BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **22**, No 1
Sofia • 2022 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2022-0002

Deterministic Centroid Localization for Improving Energy Efficiency in Wireless Sensor Networks

Sneha Vijayan[1], *Nagarajan Munusamy*[2]

[1]*Research Scholar, KSG College of Arts and Science, Coimbatore, India*
[2]*Principal, KSG College of Arts and Science, Coimbatore, India*
E-mails: snehamaelin05@gmail.com mnaagarajan@gmail.com

***Abstract*: *Wireless sensor networks are an enthralling field of study with numerous applications. A Wireless Sensor Network* (*WSN*) *is used to monitor real-time scenarios such as weather, temperature, humidity, and military surveillance. A WSN is composed of several sensor nodes that are responsible for sensing, aggregating, and transmitting data in the system, in which it has been deployed. These sensors are powered by small batteries because they are small. Managing power consumption and extending network life is a common challenge in WSNs. Data transmission is a critical process in a WSN that consumes the majority of the network's resources. Since the cluster heads in the network are in charge of data transmission, they require more energy. We need to know where these CHs are deployed in order to calculate how much energy they use. The deployment of a WSN can be either static or random. Although most researchers focus on random deployment, this paper applies the proposed Deterministic Centroid algorithm for static deployment. Based on the coverage of the deployment area, this algorithm places the sensors in a predetermined location. The simulation results show how this algorithm generates balanced clusters, improves coverage, and saves energy.*

***Keywords*: *Sensors, energy, centroid, DV Hop, WCL, deployment, clustering.*

1. Introduction

A wireless sensor network is a self-configured network that detects physical changes in the environment without the need for additional infrastructure. WSN applications include animal tracking, environmental monitoring, medical applications, military surveillance, and infrastructure maintenance. A typical WSN is made up of a group of sensor nodes that work together to monitor physical or ecological conditions and generate a large amount of data that must be routed from the network to the sink/base station via multiple hops. Amutha, Sharma and Nagar [1] discusses the application of seismic, thermal, acoustic, visual, magnetic, radar, and infrared sensors in a WSN environment. There are three types of sensors: static, mobile, and hybrid.

Static sensors do not move, whereas mobile sensors can move across the network to sense data, and hybrid sensors are a combination of static and mobile sensors. The primary concerns of WSNs include node energy levels dropping, node identification, hardware failure, node positions, network scalability, node deployment, and so on. WSN also has a range of benefits. It is self-configurable, modular, cost-effective, and lightweight, and it is based on event detection. Small-capacity batteries are used to power the sensor nodes. As a consequence, as Sneha and Nagarajan [2] points out, energy is a critical factor in deciding the lifetime of wireless sensor networks. In an unattended and uninhabited area, it is difficult to replace or recharge a sensor node's battery. As a consequence, any sensor node's lifespan is determined by its energy efficiency. One of the best ways to increase energy efficiency, according to Gnanaprasambigai [3] is to segment the entire network into many clusters. The network is divided into clusters, with one of the sensor nodes acting as cluster head. The cluster head oversees communication both within and outside of the cluster. Communication within a cluster is referred to as intra cluster communication. To put it another way, cluster members send data to cluster heads. Inter-cluster communication is the exchange of information between cluster heads and sink nodes. Finally, the sensed environment data from the sensor node is transmitted to the base station via the cluster head sensor nodes. It is necessary to know the location of the sensor nodes in order to be aware of the energy consumption of the nodes, as it contributes to energy efficiency, load balancing, and network coverage, as discussed in Ezhilarasi and Krishnaveni's paper [17]. The deployment of the sensor node is critical for improving the accuracy of the sensor node's location. The deployment of WSN nodes can be either static or dynamic. Static deployment locks the node into a position that cannot be changed during the lifetime of the network, whereas dynamic deployment allows the node's location to change in response to network changes. Though random deployment is now the norm, static deployment is still required when human intervention is not possible. When using WSN, coverage is an important factor to consider. To support connectivity and full coverage, massive static nodes are required. This can lead to increased power consumption, higher costs, and more complicated network management. Amutha, Sharma and Nagar [1] explains how coverage can be complete or partial. Full coverage implies that each deployed point is being monitored by at least one node. This is expensive and inefficient for some real-time applications. Partial coverage ensures coverage to some extent, which can save energy and increase network lifetime. The coverage model is calculated using the distance from the nearest point of interest. The localization algorithms provide location estimates that can be used to assess the coverage of the system. To improve scalability and reduce overhead caused by changes in topology, we use Location-Based Routing (LR) Protocols that rely on position information. The energy and power of nodes to extend network lifetime is also an important constraint in WSN, depending on where each node is located in the network. Ezhilarasi and Krishnaveni [4] discusses how nodes in a network must be aware of their neighbours in order to transmit a network-required message. There are several methods for locating a network node, each with advantages and disadvantages. The goal of a localization method is to determine the precise location of the node, which is difficult in most real-time applications. According to Nagarajan and Karthikeyan [18], when network nodes are aware of their location, data can be transmitted to a base station with the least amount of energy and in the shortest amount of time. We propose the Deterministic Centroid algorithm in this paper for deploying static nodes with predefined locations. Section 2 of this paper contains a review of the literature on existing Centroid algorithm deployments. Section 3 introduces the Deterministic Centroid algorithm for static deployment, Section 4 discusses the algorithm's performance using simulation results, and Section 5 summarizes Deterministic Centroid's contribution to improving energy efficiency and accuracy.

2. Related works

As a classic range-free localization algorithm, Bulusu, Heidemann and Estrin [5] proposed the Centroid algorithm. This algorithm is split into two parts. During the first phase, all anchor nodes send out packets with their location information to all other nodes in the threshold region. During the second phase, each unknown node determines its location by arithmetic mean of all anchor nodes coordinates within the threshold region. That represents by the equations

(1) $$x_u = \frac{\sum_{i=1}^{m} x_i}{m}.\, y_u = \frac{\sum_{i=1}^{m} y_i}{m},$$

where: (x_i, y_i) are the coordinates of the anchor node i; u is the unknown node; (x_u, y_u) is the location of the unknown node; m is the total number of anchor nodes that are within the threshold region. This algorithm is simple to implement, but it does not produce accurate results and necessitates a complex method to determine the threshold value.

Nichelsu and Nath [6] has proposed the DV-Hop Algorithm (Distance Vector-Hop) to calculate the approximate distance between two nodes by multiplying the average hop distance by the number of hops. It is broken down into three stages. During the first phase, each anchor sends a packet containing its location information and hop count value to its neighbouring nodes. The nodes that receive this packet then send it to their nearest neighbours after multiplying the hop count value by one. Each anchor provides a minimum hop count value to all nodes in the network, and the location information for each anchor is stored in a hop count table. In the second phase, each anchor calculates the average hop distance by the equation

(2) $$\mathrm{AvgHopDis}_i = \sum_{i=1, i\neq j}^{m} \sqrt{\frac{(x_j - x_i)^2 + (y_j - y_i)^2}{\sum_{i=1, i\neq j}^{m} h_{ji}}},$$

where: m is the total number of anchors in the given network; i is the ID of each anchor; h_{ij} is the minimum number of hop counts between anchor i and anchor j; (x_i, y_i) and (x_j, y_j) represent coordinates of anchors i and j; $\mathrm{AvgHopDis}_i$ is the average distance of a hop computed by anchor i. Then each unknown node u computes approximate distance from anchor node i using the equation given below:

(3) $$d_{ui} = \mathrm{AvgHopDis}_i \times h_{ui}.$$

The unknown node's location is determined using a multilateration method after the distance from each anchor is determined. This algorithm requires more energy to compute, which leads to more localization errors.

Zhang, Ji and Shan [7] has proposed the DV Hop-based Weighted Centroid algorithm, which reduces DV Hop localization's computational complexity and power consumption. This algorithm is broken down into two stages. During the first phase, each node receives the minimum hop count value from each anchor node. In the second phase every unknown node *u* finds its location (x_u, y_u) using the equations

$$x_u = \frac{\sum_{i=1}^{m} w_i x_i}{\sum_{i=1}^{m} w_i},\ y_u = \frac{\sum_{i=1}^{m} w_i y_i}{\sum_{i=1}^{m} w_i}, \tag{4}$$

where: $w_i = \frac{1}{h_{ui}}$ is the weight of each anchor i; h_{ui} is the minimum hop count value of node *u* from anchor i; *m* is the total number of anchor nodes. The weight is inversely proportional to the number of hops. This has been used to increase the weight of the nearest anchor. The anchor with the fewest hops is closer to the given node and thus has a greater influence on determining its location. Anchor nodes, unlike DV Hop, do not broadcast packets containing the average hop distance to other nodes. As a result, this algorithm has low computational complexity and uses less power, but accuracy in localization must be taken into account.

The Centroid value of each tetrahedron is used Centroid Algorithm for a tetrahedron from Wang Chang-zheng, Wen-liang Tang and Yan Xu [8]. The unknown nodes' locations have been determined. The nodes are initially deployed randomly in tetrahedrons in this case, and the Centroid is used to localize the nodes. Despite improved accuracy, this algorithm is more prone to errors. Because the coordinates of unknown nodes must be calculated in many rounds, the energy consumption is extremely high. Novel Centroid Localization Algorithm (Chen and Liu [9]) calculates the position of unknown sensor nodes based on the connectivity relationship between nodes. The unknown nodes estimate their position using the landmarks' coordinates. The estimated location is the centroid of the polygon formed by several landmarks. Chen and Liu [9] has proposed Oval Centroid algorithm, which is also similar to the 2D weighted Centroid Algorithm, which employs weights to calculate sensor node positions. It makes use of an accuracy control factor PCF (Pair Correlation Function), which can be set to 0 or 1. Chen et al. [10] has proposed the Novel 3D Centroid algorithm, which is based on a 3D coordinate system and takes into account geometric relationships as well as communication limitations between sensor nodes and sites. The estimated position of the unknown node is in the centre of the 3D plane diagram. When the distinguishing density exceeds 6, this algorithm provides greater precision. Chen et al. [10] proposes a new Centroid Algorithm for 3D WSNs that make use of the volume coordinate system's coordinate tetrahedron. All anchors broadcast their positions to all sensor nodes within their transmitting range, and the sensor nodes collect all signals received from various reference points. The Centroid algorithm is used to calculate the barycenter of each tetrahedron, and the average coordinates of these barycenters are used to calculate the final estimated location of each unknown node. This algorithm aids in the elimination of localization errors. Deng et al. [11] has proposed the SA-Centroid algorithm, which operates in three steps. In the first level, each node keeps a table and communicates with its neighbours. The sensor nodes in the second level receive notifications from their nearest landmark. This landmark node joins two other landmark nodes to form a triangle, and the triangle's centroid is determined. The Centroid value is used by the sensor node to calculate its temporary position. In the third step, this temporary position estimate is used to shape a polygon, and the Centroid is calculated again to compute the exact location estimate. This algorithm increases the computational cost, and it also requires symmetrical anchor node reference to improve accuracy. Xu et al. [12] has proposed the 2D Weighted Centroid algorithm, which takes into account anchors with the same weight that can be found inside a circle. Anchors that are not useful for the localization phase are discarded to eliminate RSSI error. This algorithm works better on large-scale WSNs, and its accuracy improves only when the anchor weight is set to 0 or 1. Because the estimate is based on distance, it is possible the results to be unreliable. Weighted Centroid-Improved Particle Filter Algorithm (Zhang, Feng and Guo [13]) is used to determine the improved particle filter's initial estimation point. The number of anchor nodes and the sensing data determine the location of the sensor node. This algorithm is quick to localize and works well with a small number of beacons. The Centroid Algorithm for 2D and 3D by Xu et al. [12] combines the Centroid algorithm and RSS to estimate the position of unknown nodes. Anchors, which are static nodes, are used to obtain RSS values from unknown nodes, and the best possible position is estimated using an improved weighted Centroid algorithm. Distance thresholding has been proposed to improve positioning accuracy. The 2D Weighted Centroid algorithm's threshold is the radius of a circle whose centre is an unknown node. This algorithm applies data from anchors inside the circle to them. The weight of these anchors is the same whether they are inside or outside the circle. This algorithm can be used when anchors and unknown nodes are on the same plane. The 3D Weighted Centroid Localization algorithm developed by Xu et al. [12] is an extension of the 2D Weighted Centroid Localization algorithm. The coordinate information of the anchor nodes on the *z*-axis is used here. In this algorithm, the threshold is the radius of the sphere whose centre is an unknown node. Since 3D-WCL has high localization accuracy and a low hardware cost, it is suitable for a wide range of real-time applications.

Weighted Centroid Localization Algorithm (Shi [14]) converts RSSI data from unknown nodes into distance and then uses the negative square of the distance ratio to calculate a mend weight. The modified weight is used to improve positioning precision. The specific steps covered by this modified algorithm are as follows.

1. The beacon sends relevant packets to the surrounding environment on a regular basis.

2. RSSI messages are sent to blind nodes by beacon nodes.

3. Determine the set of anchor nodes that are received by blind nodes and assign RSSI values to them.

4. The blind point coordinates are roughly confirmed.

5. The average positioning error and non-beacon positioning error are computed.

This algorithm makes the best use of the RSSI data, reducing the deficiency to some extent. This method outperforms traditional methods in terms of positioning. The positioning precision is excellent, and the positioning error is minimal. This algorithm, however, does not account for energy consumption or calculation amount. Kaur, Kumar and Gupta's [15] Enhanced Weighted Centroid DV Hop Localization (EWCL) takes into account a novel weight computation method. It computes weight by taking into account various factors such as communication radius, the influence of various anchors, and the proximity of a given node's anchors. EWCL not only saves power but also improves localization errors. This is carried out in three stages. The first phase determines the minimum number of hops counted by each node from each anchor. In the second phase, each anchor's average distance per hop is calculated. In the third phase, the location of the unknown nodes is computed. By taking into account the communication radius and the nearest anchor node, the EWCL algorithm improves localization accuracy. By limiting the broadcasting range, it reduces power consumption in the first two phases.

3. Deterministic centroid model description

The Deterministic Centroid algorithm is proposed for statically deploying sensors in areas where human intervention is prohibited. A Wireless Sensor Network (WSN) is a network of nodes that sense, aggregate, and transmit data. The proposed model divides the entire network into clusters. There are five sections in this model: node deployment, Cluster Head (CH) selection, cluster formation, next CH selection, and communication. The following constraints apply to the proposed work:

1. The sensor nodes will be deployed deterministically based on the calculated Centroid.
2. There is only one sink in the sensing region.
3. The sensor nodes and sink are both fixed.
4. The initial energy is the same for all network nodes.
5. The sink node knows the coordinates of all the member nodes.
6. Using Hop count, each node is aware of its neighbors.

The following are the advantages of the D-Centroid Algorithm over existing ones:

1. The sensors are distributed evenly across the deployment area, with equal spacing between them.
2. Because the entire network is divided into grids and sensors are strategically placed throughout the grids, the network is completely covered.
3. Because the nodes' locations are fixed, node failures can be easily detected.
4. Data is transmitted to the base station via cluster heads, which can be single hop or multihop.
5. Prior to deployment, the optimal number of cluster heads is determined.
6. Cluster heads are rotated based on the residual energy of the nodes as well as their distance from the base station.
7. Because each node in the network is aware of its neighbors prior to data transmission, data delay is reduced.

8. Cluster formation results in balanced clusters, which increase network lifetime.

The proposed work divides the entire network area into equal-sized squares to form a grid that spans the entire network. This grid concept can aid in ensuring network coverage across the network's area. The Deterministic Centroid algorithm is implemented in three stages. The first is sensor node positioning, followed by cluster formation, and finally communication. In the first phase, the entire sensing area is divided into a grid, and the coordinate values of each square in the grid are passed to the proposed algorithm, which computes the Centroid position for each square. For each square, the centroid is calculated using the following formula:

(5) $$\text{Centroid}_{(x,y)} = \left(\frac{(x_1+x_2+x_3+x_4)}{4}, \frac{(y_1+y_2+y_3+y_4)}{4}\right).$$

The sensors are positioned based on the Centroid obtained for each square. This procedure will be repeated until all of the sensors are evenly distributed across the grid.

Pseudo code 1. Grid Formation and Deployment

Input: l_a, b_a
Output: Grid (M) formation and Node deployment (N)
Procedure Grid Formation
Let area$_a$= (l_a * b_a)
Compute GridMinimumCut M_i = (l_a * b_a – 1)
Partition area$_a$ into M_i equal sets M
For each M
Compute Centroid$_C$= $\left(\frac{x_1+x_2+x_3+x_4}{4}, \frac{y_1+y_2+y_3+y_4}{4}\right)$
For each Centroid$_C$
Deploy N_i
End for
End for
End procedure

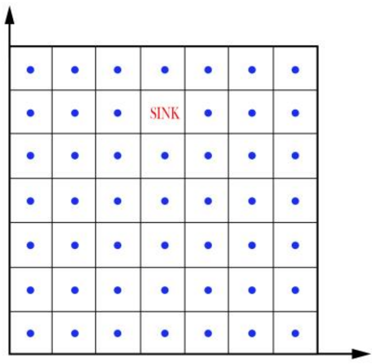

Fig. 1. D-Centroid node deployment

The optimal number of cluster heads required for the sensing area is determined in the second phase. Cluster heads for the sensing area are chosen based on the distance between the sensor nodes and the sink. Sathyapriya and Arumugam [16] discusses how to calculate the optimal cluster heads needed for the sensing area,

$$O_{\mathrm{ch}} = \frac{\max(\mathrm{dist}(N_i, S))}{\mathrm{Tr}/2}, \tag{6}$$

where O_{ch} is the optimal cluster heads, N_i represents the sensor nodes, S is the Sink and Tr is the transmission radius. The number of cluster heads selected is one important criterion to save energy. Because the initial cluster heads are chosen based on their distance from the sink, the cluster formation will be done with balanced clusters, which will increase the network's lifetime. Each CH broadcasts its coordinates as well as its ID to all network nodes. Each member node receives the coordinates from the CHs and uses Hop count to calculate the distance between itself and the CH. Following the computation of distances, each member node joins the CHs with the shortest distance, and clusters are formed. Each cluster's average number of nodes is discovered to be equal. The following are the benefits of the method discussed in paper [16]:

1. Load balancing can be accomplished using balanced clusters.
2. It reduces CHs' energy consumption.
3. It extends the network's lifetime.

After identifying the best cluster heads, the other nodes use Hop count to calculate their distance from the best cluster heads. The sensor nodes communicate with the nearby cluster head to ensure efficient transmission, avoiding delays, reducing energy consumption, and extending network lifetime.

Pseudo code 2. Cluster Formation

Input: N_i, S, Tr
Output: Cluster Heads (CH)
Procedure Cluster Formation
For each N_i
Do
Compute distance D from N_i to Sink S
Compute Optimal $O_{\mathrm{ch}} = \frac{\max(\mathrm{dist}(N_i, S))}{\mathrm{Tr}/2}$
End for
CH_j broadcasts (X_{CH}, Y_{CH}) and $\mathrm{ID}_{\mathrm{CH}}$
While (j=0; j++)
For each N_i
Compute Hopcount from N_i to CH_j
If Hopcount<=10
Form cluster with CH_j
Else
Compute Hopcount from N_i to Next CH_j
Until $j \le O_{\mathrm{ch}}$
Repeat until all N_i are included in clusters
End if
End for
End while
End procedure.

The initial cluster head selection is done deterministically based on distance from the sink, but as the data transmission and aggregation process progresses, the residual energy of the cluster heads is checked in each round. The selected CH will continue to collect data and send it to the sink until the energy reaches a certain threshold.

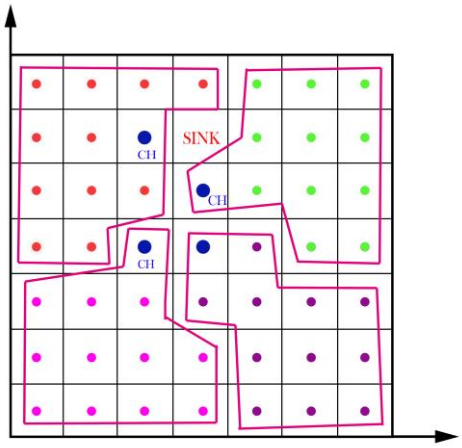

Fig. 2. Deterministic Centroid cluster formation

When the energy level of the cluster heads is less than the predefined threshold value, it remains the CH. If the residual energy reaches the threshold, the CH requests residual energy and the distance to the sink from the cluster's other member nodes. The initial and residual energy are used to calculate the energy Centroid coordinates. The member node with the highest residual energy and the closest proximity to the sink is chosen as the next cluster head. This process is repeated until all of cluster's nodes drain out. The energy Centroid coordinates are calculated using the equations discussed in paper [16],

$$X_{\mathrm{e}} = \frac{\sum_{m=0}^{n} \frac{E_{\mathrm{re}}}{E_0}.X}{n}, \tag{7}$$

$$Y_{\mathrm{e}} = \frac{\sum_{m=0}^{n} \frac{E_{\mathrm{re}}}{E_0}.Y}{n}, \tag{8}$$

where: E_0 is the initial energy of the sensor nodes; X, Y are the coordinates of sensor nodes; E_{re} is the residual energy of the sensor nodes; n is the number of nodes in the sensing region; X_{e} and Y_{e} are the energy Centroid coordinates.

The energy threshold value for each cluster is derived from paper [16] using the equation

$$\mathrm{Energy}_{\mathrm{thrs}} = \mathrm{DP} \times E_{\mathrm{elec}} \times \left(\frac{N}{C} - 1\right) + \mathrm{DP} \times E_{\mathrm{dat}} \left(\frac{N}{C}\right) + \mathrm{DP} \times E_{\mathrm{elec}} + \mathrm{DP} \times \varepsilon_{\mathrm{fs}} d^2, \tag{9}$$

where: E_{elec} is the initial energy to run transmitter electronics; E_{dat} is the energy required for data aggregation; d is the distance; N is the number of nodes in the sensing area; C is the number of clusters; $\varepsilon_{\mathrm{fs}}$ is the free space model; DP is size of the data packet.

Pseudo code 3. Selection of next CH
Input: C
Output: New Cluster Head
Procedure New Cluster Head
For each N_i in C_i
Compute Energy Centroid (X_e,Y_e) for all C
Find Residual Energy of all N_i
Compute Distance for all N_i to Sink
Calculate Energy threshold $\text{Energy}_{\text{thrs}}$ for each C
If ($\text{CH}_{\text{residual}}$ >$\text{Energy}_{\text{thrs}}$)
Then N_i remains Cluster Head
Else
Next N_i with d_{min} and $E_{\text{residual-max}}$ selected as CH
End if
End for
End procedure.

The CH collects data from the cluster's member nodes, aggregates it, and sends it to the sink. To forward the collected data to the sink the Disthrs is calculated using free space and multipath model. The distance threshold used in the following equation determines whether the data communication is single hop or multihop:

$$\text{Dis}_{\text{thrs}} = \sqrt{\frac{\varepsilon_{\text{fs}}}{\varepsilon_{\text{mp}}}}, \tag{10}$$

where ε_{fs} is the free space model and ε_{mp} is the multipath model.

If the distance to the sink is less than the distance threshold, the CH communicates with the sink directly (i.e., using single hop), otherwise it communicates with the sink via the intermediate CHs (i.e., using multihop).

Pseudo code 4. Data Communication
Input: CH, Sink
Procedure Communication
For each CH
Compute $\text{dis}_{\text{ch}(i)}$ to sink
Compute Dis_{thrs}
If ($\text{dis}_i < \text{Dis}_{\text{thrs}}$)
Then CH directly send data to Sink
Else
Select nearest neighbor CH with min dis_i to Sink
Send data packets to the neighbor CH
Neighbor CH send data packets to Sink
End if
End for
End procedure

4. Simulation results

The proposed method has been tested with a hundred nodes in the environment depicted in Table 1. The sensor nodes are distributed at random across a 100×100 m field. The sink node is assumed to be in the centre of the field's top periphery. Each cluster will contain approximately 25% of the total nodes in this work. Cluster structure, network lifetime, and energy efficiency are computed and compared with the SA-Centroid, Improved WCL, and ECWL algorithms.

Table 1. Simulation parameters

Parameter	Value
Network size	100×100
No of sensor nodes	100
Radio propagation range	300 m
Channel capacity	2 Mbits per 1s
Initial energy	1 J
Data packets	3200 bits
Distance threshold	85 m
Simulation time	180 s
ε_{fs}	10 pJ per 1 bit per 1 m^2
ε_{mp}	0.0013 pJ per 1 bit per 1 m^4

Table 2. Number of nodes in each cluster including cluster head, using SA-Centroid, Improved WCL, ECWL and Proposed Algorithm

SA-Centroid				Improved WCL Algorithm				ECWL Algorithm				Proposed Algorithm			
C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
32	30	17	21	28	22	30	20	32	27	23	18	25	25	26	24
30	28	23	19	30	25	27	18	27	32	18	23	24	26	25	25
22	26	29	23	26	24	29	21	32	23	19	26	25	26	24	25
15	20	31	34	27	23	27	23	21	29	25	25	24	26	24	26
25	24	30	21	25	24	28	21	25	24	29	22	25	24	26	25

Cluster structure. The proposed approach produces balanced clusters (i.e., the number of member nodes in each cluster is equal). SA-Centroid [11], Improved WCL [14], and ECWL [15] are used to compare cluster formation. The SA-Centroid and Improved WCL distribute the sensor nodes in the deployment area at random. The CH is chosen at random in SA-Centroid, and the location of the unknown nodes is computed using SA-Centroid. The CHs in Improved WCL are computed using the weight function. ECWL chooses clusters based on their energy Centroid. The proposed method deploys all nodes deterministically where their location is known ahead of time. The CHs are chosen based on their proximity to the sink. Fig. 3 shows that the clusters are unbalanced. Initially, clusters are formed at random, and unknown nodes in the deployment area calculate the Centroid and join with the CH closest to the Centroid coordinate. The clusters are not stable after each round, which means that the nodes keep switching to each cluster. The disadvantage of this approach is that the path for data transmission must be computed after each round because the location of the nodes changes. The clusters must adapt to the network's load. Fig. 4 depicts the Improved WCL algorithm's cluster formation. Weight is used in this algorithm to calculate the distance between nodes. There are several anchor nodes deployed, and the unknown node calculates its position using the coordinates of the anchor nodes. If an unknown node is within the range of several anchors (more than three anchors), the unknown node is considered to be within the range of intersection anchor circles. Based on geographic boundaries, the algorithm generates clusters with a predefined shape, preferably a hexagon. Each cluster chooses one node to serve as the cluster head. The cluster head keeps a list of all adjacent clusters and the locations of their member nodes. Clusters are formed at random in this approach, and the cluster heads are elected by the cluster members, which can increase energy consumption. The clusters are unbalanced, and each round necessitates a re-clustering.

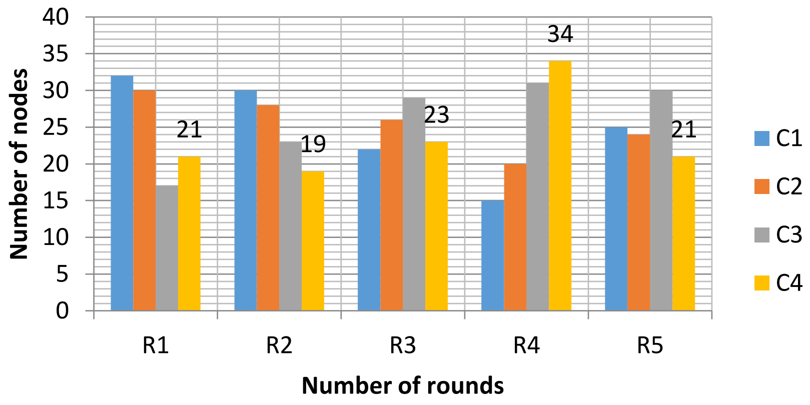

Fig. 3. SA-Centroid clustering approach

Fig. 5 depicts the cluster formation of ECWL, which does not produce significant unbalanced clusters. The cluster heads in this algorithm are not chosen at random, but rather based on the energy Centroid coordinate. This algorithm has proven to be more effective at load balancing. The re-clustering is performed based on the energy drain, with both the energy Centroid and the distance Centroid calculated. This algorithm, however, does not take coverage into account.

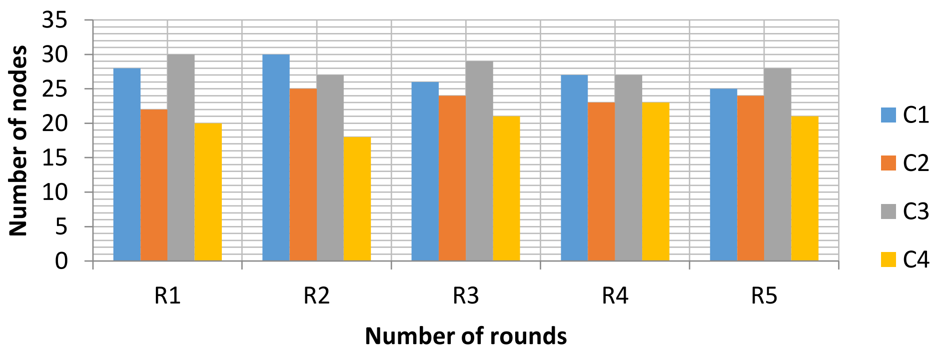

Fig. 4. Improved WCL Clustering approach

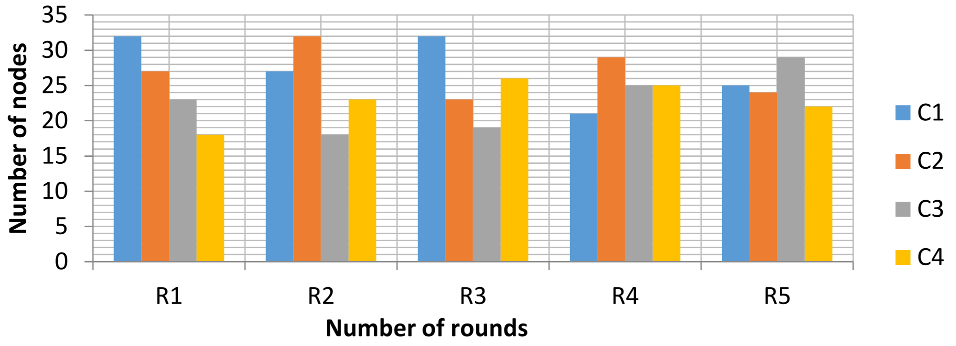

Fig. 5. ECWL Clustering approach

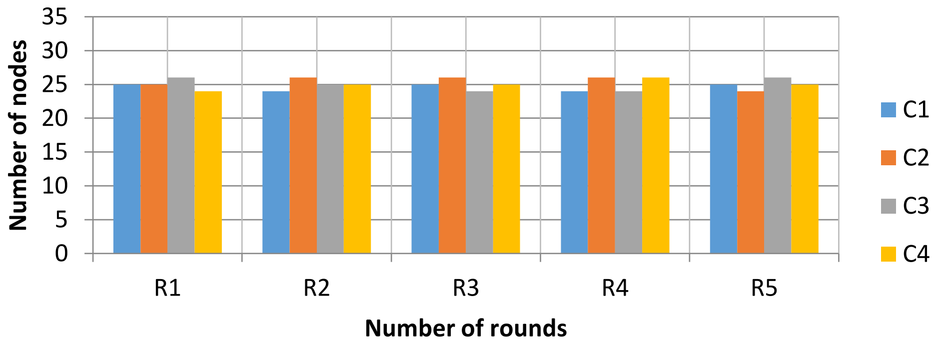

Fig. 6. Proposed D-Centroid approach

Fig. 6 depicts the proposed D-Centroid Algorithm forming balanced clusters to improve load balancing. Because this algorithm employs static deployment, the sink is aware of the location of all nodes prior to data transmission, which can help to reduce packet drop. This algorithm takes into account the deployment area's coverage and thus positions the sensors uniformly throughout the sensing area. The residual energy of the nodes is used to re-cluster them, and the distance to the sink is also taken into account. Because this algorithm generates balanced clusters, the load on the CHs is distributed evenly across all clusters, potentially increasing the network's lifetime.

Fig. 7 compares the energy consumption of the SA-Centroid, Improved WCL, ECWL, and the proposed D-Centroid algorithms. The amount of energy consumed is determined by the sensing environment, data computation, and data transmission. As a result, the following equation is used to calculate the energy consumption of each node in the network,

(11) $$E_{\text{consumed}} = E_{\text{initial}} - E_{\text{residual}},$$

where E_{consumed} is the total energy consumption of the node, E_{initial} is the initial energy of the node, and E_{residual} is the remaining energy of the node. After each round, the residual energy of the nodes is calculated. As a result, the total network's energy consumption can be calculated using the following equation:

(12) $$\sum_{c=1}^{N} E_{\text{consumed}} = \sum_{i=1}^{N} E_{\text{initial}} - \sum_{r=1}^{N} E_{\text{residual}}.$$

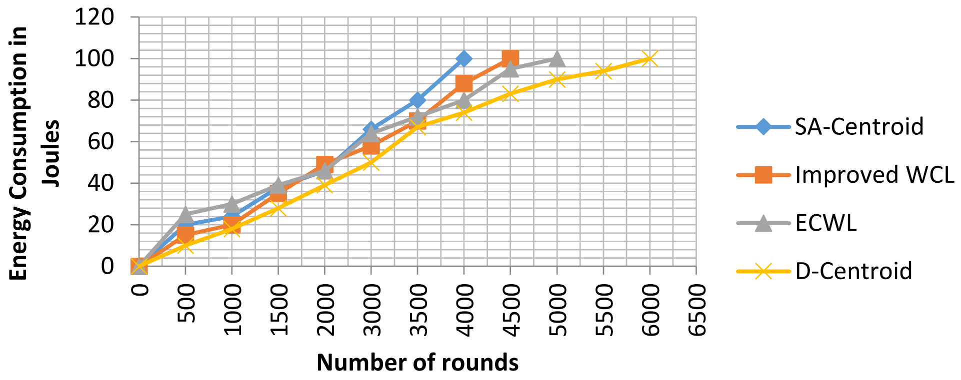

Fig. 7. Comparison of energy consumption

Fig. 7 shows that the proposed algorithm consumes 50% of the energy after 3000 rounds, whereas SA-Centroid consumes 50% after 2000 rounds, Improved WCL consumes 50% after 2300 rounds, and ECWL consumes 50% after 2700 rounds. Because the proposed algorithm's communication is based on residual energy and distance to the sink, the energy required to transmit data is reduced, resulting in an increase in network lifetime. The residual energy comparison in Fig. 8 demonstrates that the proposed approach has the highest residual energy even after 4500 rounds. This demonstrates that energy is balanced throughout the network, which can effectively balance network load while also contributing to an increase in network lifetime.

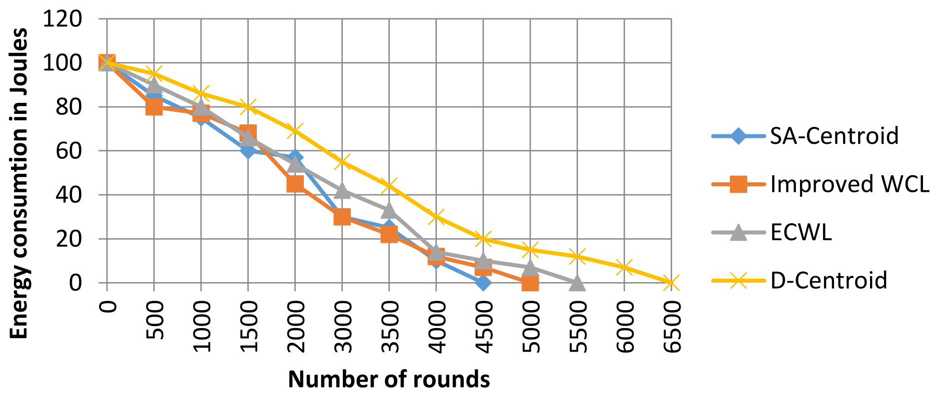

Fig. 8. Comparison of residual energy

5. Conclusion

In this paper the static deployment of nodes in wireless sensor networks is considered, and an energy-efficient static deployment algorithm D-Centroid is proposed. The entire deployment area is covered while deploying the nodes, and because it is a static deployment strategy, the sink knows the location of all nodes in the network. This algorithm determines the optimal number of cluster heads based on the number of deployed nodes. This algorithm balances the clusters it creates, thereby balancing network load and energy consumption. As the distance to the sink is taken into account before transmission, the energy required for transmission is balanced across all clusters, increasing network lifetime. The proposed algorithm outperforms SA-Centroid by 100%, Improved WCL by 49 % and ECWL by 28 %. Since all cluster nodes are given the opportunity to become CHs, the intermediate CHs should be used for multihop communication in the final rounds. We are attempting to improve data aggregation techniques by incorporating a mobile sink into this algorithm.

References

1. Amutha, J., S. Sharma, J. Nagar. WSN Strategies Based on Sensors, Deployment, Sensing Models, Coverage and Energy Efficiency: Review, Approaches and Open Issues. – Wireless Personal Communications, Vol. **111**, 2020, pp. 1089-1115. **https://doi.org/10.1007/s11277-019-06903-z**
2. Sneha, V., M. Nagarajan. Localization in Wireless Sensor Networks: A Review. – Cybernetics and Information Technologies, Vol. **20**, 2020, No 4, pp. 3-26.
3. Gnanaprasambikai, Nagarajan Munnusamy. Data Pre-Processing and Classification for Traffic Anomaly Intrusion Detection Using NSLKDD Dataset. – Cybernetics and Information Technologies, Vol. **18**, 2018, No 3, pp. 111-119.
4. Ezhilarasi, M., V. Krishnaveni. A Survey on Wireless Sensor Network: Energy and Lifetime Perspective. – Taga Journal, Vol. **14**, 2018, pp. 3099-3113. ISSN: 1748-0345.
5. Bulusu, N., J. Heidemann, D. Estrin. GPS-Less Low Cost Outdoor Localization for Very Small Devices. – IEEE Pers. Commun. Mag., Vol. **7**, 2000, No 5, pp. 28-34.
6. Niculescu, D., B. Nath. Ad-hoc Positioning System. – In: Proc. of IEEE on Global Telecommunications Conference, Vol. **5**, 2001, pp. 2926-2931.
7. Zhang, B., M. Ji, L. Shan. A Weighted Centroid Localization Algorithm Based on DV-Hop for Wireless Sensor Network. – In: Proc. of 8th International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 2012, pp. 1-5.
8. Wang, Chang-zheng, Wen-liang Tang, Yan Xu. 3D Centroid in Tetrahedron Localization Algorithm for Wireless Sensor Network. – Transducer and Microsystem Technologies, Vol. **8**, 2012.
9. Chen, M., H. Liu. Enhance Performance of Centroid Algorithm in Wireless Sensor Networks. – In: Proc. of 4th International Conference on Computational and Information Science, Chongqing, China, 2012, pp. 1066-1068. DOI: 10.1109/ICCIS.2012.144.
10. Chen, H., P. Huang, M. Martins, H. C. So, K. Sezaki. Novel Centroid Localization Algorithm for Three-Dimensional Wireless Sensor Networks. – In: Proc. of 4th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, 2008, pp. 1-4.
11. Deng, B., G. Huang, L. Zhang, H. Liu. Improved Centroid Localization Algorithm in WSNs. – In: Proc. of 3rd International Conference on Intelligent Systems and Knowledge Engineering, Xiamen, China, 2008, pp. 1260-1264. DOI: 10.1109/ISKE.2008.4731124.

12. X u, L. W a n g, Y. J i a n g, F. Y a n g, Y. D u, Q. L i. A Study on 2D and 3D Weighted Centroid Localization Algorithm in Wireless Sensor Networks. – In: Proc. of 3rd International Conference on Advanced Computer Control, IEEE, 2011, pp. 155-159.
13. Z h a n g, J., X. F e n g, J. G u o. A Hybrid Localization Algorithm Based on Error Correction. – In: Proc. of IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), IEEE, 2019, pp. 792-797.
14. S h i, H. A New Weighted Centroid Localization Algorithm Based on RSSI. – In: Proc. of 2012 IEEE International Conference on Information and Automation, IEEE, 2012, pp. 137-141.
15. A m a n p r e e t, K., P. K u m a r, G. P. G u p t a. A Weighted Centroid Localization Algorithm for Randomly Deployed Wireless Sensor Networks. – Journal of King Saud University-Computer and Information Sciences, Vol. **31**, 2019, No 1, pp. 82-91.
16. S a t h y a p r i y a, L., J. A r u m u g a m. Energy Centroid Clustering Algorithm to Enhance the Network Lifetime of Wireless Sensor Networks. – Multidimensional Systems and Signal Processing, Vol. **31**, 2020, No 3, pp. 829-856.
17. E z h i l a r a s i, M., V. K r i s h n a v e n i. An Evolutionary Multipath Energy-Efficient Routing Protocol (EMEER) for Network Lifetime Enhancement in Wireless Sensor Networks. – Soft Computing, 2019, pp. 8367-8377.
https://doi.org/10.1007/s00500-019-03928-1
18. N a g a r a j a n, M. S. K a r t h i k e y a n. A New Approach to Increase the Lifetime and Efficiency of Wireless Sensor Network. – In: Proc. of IEEE International Conference of Pattern Recognition, Informatics and Medical Engineering (PRIME’12), IEEE, 2012, pp. 231-235.

Received: 30.06.2021; Accepted: 10.12.2021