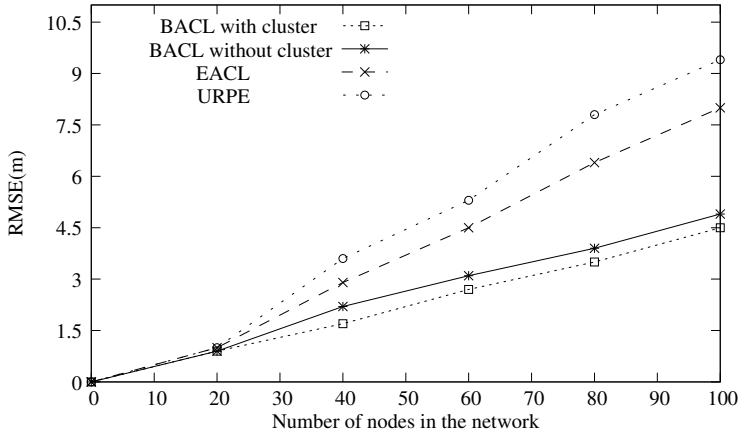


FIGURE 14 Root mean square error (RMSE) for BACL, EACL, and URPE for 30% distance measurement error

TABLE 2 Comparison of average, maximum, and standard deviation (σ) of localization error for different localization algorithms for 10% distance measurement error

Algorithm	Localization error (m)			
	min	average	max	σ
EACL	1.40	3.4	8.30	2.45
URPE	2.92	6.1	9.20	2.70
BACL without refinement	1.12	1.4	8.02	2.10
BACL with refinement	0.80	0.9	7.40	1.91

simplify the calculations, we consider that both transmitting and receiving a packet related to localization consumes the same energy.

For BACL, let $V_p^{(2)}$ and $V_p^{(3)}$ be the two reference nodes participating in a bilateration process to localize the node V_c. Since only the nodes with token can broadcast, V_c receives one M_d packet from $V_p^{(2)}$ and $V_p^{(3)}$, respectively. Once V_c calculates its position, it broadcasts it using an M_{cal} packet. V_c receives the token number with an M_{dtoken} packet. Hence

$$M_{\text{BACL}} = (2M_d + M_{\text{cal}}) + M_{\text{dtoken}}, \tag{48}$$

TABLE 3 Comparison of average, maximum, and standard deviation (σ) of localization error for different localization algorithms for 20% distance measurement error

Algorithm	Localization error (m)			
	min	average	max	σ
EACL	1.50	3.8	8.50	2.60
URPE	3.10	6.6	9.78	2.86
BACL without refinement	1.42	1.7	8.74	2.40
BACL with refinement	0.98	1.2	7.80	2.10

TABLE 4 Comparison of average, maximum, and standard deviation (σ) of localization error for different localization algorithms for 30% distance measurement error

Algorithm	Localization error (m)			
	min	average	max	σ
EACL	2.20	4.3	8.9	2.80
URPE	3.50	6.9	9.9	2.91
BACL without refinement	1.80	2.0	8.8	2.64
BACL with refinement	1.48	1.5	7.9	2.43

TABLE 5 Comparison of average, maximum and standard deviation (σ) of localization error for anchor placement at 1, 2 and 4 edges of the network for 10% distance measurement error

Algorithm	Localization error (m)			
	min	average	max	σ
BACL anchors on 1 side, without refinement	1.12	1.4	8.02	2.10
BACL anchors on 2 sides, without refinement	0.93	1.1	7.80	2.00
BACL anchors on 4 sides, without refinement	0.70	0.9	6.91	1.85
BACL anchors on 1 side, with refinement	0.80	0.9	7.40	1.91

or

$$M_{\text{BACL}} = (2p + p) + p, \tag{49}$$

$$M_{\text{BACL}} = 4p \text{ per node, without refinement.} \tag{50}$$

For an N-node network, the number of packets without refinement is

$$M_{\text{BACL}} = 4N. \tag{51}$$

For the refinement phase, since we have considered four neighbors, the node receives four M_d messages from the one-hop neighbors that indicate their neighbor count, and one M_d from the node with the highest neighbor count that announces itself as cluster head. The four reference nodes choose their cluster head using M_d packets. Since four node pairs participate in bilateration to refine the V_c location, the packets are $(2M_d + M_{\text{cal}})$ and one M_d packet to pass the token. The total packets per node for BACL, M_{BACL} is

$$M_{\text{BACL}} = (2M_d + M_{\text{cal}})4 + 5M_d + 4M_d + M_{\text{dtoken}}, \tag{52}$$

$$M_{\text{BACL}} = (2p + p)4 + 5p + 4p + p, \tag{53}$$

$$M_{\text{BACL}} = 22p \text{ per node.} \tag{54}$$

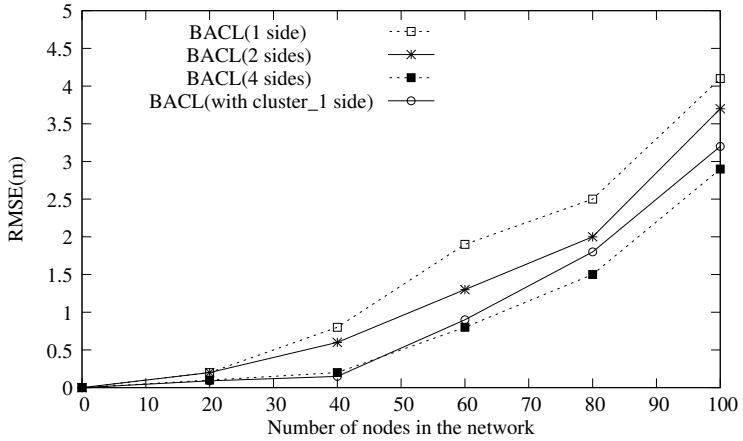


FIGURE 15 Comparison of the performance of BACL with anchors placed on 1, 2, and 4 edges of the network for 10% distance measurement error

TABLE 6 Comparison of average, maximum, and standard deviation (σ) of localization error for different sizes of network with 10% distance measurement error

Algorithm	Localization error (m)			
	min	average	max	σ
	100 nodes on the field			
BACL without cluster	1.1	1.4	8.0	2.1
BACL with cluster	0.8	0.9	7.4	1.9
	150 nodes on the field			
BACL without cluster	1.7	2.1	10.8	2.6
BACL with cluster	1.3	1.7	8.0	2.1
	200 nodes on the field			
BACL without cluster	2.8	3.5	11.9	2.9
BACL with cluster	1.9	2.2	9.8	2.4

For an N-node network, the number of packets with location refinement using clustering is

$$M_{\text{BACL}} = 22N. \tag{55}$$

For URPE, all four reference nodes broadcast their location. So, V_c receives all four packets, but chooses only three reference nodes for trilateration. Then it broadcasts its calculated position using one M_{cal} packet. Let M_{anch} be the packets received from the broadcasts of anchor nodes and M_t be the packet containing the threshold value that is passed on to V_c. The total packets for URPE, M_{URPE} becomes

$$M_{\text{URPE}} = 4M_d + M_{\text{cal}} + M_t + 10M_{\text{anch}}, \tag{56}$$

$$M_{\text{URPE}} = 4p + p + p + 10p, \tag{57}$$

$$M_{\text{URPE}} = 16p \text{ per node.} \tag{58}$$

$$\tag{59}$$

For an N-node network

$$M_{\text{URPE}} = 16N \text{ packets.} \tag{60}$$

TABLE 7 Communication overhead for localization methods for a network with N nodes

Method	**Overhead** (packets)
EACL	$31N$
URPE	$16N$
BACL	$4N$ without refinement
BACL	$22N$ with refinement

In EACL, a minimum of three rounds of broadcast from the anchor nodes is received by a node, from which it chooses the best three to locate the node

$$M_{\text{EACL}} = (10M_{\text{anch}})30 + M_t, \tag{61}$$

$$M_{\text{EACL}} = 31p \text{ per node.} \tag{62}$$

For N nodes in a network

$$M_{\text{EACL}} = 31N \text{ packets.} \tag{63}$$

The communication overheads are also given in Table 7. We see that BACL without clustering refinement has the lowest communication overhead. BACL with clustering has better localization accuracy, but also uses more messages, hence increasing communication and energy overhead. EACL has the worst communication overhead.

5.3 | Processing Energy

For the processing effort, we consider four one-hop neighbors as in the setting in Fig. 10. We also consider most calculations done by the nodes for localization equivalent, denoted by c. Let the computations of x and y coordinates between reference node and unknown sensor node denoted by I_x and I_y, respectively and threshold computation as I_{th}.

For BACL, the number of computations required is given by I_{BACL}. During the localization phase, I_x, I_y are needed to compute the coordinates

$$I_{\text{BACL}} = (I_x + I_y), \tag{64}$$

$$I_{\text{BACL}} = 1c + 1c = 2c \text{ per node.} \tag{65}$$

For N nodes, without refinement, the number of computations is given by

$$I_{\text{BACL}} = 2N. \tag{66}$$

During localization refinement phase, four pairs $(V_p^{(1)}, V_p^{(2)})$, $(V_p^{(2)}, V_p^{(3)})$, $(V_p^{(3)}, V_p^{(4)})$, and $(V_p^{(4)}, V_p^{(1)})$ participate in cluster head refinement. To calculate the cluster head coordinates and the average, it needs

$$I_{\text{BACL}} = 4(I_x + I_y) + I_{\text{avg}}, \tag{67}$$

$$I_{\text{BACL}} = 4(1c + 1c) + 1c = 9c \text{ per node.} \tag{68}$$

For N nodes, with refinement, the number of computations is

$$I_{\text{BACL}} = 9N. \tag{69}$$

Similarly, the network level overhead for EACL is I_{EACL}. Each node calculates the confidence value and compares it with the threshold

$$I_{\text{EACL}} = I_x + I_y + I_{\text{th}}, \tag{70}$$

which are repeated three times to select the best anchors and refine the errors

$$I_{\text{EACL}} = 3(I_x + I_y + I_{\text{th}}), \tag{71}$$

$$I_{\text{EACL}} = 3(3c) = 9c \text{ per node.} \tag{72}$$

Computation for N nodes

$$I_{\text{EACL}} = 9N. \tag{73}$$

TABLE 8 Computation overhead for localization methods for a network with N nodes

Method	**Overhead** (packets)
EACL	$9N$
URPE	$6N$
BACL	$2N$ without refinement
BACL	$9N$ with refinement

URPE considers 3 non-collinear nodes

$$I_{\text{URPE}} = (I_x + I_y + I_{\text{th}}), \tag{74}$$

$$I_{\text{URPE}} = (1c + 1c + 1c) = 3c. \tag{75}$$

In URPE, the best anchor nodes are selected depending on their error value and the location of the nodes is calculated again. So, for two passes

$$I_{\text{URPE}} = (3c)(2) = 6c \text{ per node} \tag{76}$$

and for the entire network, the total computation is given by

$$I_{\text{URPE}} = 6N. \tag{77}$$

The computation overhead comparison is given in Table 8. We see that BACL without clustering refinement has the lowest computation overhead. BACL with clustering has better localization accuracy, but also requires more processing. URPE has the worst computation overhead.

6 | CONCLUSION

Localization with bilateration is lightweight, since it requires only two reference nodes. But it is rarely used due to the difficulty in filtering the mirror nodes. The proposed method, BACL, effectively filters the mirror nodes by comparing with the coordinates of the reference nodes as localization process progresses in a single direction only. Error accumulation in the network can be reduced using clusters to refine node positions. Simulation results show that BACL localization error is 53% less than URPE and 40% less than EACL for 10% error in the distance evaluation based on the RSSI indication. The localization error of each method increases when the distance evaluation error increases. For 30% RSSI-based distance evaluation errors, the localization error increases increases by 5% for URPE, 10% for EACL, and 13% for BACL, the latter remaining the most accurate. BACL localization error decreases with the increase of the number of field sides on which anchor nodes are placed. The lowest localization error is obtained when the anchor nodes are placed on all four sides of the network. The results also indicate that localization error with cluster-based refinement with anchor nodes only on one side is comparable with the localization error without refinement with anchors placed on two sides. Localization refinement using clustering reduces the errors at the expense of more communication and computation, but performs better than URPE and EACL state-of-the-art methods. There is a trade-off between communication and computation overheads, and the number of anchors in the network. Future work will focus on reducing the error accumulation, allowing for either fewer anchors or less communication overhead during the cluster-based refinement.

References

[1] Mihai T. Lazarescu. Design of a wsn platform for long-term environmental monitoring for iot applications. IEEE journal on emerging and selected topics in circuits and systems, 3(1):45–54, 2013.

[2] Md Zakirul Alam Bhuiyan, Guojun Wang, and Athanasios V. Vasilakos. Local area prediction-based mobile target tracking in wireless sensor networks. IEEE Transaction on computers, 64(7):1968–1982, 2015.

[3] Xuefeng Liu, Jiannong Cao, and Shaojie Tang. Fault tolerant complex event detection in wsns:a case study in structural health monitoring. Proceedings IEEE INFOCOM, pages 1384–1392, 2013.

[4] Cem Ayyildiz, H. Emre Erdem, Tamer Dirikgil, Oguz Dugenci, Taskin Kocak, Fatih Altun, and V. Cagri Gungor. Structure health monitoring using wireless sensor networks on structural elements. Ad Hoc Networks, 82:68–76, 2019.

[5] Sahar Idwan, Junaid Ahmed Zubairi, and Imran Mahmood. Smart solution for smart cities using wireless sensor network for smart dumpster management. International conference on collaboration technologies and systems, pages 493–497, 2016.

[6] Mihai T. Lazarescu. Design and field test of a wsn platform prototype for long-term environmental monitoring. Sensors, 15(ISSN 1424-8220):9481–9518, 2015.

[7] H Yunmin Cho and Younggoo Kwon. Rss-based indoor localization with pdr location tracking for wireless sensor networks. International journal of electronics and communication, 70:250–256, 2016.

[8] G. Morral and P. Bianchi. Distributed on line multidimensional scaling for self localization in wireless sensor networks. Signal processing, 120:88–98, 2016.

[9] Angel Stanoev, Sonja Filiposka, Visarath In, and Ljupo Kocarev. Co-operative method for wireless sensor network localization. Ad-Hoc networks, 40:61–72, 2016.

[10] Randa M. Abelmoneem and Eman Shaaban. Locally centralized SOCP based localization technique for wireless sensor network. Procedia computer science, 73:76–85, 2015.

[11] Hao Xu, Huafai Sun, Yongqiang Chang, and Hao Liu. Wireless sensor networks loclization based on graph embedding with polynomial mapping. Computer networks, 106:151–160, 2016.

[12] Juan Cota Ruiz, Pablo Rivas P, Ernesto Sfuentes, and Rafel Gonzalez Landaeta. A recursive shortest path routing algorithm with application for wireless sensor network localization. IEEE sensors journal, 16(11):4631–4637, 2016.

[13] Xiaoyong Yan, Aiguo Song, Zhong Yang, and Wankou Yang. An improved multihop based localization algorithm for wireless sensor network using learning approach. Computers & Electrical Engineering, 48:247–257, 2015.

[14] Haider Safa. A novel localization algorithm for large scale wirless sensor networks. Computer communication, 45:32–46, 2015.

[15] Pinar Kirci and Hakima Chaouchi. Recursive and ad hoc routing based localization in wireless sensor network. Computer standards and interfaces, 44:258–263, 2016.

[16] Linqing Gui, Thierry Val, Anne Wei, and Rejane Dalce. Improvement of range free localization technology by a novel DV-hop protocol in wirless sensor networks. AdHoc Networks, 24:55–73, 2015.

[17] Chia Ho Ou. A localization scheme for wireless sensor networks using mobile anchors with directional antennas. IEEE sensor journal, 11(7):1607–1616, 2011.

[18] Guangjie Han, Jinfang jiang, Jia Chao, and Xuan Yang. Path planning for a group of mobile anchor nodes based on regular triangles in wireless sensor networks. Neurocomputing, 00:1–11, 2017.

[19] Guangjie Han, Chenyu Zhang, Jinfang Jiang, Xuan Yang, and Mohsin Guizani. Mobile anchor nodes path planning algorithms using network density based clustering in wireless sensor network. Journal of Network and Computer Applications, 85:64–75, 2017.

[20] Han Y, Zheng C, and Sun D. Localization of large scale underwater sensor networks based on recursive position estimation. OCEANS MTS/IEEE Washington, pages 1–4, 2015.

[21] Bouhdid B, Akkari W, and Belghith A. Energy aware cooperative localization approach for wireless sensor networks. IEEE Intelligent computer communication and processing, pages 429–435, 2017.

[22] Badia Bouhdid, Wafa Akkari, and Sofien Gannouni. Low cost recursive localization scheme for high density wireless sensor networks. International Journal on Semantic Web and Information Systems, 13(3):68–86, 2017.

[23] Danilo Spano and Fabio Ricciato. Opportunistic time of arrival localization in fully asynchronous wireless networks. Pervasive mobile computing, 37:139–153, 2017.

[24] Y. Zou, Q. Wan, and H. Liu. Semidefinite Programming for Tdoa Localization with Locally Synchronized Anchor Nodes. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3524–3528, 2018.

[25] Zhang Yong, He Peiyu, and Wang Haijiang. Coherent sources localization based on a two node distributed sensor netwroks. Signal processing, 126:103–110, 2016.

[26] Subrata Dutta, Nandini Mukherjee, and Sarmistha Neogy. Location estimation of wireless sensor nodes using a few beacon nodes in a 2 D geographical area. IEEE International conference on Applications and innovations in mobile computing, pages 81–88, 2015.

[27] Song Yang, Dong Enqing, Liu Wei, and Peng Xue. An iterative method of processing node flip ambiguity in wireless sensor networks node localization. International conference on information networking (ICOIN), 2016.

[28] Shuai Han, Jin Yue, Weixiao Meng, and Cheng Li. A novel flip ambiguities detection algorithm for wsn localization. IEEE Globecom Worshops, 2016.

[29] S. H. Lee, S. Lee, H. Song, and H. S. Lee. Gradual cluster head election for high network connectivity in large-scale sensor networks. 13th International Conference on Advanced Communication Technology (ICACT2011), pages 168–172, 2011.

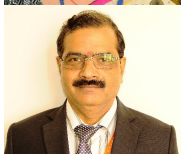
[30] S. Moad, M. A. Azim, N. Bouabdallah, and R. Langar. CDEEC: A connectivity Degree-based energy efficient clustering protocol for wireless sensor networks. IFIP Wireless Days (WD), pages 1–7, 2011.

[31] Younis and S. Fahmy. HEED: A hybrid, energy-efficient, distributedclustering approach for ad-hoc sensor networks. IEEE Transactions onMobile Computing, vol. 3(4):366–379, 2004.

AUTHOR BIOGRAPHY

Nikhath Tabassum received her M.Tech and B.E degree in 2009 and 2005 respectively. She is currently working towards her Ph.D degree in wireless sensor networks at REVA University. Her research interest include wireless sensor networks and antennas.

Geetha D. Devanagavi received her Ph.D, M.Tech, and B.E degree in 2014, 2005, and 1993 respectively. She is currently a Professor at REVA university. She has 21 years of teaching experience. Her research interest include wireless sensor networks, network security, and computer networks. She has good number of publications in reputed international journals.



Rajashekhar C. Biradar received his Ph.D, M.Tech, and B.E degree in 2011, 1997, and 1990 respectively. He is currently a Professor and Director of School of Electronics and Communication Engineering at REVA university. He has 27 years of teaching experience. His research interest include Ad-hoc networks, sensor networks, mesh networks, network security, and wireless sensor networks. He has good number of publications in reputed international journals. He has published 45 papers in peer reviewed national and international journals, 70 papers in reputed national and international conferences and 3 book chapters. He has been listed in Marquis' Who's Who in the World (2012 Edition), USA and Top 100 Engineers by IBC, UK. As per Google Scholar, he has more than 230 citations.

Mihai T. Lazarescu (M'92) received the Ph.D. degree in 1998 from the Politecnico di Torino, Italy, where he serves now as Professor Assistant. He was Senior Engineer at Cadence Design Systems working on high-level synthesis (HLS) of embedded systems, founded several startups working on real-time embedded Linux and WSN for long-term environmental monitoring, and participated in numerous European- and national-founded research projects on topics related to design of ASIC, EDA for HLS, and WSN hardware, software and development tools. He authored and co-authored more than 50 scientific publications at international conferences and journals, several books, and international patents. He is IEEE Senior Member, served as Guest Editor for ACM and MDPI journals, and as PC Chair for international conferences. His research interests include real-time embedded systems, low-power sensing and data processing for IoT and WSN platforms, high-level hardware/software co-design, and high-level synthesis.