

Optimal Clustering-based Routing Protocol Using Self-Adaptive Multi-Objective TLBO For Wireless Sensor Network

Ali Sedighimanesh

Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran
ali.sedighimanesh@gmail.com

Hessam Zandhessami*

Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran.
Zandhessami@srbiau.ac.ir

Mahmood Alborzi

Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran.
Mahmood_alborzi@yahoo.com

Mohammadsadegh Khayyatian

Institute for science and technology studies, shahid Beheshti university, Tehran, Iran.
M_khayatian@sbu.ac.ir

Received: 28/Nov/2020

Revised: 27/Mar/2021

Accepted: 22/Apr/2021

Abstract

Wireless sensor networks consist of many fixed or mobile, non-rechargeable, low-cost, and low-consumption nodes. Energy consumption is one of the most important challenges due to the non-rechargeability or high cost of sensor nodes. Hence, it is of great importance to apply some methods to reduce the energy consumption of sensors. The use of clustering-based routing is a method that reduces the energy consumption of sensors. In the present article, the Self-Adaptive Multi-objective TLBO (SAMTLBO) algorithm is applied to select the optimal cluster headers. After this process, the sensors become the closest components to cluster headers and send the data to their cluster headers. Cluster headers receive, aggregate, and send data to the sink in multiple steps using the TLBO-TS hybrid algorithm that reduces the energy consumption of the cluster heads when sending data to the sink and, ultimately, an increase in the wireless sensor network's lifetime. The simulation results indicate that our proposed protocol (OCRP) show better performance by 35%, 17%, and 12% compared to ALSRP, CRPD, and COARP algorithms, respectively. Conclusion: Due to the limited energy of sensors, the use of meta-heuristic methods in clustering and routing improves network performance and increases the wireless sensor network's lifetime.

Keywords: Clustering-based Routing; Hybrid Algorithms; Energy Efficiency; Gateway Zone.

1- Introduction

Wireless sensor networks are of the growing technologies with various and broad applications in different fields such as industry, agriculture, military, etc[1]. A wireless sensor network is made of many sensor nodes distributed in the desired environment. In many wireless sensor networks applications, sensor nodes are distributed in a random manner and unplanned in the environment [2]. The sensors are in charge of sensing, collecting, and processing their surroundings' data and sending them to the center or the sink. Because of the limited energy of sensors, minimizing the energy consumption of sensor nodes is one of the main challenges in these networks. The application of clustering techniques is one of the most effective methods in reducing the energy consumption of sensor nodes [3],[4]. The sensor nodes send the collected data to the cluster head node by

employing the clustering approach; sending data to the sink is the task of the cluster heads. In general, the applied routing protocols are divided into two groups: flat and hierarchical. In flat protocols, the cluster heads transfer the data to the sink directly and in a single-step manner; this process causes head clusters far from the sink to consume a great deal of energy. However, in hierarchical protocols, the cluster heads send the data hierarchically, rather than directly, to other cluster headers until the data is sent to the sink. In this state, routing between the cluster headers for sending data to the sink is a challenging issue that led to creative and new methods [5]. In general, hierarchical protocols perform better than flat protocols to balance power consumption and extend the network's lifetime. In wireless sensor networks, there are three types of nodes: cluster head, member node, and sink node, each of which is in charge of a function in the wireless sensor network [6]. Member nodes sense environment data and transmit them to the cluster head through multiple time division multiple access (TDMA). Cluster heads receive data from

1. Optimal clustering-based routing protocol

* Corresponding Author

member nodes and send the collected data to the sink via single-hob or multi-hob. The selection of cluster headers can be performed separately by sensors or sink or maybe pre-implemented by the wireless network designer. In the present paper, the selection of cluster headers has been made by the sink because of enough energy and the capability to make multidimensional calculations [7],[8]. In the present study, the authors present a new innovative algorithm for the hierarchical optimal cooperative routing protocol (OCRP) that performs clustering of sensor nodes and the routing process between cluster heads in each round. The remaining structure of this article's contents is as follows: Section 2 is related to the previous works, the proposed protocol is discussed in Section 3, and results and conclusion will be addressed in Section 4 and Section 5, respectively.

2- Related Works

Due to the expansion of the application of sensors and wireless sensor networks in recent years and the limited and non-rechargeable energy of sensors, increasing the efficiency and lifetime of wireless sensor networks is one of the most important challenges. Hence, various protocols have been presented to solve these basic challenges. Three of these proposed algorithms will be evaluated in the following.

2-1- Application Specific Low Power Routing Protocol (ASLPR)

The ASLPR protocol [9] applies various parameters, such as distance from the base station, the distance between the cluster heads and the sensor node, and remaining energy, for selecting the cluster head. For this purpose, each node first picks a random number between 0 and 1. In the case that the random number assigned to a node is less than T_{ASLPR} in Eq. (1), this node is chosen as the cluster head.

$$T_{ASLPR} = \begin{cases} Z(n) & \text{if } E(n) \geq t_1 \times \frac{1}{N} \sum_{i=1}^N E(i) \\ 0 & \text{if } E(n) < t_1 \times \frac{1}{N} \sum_{i=1}^N E(i) \end{cases} \quad (1)$$

$$Z(n) = \alpha_1 T_1(n) + \alpha_2 T_2(n) + \alpha_3 T_3(n) + \alpha_4 T_4(n) \quad (2)$$

Where N implies the total number of live nodes in the current frequency, and $E(n)$ equals the remaining energy of node n . In Eq. (2), $T_1(n)$ is referred to the partial energy threshold of the nodes, and α_1 denotes the weight of this partial threshold. Moreover, $T_2(n)$ is the partial threshold for the distance between the base station and the nodes, as well as α_2 is referred to as the weight of this partial

threshold. Meanwhile, $T_3(n)$ is the partial threshold for the distance between the cluster head and the node, and α_3 implies the weight of this partial threshold. The partial threshold $T_4(n)$ implies the number of frequencies in which a node is the cluster head, and α_4 denotes the weight of this partial threshold.

2-2- Clustering Routing Protocol for Dynamic Network (CRPD)

CRPD [10] provides a clustering-based routing protocol for dynamic networks and consists of 4 steps: 1) Neighbor discovery, 2) cluster head selection and cluster formation, 3) path construction and data collection, and 4) re-clustering and rerouting. This algorithm's basic idea is to select a sensor node with more energy and a larger degree as a cluster head in charge of collecting data transmission in each round.

In this algorithm, four principles are followed to select a cluster head and create a cluster as follows:

1. Sensor nodes with the highest degree (highest number of neighbors) are selected.
2. The remaining energy of the sensor nodes (E_r) must be higher than the threshold energy $E_{\text{threshold}}$ ($E_{\text{threshold}} = 0.4 E_0$).
3. In the case that the E_r of the node with the highest degree is not higher compared to the $E_{\text{threshold}}$, another node with the highest degree among the neighbors and the E_r higher than $E_{\text{threshold}}$ is selected as the cluster head.
4. The cluster heads cannot be adjacent. Also, cluster head selection relies on a couple of factors: remaining energy (E_r) and the degree of sensor nodes.

According to the formed cluster, every single sensor node transfers the data to the cluster head, and since the nodes are aware of their and sink's positions, the cluster heads select a node closer to the sink to select the next step in sending data from their neighbors to the sink.

2-3- Cuckoo Optimization Algorithm- based Routing Protocol (COARP)

In COARP [11], measurements are made to determine cluster heads in a central control system named sink. The network model is a single-step model in which the cluster heads directly connect with the sink. In each round, the sink is aware of the network nodes' energy level and position, and each node senses and collects surrounding data and then processes and transfers the data to the cluster head in a data packet. The COARP clustering method consists of steps: 1) Setup, which includes determining the cluster head and creating the cluster, 2) Registration, which includes generating the schedule, and transferring the data. In COARP, cluster heads are precisely selected by the Cuckoo algorithm in the sink. Afterward, the process of creating the clusters and the registration step is performed. Each cluster head receives data

related to all member nodes in its cluster and then transfers the received data in one step to the central station in the form of a packet.

3- Proposed Protocol

The present article's objective is the reduction of the energy consumption of sensor nodes through clustering and routing in multistage communication. The proposed protocol in this paper consists of four steps: 1) determining the gateway zone, 2) selecting the cluster head, 3) clustering and 4) routing to send the collected data. The general algorithm of the proposed method is as follows:

OCRP Algorithm

```

1  Divide area to sensing and gateway
2  Select nodes in sensing area for clustering;
3  Clusterheads=TLBO Algorithm;
4  For i=1: number of nodes
5      If node_i is in sensing area && node_i is
        normal node;
6          Node_i joins to nearest clusterhead;
7          End_IF
8  End_For
9  Route=Tabu Search Algorithm;
10 For i=clusterheads
11     CH_i joins to route;
12 End_For
13 Route connect to nearest gateway node;
14 For i=gateway nodes
15     If node_i is awake
16         Node_i send data to Base Station;
17     End_If
18 End_For

```

3-1- Gateway Zone

The sensor nodes are specified in the distributed environment and the sink's position in a random manner. Then, 20% of the environment is determined, which is closer to the sink as the gateway zone. The sensor nodes placed in this section are considered as relay nodes, the role of which is to receive data from the cluster headers and send them directly to the sink node.

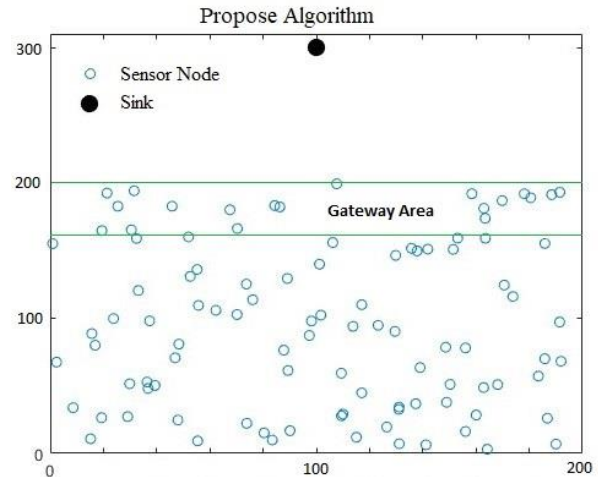


Fig. 1. Distribution of sensor nodes and determining the gateway zone

In this paper, first, a zone is considered as a gateway; the sensors within this zone are responsible for receiving data from the cluster headers and sending them to the sink. The purpose of this paper is the reduction of the energy consumption of sensors at the time of sending and inserting them in a specific cluster in multi-step communications. In this section, the authors will describe the proposed protocol in detail, which includes the following subsections: pseudocode of the proposed protocol, distribution of nodes, selection of cluster heads, cluster headers membership, and routing.

3-2- Selection of Cluster Heads and Clustering

After determining the gateway zone, the sensors outside the gateway zone should be clustered. In order to specify the headers from the existing sensor nodes, the method of probabilistic distribution in TLBO is applied. The TLBO algorithm is a population-based optimization and also a swarm intelligence algorithm inspired by the teaching-learning process in a classroom.

3-2-1- TLBO Algorithm

The TLBO algorithm consists of four phases, as follows:

- 1) **Initialization:** Creating the initial population in optimization algorithms is essential for obtaining the solution to a specified problem and is useful for optimal problem solving.
- 2) **Teaching phase:** In this section, the best learners are often picked as teachers; they transfer their knowledge to the learners to improve their knowledge level.
- 3) **Learning phase:** In this stage, learners communicate with each other for increasing their knowledge level.
- 4) **Reviewing after class:** Some learners enhance their effectiveness through reviewing the knowledge.

Learner initialization: Each learner possesses N decision variables where N implies the number of live sensors in that round. A random number in the interval [0,1] is assigned to each variable ($X_i = rand(0,1)$).

Teaching phase: Throughout this phase, the best learner in the population is considered as a teacher. Although, in the case that the best learner falls into the trap of local optimization, it is probable that other learners to not improve effectively. In the meantime, the use of the average of the total population is illogical, which may lead to a gradual evolution of the population. Accordingly, the average of the top learners, which includes the top six learners in the population, is applied instead of the average of the total population. In the present article, the top learners are selected by truncation to build the Gaussian model.

$$N(x_j, \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{1}{2} \left(\frac{x_j - \mu_j}{\sigma_j}\right)^2\right) \quad (3)$$

Two principle parameters are available for the high Gaussian distribution: standard deviation and mean. These parameters significantly affect the search performance. Due to the fact that the teacher must update his knowledge in a timely manner, a dynamic update mechanism is used for adjusting the Gaussian distribution. Standard deviation determines the search accuracy, and the mean controls the search direction. If the mean fails to reach the overall optimal region, the standard deviation is reduced to its minimum value, and the search will be slowed down. The standard deviation and mean are obtained from Eqs. (4) and (5).

$$\mu_j = (1-\alpha) \cdot \text{sup}\mu_j + \alpha \cdot (x_{\text{best}1,j} + x_{\text{best}2,j} - x_{\text{worst},j}) \quad (4)$$

$$\sigma_j = (1-\alpha) \cdot \text{sup}\sigma_j + \alpha \cdot \text{who}\sigma_j \quad (5)$$

Where $\text{sup}\sigma_j$ and $\text{sup}\mu_j$ imply the standard deviation and mean of better and more effective solutions, respectively, $\text{who}\sigma_j$ implies the average of the total population. $x_{\text{best}1}$, $x_{\text{best}2}$, x_{worst} are the best, second best, and worst learners in the population. The sign α implies the learning rate applied for controlling the update rate of Gaussian distribution $\alpha \in (0,1)$.

For the purpose of avoiding the falling in the trap of local optimization, the teacher uses crossover learning for transferring knowledge to the learner; in other words, if $r_i < 0.5$ for each dimension of a new learner, the current dimension is developed using the Gaussian distribution, otherwise equals the related dimension of the former learner. $r_i \in (0,1)$ (0,1) is a random number that follows the uniform distribution. According to the mentioned scheme, the teaching phase is summarized as follows:

Teaching Phase OCRP Algorithm

1. Initialize learners;
 2. Evaluate learners;
 3. M=6;
 4. Beta=0.1;
 5. While (stopping condition is not met)
-

6. Select M superior learners;
 7. SupMu=mean of superior learners;
 8. SupSigma=standard deviation of superior learners;
 9. Mean=mean of all population;
 10. Xbest1=best learner;
 11. Xbest2=second best learner;
 12. Xworst=worst learner;
 13. For i=all learners
 14. For j=all dimension
 15. If rand<0.5
 16. Mui=(1-Alpha)
 - *SupMu+Alpha*(Xbest1+Xbest2-Xworst);
 17. Sigmai=(1-Alpha)*SupMu+Alpha+Mean;
 18. Xnewi(j)= sample from the Gaussian distribution;
 19. Else
 20. Xnewi(j)=Xi(j);
 21. End_If
 22. End_For
 23. Evaluate new learners;
 24. If new learner is better than existing
 25. Xi=Xnewi;
 26. End_If
 27. End_For
-

Learning phase: In this phase, the learner should be able to discuss not only with classmates but also with the best learner and improve their knowledge according to their current information. This phase consists of two parts: local learning and permutation-based crossover learning, the details of which are presented as follows:

In local learning, local direction and data are integrated into the new learner. According to Eqs. (6) and (7), $T_{x_{\text{new}}}$ temporary learners are developed by a linear combination of x_{rnd} , x_{best} , and a random vector; x_{rnd} implies a random learner picked from a population that is dissimilar to x_i . r_1 , r_2 , and r_3 are random numbers in the interval [0,1]. The coefficient β is a randomized parameter equal to $\beta=0.1$. x_i and x_{rnd} determine the direction of learning, and β . ($r_3 - 0.5$) specifies the local search domain.

$$T_{x_{\text{new}}} = x_i + r_1 \cdot (x_i - x_{\text{rnd}}) + r_2 \cdot (x_{\text{best}} - x_i) + \beta \cdot (r_3 - 0.5), \text{ if } f(x_i) < f(x_{\text{rnd}}) \quad (6)$$

$$T_{x_{\text{new}}} = x_i + r_1 \cdot (x_{\text{rnd}} - x_i) + r_2 \cdot (x_{\text{best}} - x_i) + \beta \cdot (r_3 - 0.5), \text{ if } f(x_i) \geq f(x_{\text{rnd}}) \quad (7)$$

Learners only communicate with their classmates during the learning phase. Obviously, learners disregard the knowledge exchange that would lead to the loss of historical data. Accordingly, the differential evolution's double crossover is applied to save some historical data in each learner. In Eq. (8), the crossover function is described where C_r denotes the crossover rate in the interval [0,1]. $j_{\text{rnd}} \in \{1, 2, \dots, n\}$ is a randomly selected index which ensures that x_{new} receives at least one dimension from $T_{x_{\text{new}}}$.

$$x_{new,j} = \begin{cases} Tx_{new,j} & \text{if}(r_i \leq Cr \text{ or } j = j_{rnd}) \\ x_{old,j} & \text{otherwise} \end{cases} \quad (8)$$

In the above phase, learners exchange knowledge only between continuous vectors. A two-point (TP) crossover operator is randomly picked from two previous learners x_{r1} , x_{r2} , and a population that is different from the current learner. Afterward, a new pair of learners $x1_{new}$ and $x2_{new}$ are created after the crossover operation. If the better learner from $x1_{new}$ and $x2_{new}$ shows better performance than the current learner, the better learner will be replaced by the current learner.

Learning phase OCRP Algorithm

1. For i=all learners
 2. Randomly select another learner which is different from i (Xk);
 3. Generate a temporary learner;
 4. If Xi is better than Xk
 5. $Tx_{new}=Xi+rand*(Xi-Xk)$
+ $rand*(X_{best}-Xi)$ + $Betha*(rand-0.5)$;
 6. Else
 7. $Tx_{new}=Xi+rand*(Xk-Xi)$
+ $rand*(X_{best}-Xi)$ + $Betha*(rand-0.5)$;
 8. End_If
 9. For j=all dimension
 10. If $rand \leq Cr$ OR $j=J_{rnd}$
 11. $X_{newi}(j)=Tx_{new}(j)$;
 12. Else
 13. $X_{newi}(j)=Xi(j)$;
 14. End_If
 15. End_For
 16. If new learner is better than existing
 17. $Xi=X_{newi}$;
 18. End_If
 19. End_For
 20. For i=all new learners
 21. Randomly select two learners x1 and x2 from the population that are different from Xi;
 22. $x1_{new}=TP(x1, Xi)$;
 23. $x2_{new}=Tp(x2, Xi)$;
 24. If $x1_{new}$ is better than $x2_{new}$
 25. If $x1_{new}$ is better than Xi
 26. $Xi=x1_{new}$;
 27. End_If
 28. Else
 29. If $x2_{new}$ is better than Xi
 30. $Xi=x2_{new}$;
 31. End_If
 32. End_If
 33. End_For
-

3-3- Routing

The cluster heads receive and collect data from the member sensor nodes of their cluster; then, the data must be sent to the sink. For this purpose, hierarchical routing should

be applied so that the cluster heads send data to the sink in a multistage manner through creating an optimal path by the combination of TLBO and TS algorithms. According to the quantity of cluster heads in the network environment, initially, the TLBO algorithm is applied, and a response is obtained, then the response is optimized by the TS algorithm, and the cluster heads determine the optimal path for sending data. In this method, the authors first obtain two populations with 50 members, calculate, and sort the costs of each learner, and then select the least costly learners based on the number of learning members in the population. Each member of the population has a Cost and Position variables equal to the number of cluster heads minus one ($N_{Ch}-1$). The first population is created randomly, and the second population is obtained using Eq. (9).

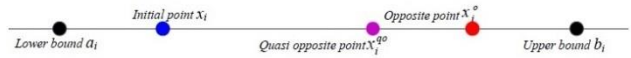


Fig. 2. Opposition-based learning and quasi-oppositional learning

$$x_i^o = a_i + b_i - x_i \quad (9)$$

$$x_i^{qo} = \frac{a_i+b_i}{2} + rand. \left(x_i^o - \frac{a_i+b_i}{2} \right) \quad (10)$$

Teaching phase: The update mechanism during the teaching phase is described as follows:

$$newX_i = X_i + rand. (Teacher - TF. Meam) \quad (11)$$

$$Mean = \frac{1}{NP} \sum_{i=1}^{NP} X_i \quad (12)$$

Where, $newX_i$ implies the new position of the learner, X_i denotes the n th learner, $Teacher$ implies the teacher with the best fitting, NP is referred to the quantity of learners in the population, and TF is the teaching factor which determines the size of the average to be altered. $rand$ denotes a random vector, and its element is a random number in the interval $[0,1]$.

Learning phase: The update mechanism for the i th learner during the learning phase is expressed as follows:

$$newX_i = \begin{cases} X_i + rand. (X_i - X_k) & \text{if } f(X_i) < f(X_k) \\ X_i + rand. (X_i - X_k) & \text{otherwise} \end{cases} \quad (13)$$

Where $newX_i$ denotes the new position of the i th learner, X_k is the randomly selected learner from the class. The learner fitting values of X_i and X_k are represented by $f(X_i)$, and $f(X_k)$, respectively, and $rand$ is a random vector in the interval $[0,1]$.

After completing the TLBO phases in each iteration, the population with the best cost is selected and given to the TS algorithm. This algorithm is implemented to the specified number of iterations and optimizes the desired response. In order to calculate the cost, the Prüfer algorithm is applied to form a tree between the nodes of the cluster head, and its cost is calculated.

The TS algorithm receives a solution, and actions, including Swap, Reversion, and Insertion, are applied to

the solution variables. According to the considered actions, all states of actions related to this operation is created in a list called Action List. These actions are applied to the obtained solution, and the cost and position are updated for each action. If the cost is lower, it replaces the best solution, and the action is placed on the Tabu List until a certain number of rounds are performed. The desired number of actions is calculated using Eq. (14) as follows:

$$N_{\text{Action}} = N_{\text{Swap}} + N_{\text{Reversion}} + N_{\text{Insertion}} \quad (14)$$

$$N_{\text{Swap}} = n \times (n-1)/2$$

$$N_{\text{Reversion}} = n \times (n-1)/2$$

$$N_{\text{Insertion}} = n \times n \quad (n = \text{number of position variables})$$

$$\text{Tabu List} = \text{Round}(0.5 \times N_{\text{Action}})$$

In order to calculate the cost of each solution, first, it is converted to the corresponding tree using the Prüfer algorithm, then routing is performed according to the obtained tree, and the cost is calculated from Eq. (15) where E_1 denotes the grid energy before application and E_2 implies the energy calculated after applying the routing operation.

$$\text{Cost} = E_1 - E_2 \quad (15)$$

This process continues until obtaining the best solution. Finally, the solution is given to the Prüfer algorithm with an optimal tree as its output according to which the routing is performed. If the maximum iteration of the algorithms in routing is considered as M , the number of iterations of the TS algorithm is equal to $M \times M$ since TS is entirely implemented for each iteration of the TLBO algorithm.

3-4- Network Operations and Calculation of Energy Consumption

In the proposed algorithm, network operations are divided into setup and stability phases. Each node's energy consumption in each round is calculated by evaluating what happened in both stages.

3-4-1- Setup Phase

Sink employs the k_{CP} control packet to communicate with sensor nodes. These control packets contain short messages that request position, energy level, and ID from all sensor nodes. For the purpose of receiving control packets from the sink, as in Eq. (16), the energy $E_{Rx}(k_{CP})$ is consumed. Moreover, all nodes also use the energy $E_{Tx}(k_{CP}, d)$ to transfer control packets, comprising of data around their positions, energy levels, and IDs, to the sink.

$$E_{Rx}(k) = kE_{\text{elect}} \quad (16)$$

$$E_{Tx}(k, d) = \begin{cases} KE_{\text{elect}} + \epsilon_{mp}Kd^4, & \text{if } d > d_0 \\ KE_{\text{elect}} + \epsilon_{fs}Kd^2, & \text{if } d < d_0 \end{cases} \quad (17)$$

Where $d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$ depends on the threshold distance, amplifier energy, ϵ_{mp} or ϵ_{fs} , receiver distance, and the allowable rate of bit error. According to the proposed algorithm, the sink processes the control packets and specifies which nodes become the cluster head, and each node becomes the member of which cluster head. Additionally, all nodes also use $E_{Rx}(k_{CP})$ energy for receiving their status data from the sink (whether members or CH). The consumed energy of all cluster heads to send TDMA schedules to members is presented in Eq. (18) as follows:

$$E_{Tx(\text{ch}_i)}(K_{CP}, d_{i-\text{tomem}}) = \sum_{i=1} \text{ch}_i * \begin{cases} K_{CP}E_{\text{elect}} + \epsilon_{mp}K_{CP}d_{i-\text{tomem}}^4, & \text{if } d < d_0 \\ K_{CP}E_{\text{elect}} + \epsilon_{fs}K_{CP}d_{i-\text{tomem}}^2, & \text{if } d > d_0 \end{cases} \quad (18)$$

The members consume energy for receiving TDMA schedules from the cluster head, which is calculated using Eq. (17).

3-4-2- Stability Phase

In a steady-state, active nodes send k -bit data to their cluster head according to the TDMA schedule received from the sink. The cluster head is all the time ready to receive this sensed data from its members and processes and collects all the data received from its members before transferring to the sink. The consumed energy by the transmitter sensor of the cluster head, E_{DA} , is calculated by Eq. (17).

$$E_{DA(m_i+1)}(k) = KE_{DA} * (\sum_{i=1} m_i + 1) \quad (19)$$

The dissipated energy for transferring sensed data to the cluster head is calculated using Eq. (20) as follows:

$$E_{Rx(m_i)}(k) = \sum_{i=1} m_i KE_{\text{elec}} \quad (20)$$

Where m_i implies the member nodes in the series $i = 1, 2, 3, \dots, n - L$. The signs L and n are the total number of cluster heads and sensor nodes, respectively. Consumed energy by the cluster head for collecting the sensed data from itself and members is calculated by Eq. (19) as follows:

Figure (3) demonstrates a simulation view of the proposed protocol when being simulated.

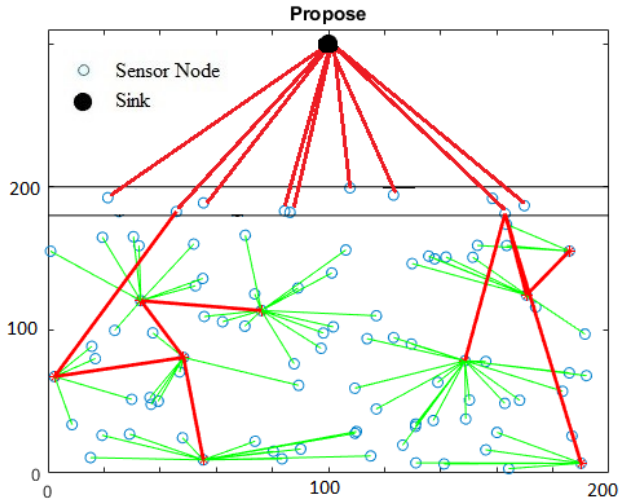


Fig. 3. A simulation view of the proposed method in MATLAB software

4- Results and Discussions

All algorithm simulations have been performed in MATLAB R2019b. For proving the strength of the proposed protocol in various scenarios, it is compared with known protocols such as ALSPR, CRPD, and COARP in terms of First Node Dead (FND), Half Node Dead (HND), Last Node Dead (LND), and the total number of data packets received in the sink from the beginning of the simulation to end of the network lifetime.

Assumptions

In the proposed protocol, the important assumptions of the network and radio models are given as the following:

- ❖ The sink is a rich source and is located in a fixed position.
- ❖ All sensors are fixed after distribution to the environment, and the energy of all sensors is identical at the beginning of the process.
- ❖ All sensors possess global positioning systems or other positioning devices connected to them.
- ❖ The desired communication channel is assumed to be symmetric.

Table 1. Adjusting the parameters of the TLBO algorithm

Parameter	Value
Population or Learner	50
Number of iterations	100
Number of Variables	length (Alive Nodes)
Variables Lower Bound	VarMin=0
Variables Upper Bound	VarMax=1

Table 2. Adjusting the parameters of the TS algorithm

Parameter	Value
Population or Solution	1
Number of iterations	100
Number of Variables	Nch-1 (Nch= Number of

	Cluster Head
Variables Lower Bound	VarMin= 0
Variables Upper Bound	VarMax=1
N_{Action}	$N_{Swap} + N_{Reversion} + N_{Insertion}$
$N_{Swap} = N_{Reversion}$	$N \times (N-1)/2$
$N_{Insertion}$	$N \times N$ (N=Number of position variables)

Table 3. Simulation parameters

Parameter	Value
Initial energy of the nodes	1j
\square_{fs}	10 (pj/bit/m ²)
\square_{mp}	0.0013 (pj/bit/m ⁴)
E_{elec}	50 (nJ/bit)
E_{da}	5 (nJ/bit)
Data packet size	4100 (bit)

Table 4. Used scenarios

Number	Number of sensors	Network size	Sink location
1	200	200 m × 200 m	(100 m, 250 m)
2	200	400 m × 400 m	(200 m, 450 m)
3	200	600 m × 600 m	(300 m, 650 m)
4	300	200 m × 200 m	(100 m, 250 m)
5	300	400 m × 400 m	(200 m, 450 m)
6	300	600 m × 600 m	(300 m, 650 m)

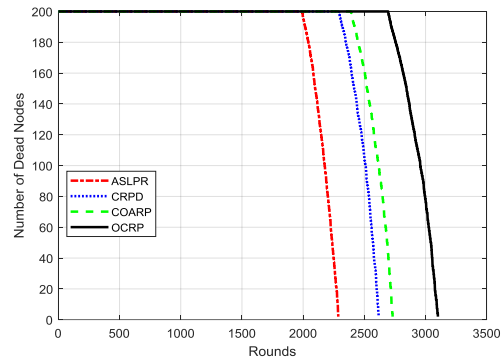


Fig. 4. Number of dead nodes in each round in the first scenario.

Based on the results obtained in Figure (4) in the first scenario, FND, HND, and LND in the proposed method have been increased and showed better performance compared to the ALSPR, CRPD, and COARP by 30%, 17%, and 12%, respectively, which indicates that in the proposed method, the energy of the sensors in each round has been consumed less than other methods. Figure (5) demonstrates the comparison of the network’s lifetime, which shows that the proposed method performed better than other methods in consuming the total network energy and spent on average less energy in each round for clustering and sending data.

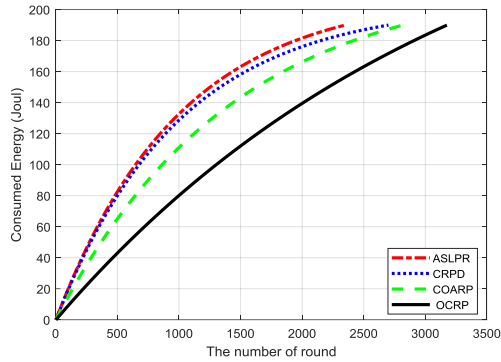


Fig. 5. Network energy consumption in each round in the first scenario.

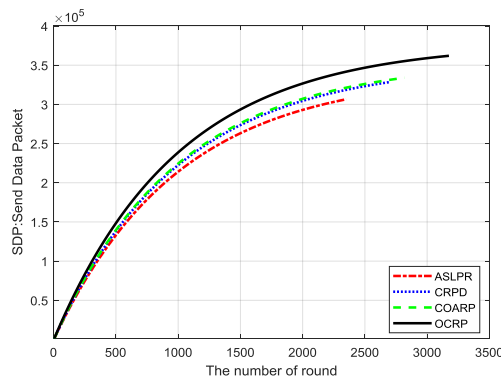


Fig. 6. Packets sent to the sink in each round in the first scenario.

Figure (6) indicates the packets sent to the sink in each round, which shows the increase in the number of packets to be sent to the sink in each round by the proposed method in comparison with other approaches. In the first, second, and third scenarios, the number of sensors distributed in the environment is constant; however, the size of the network environment and the sink's position is changed to evaluate the performance of the proposed method compared to other techniques. The results obtained from the second and third scenarios, as the first scenario, indicate that the proposed method has led to an increase in FND, HND, and LND compared to other methods. Moreover, the network's lifetime and the number of packets to be sent to the sink in the proposed method have been increased compared to other methods

Table 5. Comparison of FND, HND, and LND of the proposed method with other methods with 200 node sensors.

	Environment= 100 × 250 Sink= 100 × 250	FND	HND	LND	Environment= 200 × 400 Sink= 200 × 400	FND	HND	LND	Environment= 300 × 600 Sink= 300 × 600	FND	HND	LND
		ASLPR	CRPD	COARP		ASLPR	CRPD	COARP		ASLPR	CRPD	COARP
ASLPR		1992	2188	2349		397	630	1028		44	131	154
CRPD		2290	2516	2701		456	721	1180		49	148	175
COARP		2390	2625	2818		476	750	1222		52	155	182
My Propose (OCRP)		2691	2953	3175		538	845	1367		61	176	205

As can be seen in Table 5, our proposed method performs better in the death of the first node, HND and LND, in different scenarios than the ASLPR, CRPD, and COARP methods. Increasing the simulation environment and the number of sensor nodes is constant, causing the nodes to spend more energy to communicate with each other, and as a result, this increases energy consumption and reduces the life of the network, the results shown in Table 5 testify to This is a claim.

In the fourth, fifth, and sixth scenarios, the number of nodes is increased, and by increasing the number of sensor nodes, the same simulations are carried out. The network's lifetime and the number of packets to be sent to the sink are increased with an increase in the number of sensor nodes within the network. According to Figure (7), FND, HND, and LND in the proposed method are higher than other methods due to the increase in the number of sensor nodes in the network environment in the fourth scenario. Figures (8) and (9) show the better efficiency of the proposed method with regard to the number of packets to be sent to the sink and the network's lifetime.

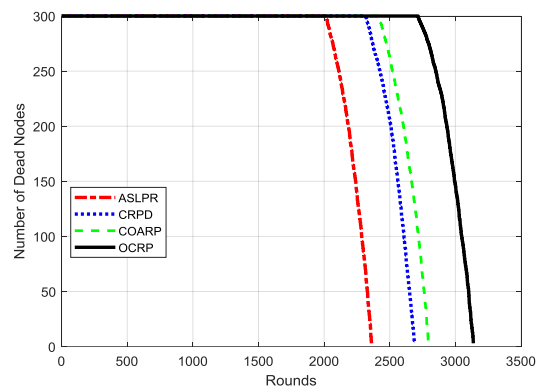


Fig. 7. Number of live nodes in each round in the fourth scenario.

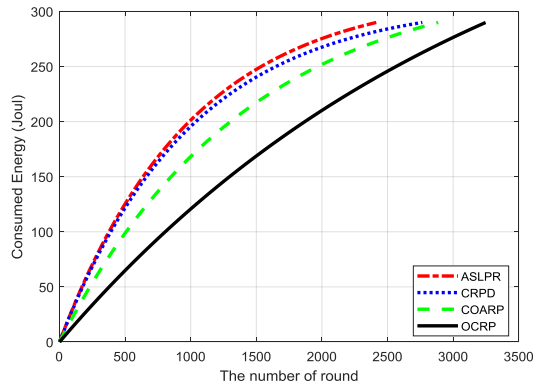


Fig. 8. Network energy consumption in each round in the fourth scenario.

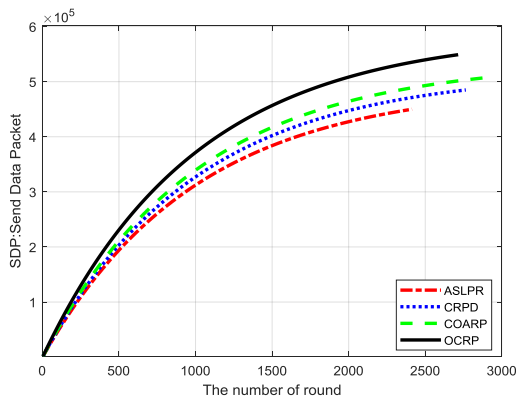


Fig. 9. The packets sent to the sink in each round in the fourth scenario.

As the simulation environment increases, the sensors must consume more energy to communicate with each other because more energy is required for transferring data as well as routing with the increase in the distance of sensor nodes. According to Table (6), the obtained results for FND, HND, and LND of the proposed method compared to other approaches are shown.

Table 6. Comparison of FND, HND, and LND of the proposed method with other methods with 300 node sensors.

	Environment= 100 × 250 Sink= 100 × 250	FND	HND	LND	Environment= 200 × 450 Sink= 200 × 450	FND	HND	LND	Environment= 300 × 650 Sink= 300 × 650	FND	HND	LND
ASLPR		2014	2210	2414		423	712	1105		48	137	158
CRPD		2315	2540	2765		486	818	1270		55	157	180
COARP		2409	2645	2886		507	854	1326		57	164	186
My Propose (OCRP)		2715	2975	3247		571	961	1491		66	185	210

By increasing the number of sensor nodes to 300 and simulating in 3 different environments, the proposed method has increased the network life compared to other methods. Therefore, according to Tables 5 and 6, we can conclude that in different cases, increasing the number of sensor nodes and changing the size of the simulation environment, the performance of the proposed method is better than other methods and has a longer lifetime.

5- Conclusion

In recent years, in wireless sensor networks, different protocols such as ALSPR, CRPD, and COARP have been proposed; the main purpose of each is to increase the network’s lifetime. The use of clustering and hierarchical routing leads to the reduction of the sensors’ energy consumption and, ultimately, the increase in the wireless sensor network’s lifetime. In the present article, the selection of the optimal cluster heads from the existing nodes is performed, and the sensors are clustered using the novel approach called Gaussian TLBO; then the multistage routing and the TS_ TLBO hybrid algorithm are applied to transfer the data collected by the cluster heads so that the data is sent to the gateway nodes and finally to the sink. According to the simulation results, the proposed algorithm reduces sensor nodes’ energy consumption because of the proper selection of cluster heads from the available sensor nodes. Also, due to hierarchical routing and the use of gateway nodes, the energy of cluster heads needed to transfer data to the sink is decreased, which leads to an increase in the network’s lifetime. The proposed OCRP algorithm reduces the energy consumption compared to the ALSPR, CPRD, and COARP algorithms by approximately 35%, 17%, and 12%, respectively. As future work, the following suggestions can be made to improve and expand the proposed method:

- 1) The use of the sleep/wakeup technique for gateway nodes to reduce their energy consumption.
- 2) Making the sink a stimulant to collect data

- 3) The use of reinforcement learning algorithms for selecting the cluster head.

References

- [1] N. R. Roy and P. Chandra, "Energy dissipation model for wireless sensor networks: a survey," *Int. J. Inf. Technol.*, vol. 12, no. 4, pp. 1343–1353, 2020.
- [2] M. Sedighimanesh* and H. Z. H. and A. Sedighimanesh, "Routing Algorithm based on Clustering for Increasing the Lifetime of Sensor Networks by Using Meta-Heuristic Bee Algorithms," *International Journal of Sensors, Wireless Communications and Control*, vol. 10, no. 1, pp. 25–36, 2020.
- [3] A. Belfkih, C. Duvallet, and B. Sadeg, "A survey on wireless sensor network databases," *Wirel. Networks*, vol. 25, no. 8, pp. 4921–4946, 2019.
- [4] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, "Clustering objectives in wireless sensor networks: A survey and research direction analysis," *Comput. Networks*, vol. 180, p. 107376, 2020.
- [5] V. Parashar, B. Mishra, and G. S. Tomar, "Energy Aware Communication in Wireless Sensor Network: A Survey," *Mater. Today Proc.*, vol. 29, pp. 512–523, 2020.
- [6] D. K. Sah and T. Amgoth, "Renewable energy harvesting schemes in wireless sensor networks: A Survey," *Inf. Fusion*, vol. 63, pp. 223–247, 2020.
- [7] M. Sedighimanesh* and H. Z. and A. Sedighimanesh, "Presenting the Hybrid Algorithm of Honeybee - Harmony in Clustering and Routing of Wireless Sensor Networks," *International Journal of Sensors, Wireless Communications and Control*, vol. 9, no. 3, pp. 357–371, 2019.
- [8] A. Kochhar, P. Kaur, P. Singh, and S. Sharma, "Protocols for wireless sensor networks: A survey," *Journal of Telecommunications and Information Technology*. 2018.
- [9] M. Shokouhifar and A. Jalali, "A new evolutionary based application specific routing protocol for clustered wireless sensor networks," *AEU - Int. J. Electron. Commun.*, vol. 69, no. 1, pp. 432–441, Jan. 2015.
- [10] S. Wang, J. Yu, M. Atiquzzaman, H. Chen, and L. Ni, "CRPD: a novel clustering routing protocol for dynamic wireless sensor networks," *Pers. Ubiquitous Comput.*, vol. 22, no. 3, pp. 545–559, 2018.
- [11] M. Khabiri and A. Ghaffari, "Energy-Aware Clustering-Based Routing in Wireless Sensor Networks Using Cuckoo Optimization Algorithm," *Wirel. Pers. Commun.*, vol. 98, no. 3, pp. 2473–2495, 2018.

Ali Sedighimanesh received the B.S. degree in Information technology from Islamic Azad University, Parand Branch, Iran in 2009, and M.S. degree in Information technology from Islamic Azad University, Qazvin Branch, Iran, in 2013. Currently, he is Ph.D. Candidate in Islamic Azad University, Tehran Branch, Iran. His research interests include wireless sensor networks, IoT (internet of things), Blockchain, Web mining, and Data mining.

Hessam Zandhessami is an assistant professor of Information technology at Azad University, Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran. His research is focused on, IOT (internet of things), Web mining and Methodologies in Software Engineering, Management.

Mahmood Alborzi is a full-time Assistant-Professor of Information technology at Azad University, Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran. He received his Ph.D. degree PhD in Artificial Neural Networks Brunel University, UK, England in 1996. He is conducting research activities in the areas of Image and Data Analysis Software Systems, Artificial Neural Networks, data mining.

Khayyatian mohammadsadegh is a full-time Assistant-Professor of technology management at Shahid Beheshti University, Tehran, Iran. He received his Ph.D. degree PhD in technology management Allameh Tabatabaei University. He is conducting research activities in the areas of Image and Data Analysis Software Systems, Artificial Neural Networks, data mining, technology management.