Journal of Computer and Knowledge Engineering, Vol. 2, No. 1, 2019.
DOI: 10.22067/cke.v2i1.80831

A Heuristic Method for Node Management in Large-Scale Wireless Sensor Network Considering Connectivity and Energy Consumption

Rahil Dehesh[1] Mahdi Mehrabi[2*]

Abstract: In a large-scale wireless sensor network (WSN), the nodes are usually randomly spread. In such a network, more active nodes consume more energy and shorten the network lifetime. An appropriate approach is nodes management, mainly turning off or deactivating some redundant nodes during some periods of time. The redundant nodes are those that deactivating them does not affect the overall objective operation of the network such as full connectivity and coverage. In this paper two methods for detecting redundant nodes in large-scale WSNs are presented. The proposed methods can detect more redundant nodes, especially lateral redundant nodes, based on heuristic graph theories in the network graph. The simulation results indicated that the proposed methods performs well both in dense and non-dense WSNs and reduces the overall energy consumption of the network better than the previous method.
Keywords: wireless sensor network, large-scale wireless sensor network, node management, network connectivity, network coverage, energy consumption

1. Introduction

A Wireless Sensor Network (WSN) consists of a large number of small sensor nodes which communicate over wireless media. The sensors have the ability to sense, based on the type of sensors, process and send its information (directly or through each other) to the sink which cause the appearance of an idea to create and spread a network known as a WSN. WSN can be applied in agriculture, medical, and industrial, and military cases as well as monitoring residences, moving target tracking, fire detection, traffic monitoring, and so forth [1& 2]. On the other hand, recent developments in integrated circuit manufacturing technology and the development of wireless communication technology have made the strong background for designing large-scale WSN [3].

In a WSN, as a wireless network, there are some fundamental concepts like connectivity, coverage, and energy consumption [4]. Connectivity means all nodes in a spread network must be accessible mainly by other nodes. Coverage means that any point in the intended area is included in the sensing range of at least one sensor. In many cases, it is impractical to place sensors at predetermined locations, and placing them randomly is preferred, thus coverage becomes more important. One way of examining coverage is r-strip. R-strip means each sensor should be placed at the distance r from its neighbors, with r being the sensing area. Bai et al. [5] offered the pattern for optimal deployment of sensors in terms of the number of required sensors to achieve full coverage.

To formulate the coverage in a WSN; where a region is covered by several sensors, a concept called k-coverage is introduced, which *k* is the number of sensors that see anywhere in the intended area. Cardei and Wu [6] offered a plan in which a node performed k-coverage in its environment, and if it found k nodes that covered its intended area, it put itself in sleep mode and then checked again after a random period of time. He et al. [7] presented another algorithm to check the network connection and coverage. In their algorithm, the network was divided into different parts which had the advantage of both synchronous and asynchronous networks. In a synchronous network, sensors need to cooperate with each other for connectivity and coverage, while in an asynchronous network, each sensor works independently. Rebai et al. [8] proposed a linear programming model to improve the coverage of the desired area with the minimum number of sensors and to maintain a connection between them. They used a local search to find non essentials censors and a genetic algorithm that minimized the total number of deployed sensors, while the network was covered with at least one sensor. The linear model was used to provide an optimal solution .

Energy consumption of nodes is one of the most important challenges in the WSN. The sensors work with small batteries which are difficult to replace. When the number of sensors increases in the network, they build a large-scale WSN. The large number of active nodes in the network consume a great amount of energy and the lifetime of the network becomes shorter. So it needs efficient ways to reduce energy consumption in order to increase the lifetime of the network. In this regard, Lin et al. [9] proposed a clustering protocol to partition a large-scale network and Based on this clustering scheme, different energy saving methods were proposed that could efficiently save energy. Another approach is reducing the required active nodes while maintaining the proper function of the large-scale network so that we have the overall objectives of a network such as full connectivity and coverage [10].

The main contribution of this paper is presenting a method, based on heuristic graph theories to reduce active nodes in a large-scale WSN while preserving connectivity and coverage in order to increase overall lifetime of the network. Other parts of the paper is organized as follows: Section 2 reviews the power control techniques, topology control and background of the WSN. In section 3, a method is presented for selecting sensors in WSN. Section 4 presents

Manuscript received May, 25, 2018; accepted August, 31, 2019.
[1]M.S. Department of computer engineering, Shiraz branch, Islamic Azad University, Shiraz, Iran.
[2*]Assistant Professor, Department of computer engineering, Shiraz branch, Islamic Azad University, Shiraz, Iran.
Email: mehrabi@iaushiraz.ac.ir

the simulation of the proposed method comparing to existing methods. And conclusion is presented in section 5.

2. Related works

Because of the large number of sensors and the cost of access to them in a large-scale WSN, replacing and recharging their batteries are not straightforward, so energy efficiency is very important in these networks. For this reason, several methods have been offered to manage energy consumption of the overall network which increases the lifetime of the network. A simple method to accomplish this is adjusting the transmission power of the nodes. If the transmission range is large, each node can communicate with further neighbors, but this increases the energy consumption over time. A proper transmission power of the nodes must be set to reduce energy consumption of the network. Li et al. [11] say that each node could transfer with minimum required power such that in each cone with degree α there is at least one neighbor receiving the node. They indicated if $\alpha \leq 5\pi/6$ the network connectivity was maintained but if $\alpha > 5\pi/6$ the connection may not be maintained. In their algorithm, at first each node broadcasted the message using low transmission power and received an ACK. In the next step, they increased the transmission power to find more neighbors. If in the cone with center node u and degree α around it, there was no node or there was a Gap, then the node u broadcasted with more power. This work continued until node u doesn't find any α-Gap or broadcasted with maximum power. By receiving the message from node u, node v answered with an ACK message. With the reception of the ACK from node v, node u added v into its neighbors set N(u) and the direction of v into direction set D(u). They improved their algorithm and they also checked out the mechanism named Node Discovery Protocol (NDP) that could recognize network changes such as node movement or add new nodes to the network. NDP is a simple monitoring protocol for each node which contains sending nodes ID and monitoring transmission power to find the neighbors of each node. A node u must broadcast with enough power and be able to communicate with all its neighbors. In that reconfiguration algorithm, they preserved connection and also they guaranteed that each cone with degree α around node u was covered. The weakness of their approach was that the initial power P or the increasing power in each step was not reviewed.

Misra et al. [12] introduced the RNPC algorithm in which relay nodes were placed in a subset of specific locations to ensure network connectivity for nodes with constant and predefined transmission power. They assumed that all sensor nodes have communication range $r>0$ and all relay nodes have the transmission range $R\geq r$ that r and R are the given constants. They created a connected hybrid communication graph consisting of sensors, relay nodes, and base stations. Then, non-negative weights were assigned to the edges, so that the weight of an edge was equal to the number of relay nodes incident to it. Thus a complete graph with weighted edges was made. After all, they expanded the minimum spanning tree of the graph to obtain the location of relay nodes.

Since each node in the large-scale WSN could have different transmission power, an approach to reduce power consumption is to find the minimum of required power for each node. In this regard, Teng et al. [13] proposed the adaptive transmission range based on topology control. Because of the expansion of transmission range, nodes in the network form multiple paths for data collection to the sink node. By extending the transmission range, data packets could be transmitted to the sink with fewer hops. Thereby the delay of data collection was reduced and the reliability of data transmission was improved. As extending the transmission range consumed more energy, they found the imbalanced energy consumption of the network. Their approach reduced the maximum energy consumption, but had some limitations. Clustering the nodes in a large-scale WSN can be also considered as a solution to reduce energy consumption in a network. The sensors were grouped in a cluster series and a central node was selected as cluster head. The task of head is managing other nodes in its cluster. However, selecting the appropriate cluster could affect the amount of energy consumption in the clustered networks. In a method presented by Xu et al. [14], a clustered network with coverage and connectivity was introduced for large-scale and high-density WSN. Furthermore, their algorithm could adjust the number of active nodes adaptively according to the required coverage to balance the energy consumption. Hence, Leu et al. [15] offered an energy aware clustering method that selected a cluster head based on remaining energy of each sensor and the average regional energy of all sensors in each cluster. Their work indicated significant improvement of energy consumption compared to random cluster head selection.

Some works have been done to solve the power constraint problem and energy limitation in WSNs. Minimizing energy consumption is an important consideration in the design of energy constrained WSNs. The energy consumption approaches reduce the average energy consumption of network by using techniques such as reducing the number of data packets, clustering, and cooperative communication. In this regard, energy efficient deployment strategies for heterogeneous and relay networks and cooperative communication were discussed [16]. Garcia et al. [17] organized sensors in cooperative groups to reduce the number of the messages transmitted inside a WSN. Their cooperative group-based network could reduce the overall energy consumption of the WSN. The limitation was that they considered group-based WSN with fixed sensors where mobile sensors move only inside the boundaries of the groups. On the other hand, sensors in different positions could have different numbers of packets to handle, resulting in unbalanced energy consumption. To efficiently use the residual energy, Chen et al. [18] adopted a high data transmission power in non-hotspot areas to achieve a higher reliability at the cost of large energy consumption. Relatively low transmission power was adopted in hotspot areas to longer network lifetime. Although building reliability cooperative communications network was the key issue for WSNs, but required new techniques to improve communication reliability as well as lifetime for harvesting energy especially in practical applications.

Optimal selection of paths for data transferring is another approach that saves energy in WSNs. Routing protocols in WSNs might differ depending on the application and network architecture. The sensor nodes communicate together by many wireless strategies and these communications strategies are administered by routing protocols. Performance of WSNs largely depends on the routing protocols, which are application based [19, 20]. Anisi et al. [21] reviewed the routing approaches based on their features. Most of the proposed data routing algorithms consider the energy of the nodes as a factor in proposing a data routing approaches in WSNs. Malathi et al. [22] proposed a cluster based hierarchical routing protocol considering energy. The protocol could minimize the overhead energy while clustering and improve the life time of the network. Amgoth et al. [23] proposed an energy aware routing algorithm for cluster based WSNs. The algorithm was based on a clever strategy of cluster head (CH) selection, residual energy of the CHs and the intra-cluster distance for cluster formation. Their proposed algorithm also balances energy consumption of the CHs during data routing process. In this regard, Brar et al. [24] proposed a directional transmission-based energy aware routing protocol. Their protocol could identify energy efficient optimal paths to reduce energy consumption in the network.

Another way to reduce the overall energy consumption in a large-scale WSN is decreasing active nodes. Redundant active nodes in a large-scale WSN waste the network resources. As there are a large number of nodes in a large-scale WSN which do a similar task, sometimes there is no need to activate all the nodes and may periodically disable some of them. It is worth noting that every attempt to manage nodes in a large-scale WSN should consider connectivity and coverage in the network. Yet, different methods are presented to deactivate or make them into sleep or even remove redundant nodes in a period of time. A plan to select nodes for increasing energy efficiency in the solar-powered WSN is presented by Horng et al. [25]. In their plan, sensors were classified into different parts and active sensors were selected based on the resulting classification. That plan resulted in the maximum monitoring area with lower energy consumption. Although they provided the node selection scheme of the cross-stream cases in their paper, not all cross-stream environments were suitable for their scheme.

On the other hand, when nodes in a network do a certain task such as tracking a moving target, all of them need not to be activated and some of them can be disabled including those which are not in the direction of the target, and in this way their energy is preserved. Kaplan [26] offered an algorithm to select active nodes based on the target position. When a target arrived, they used the situation of the target to determine relative position of required nodes to track it. Furthermore, he evaluated the algorithm to select a suitable active set by creating various configurations of network sensors and showed that the active nodes may be restricted to track a target, and this would reduce overall energy consumption and increase the lifetime of the network. But the measurement equation in that paper did not lead to a finite set of outcomes. In another research, Demigha et al. [27] checked out several number of recent target tracking algorithms with the aim of keeping the network energy. They expressed that only the subset of nodes near the target must be enabled and other nodes must remain inactive until they receive a straight activation message from previous tracking nodes, showing that the target is probably in their sense range. This may be done by target position prediction. Armaghani et al. [28] presented an active sensor selection plan to track different groups of targets. At first, they used the clustering method to cluster the network, and then they found the groups of targets that can be tracked by each cluster. The static WSN considered in their work was not resilient against unspecified or dynamic conditions, e.g., sensor fault and demise.

Formally, in the large-scale WSNs the simple solution for reducing energy consumption is periodically deactivating or turning off some redundant nodes that their absence do not affect major functionality of the network mainly connectivity and coverage. If the node is dead at the end of the period, then it sleeps and continues to charge energy until the amount of energy is usable. On the other hand, since different nodes may be candidates for deactivating, this issue could be considered from different perspectives. Wang et al. [29] checked out in their algorithm how the number of active nodes could be minimized to reduce energy consumption in a large-scale network with keeping connectivity and coverage. They selected the number of nodes to deactivate in each step of the algorithm, and then they checked out the network. If deactivating these nodes preserved the conditions of network connectivity and coverage, the nodes could be deactivated. Otherwise, the node would remain active.

3. The proposed method

In this section, the method is introduced. The proposed method detects redundant nodes in a large-scale WSN. Sensors in a large-scale network may be selected to be deactivated alternatively in a way that after removing them from the network, the connectivity and coverage they are two main parameters of a network are preserved. Most of previous methods considered a specific neighborhood around nodes or a cluster to distinguish redundant nodes or to make an energy optimization plan. Since detecting redundant nodes in a local region may restrict the detection, the proposed method forms the graph of all nodes in WSN, and then based on basic graph theories, detects the nodes which can be deactivated without affecting connectivity and coverage of the overall network. The network communicating graph is used to verify the communication issues in a network. Then, the graph traversal algorithm and minimum spanning tree are used to identify and select the best redundant nodes to be turned off without affecting the connectivity of whole network. This means that in identifying redundant nodes, the proposed method selects the nodes in which turning them off does not increase the overall internal communicating distance of the nodes in the network not in a local area. In the following, the proposed method would explained step by step.

3.1 The network graph

The first and main part of our approach is forming a graph of the network nodes that are spread in a surface area. The importance of such graph makes us use graph theories to find redundant nodes and check the connectivity of the network. A non-directed graph is used in this paper.

For this purpose, we consider each of the sensors in networks as a node or a vertex of a graph and the distance between nodes as edge weights in the graph. The edges are only considered nodes when the distance between them is lower than the maximum transmission range of a node. A graph G is consists of two sets V and E as V is a vertex set and E is an edge set of the graph, and graph G is indicated as G(V,E). Since a vertex in the graph corresponds to a node in the network, it is worth noting that in the following, nodes and vertices may be used interchangeably.

Here the function $Edg_j(i)$ is defined to describe adjunct of two sensor nodes i and j in equation (1).

$$Edg_j(i) = \begin{cases} 1 & if\ dist(i,j) \leq r = constant \\ 0 & \text{else} \end{cases} \quad (1)$$

In equation (1), dist(i,j) is the distance between two nodes i and j, R is transmission range of sensors. If the distance between two nodes is less than the transmission range of nodes say R, then $Edg_j(i)$ is equal to 1. That means that two nodes are adjacent and we will consider an edge between them which its weight is dist(i,j). We repeat it for all network nodes and obtain the edges. An example of such a graph without the links weight is depicted in Figure 1.

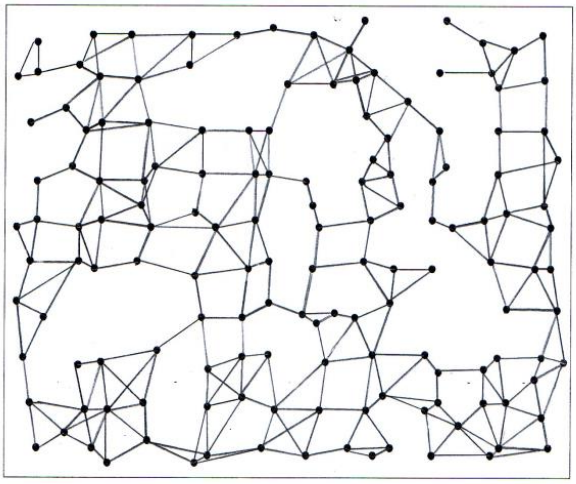

Fig. 1. An example of formed graph based on a sample WSN

3.2 Checking the network connectivity

As mentioned before, one of the most important concerns of the WSN is connectivity. In this section, we present how to check the connectivity of a network that its graph has formed. To verify this property, a common way is using graph-based theories. To have the full connectivity in a WSN, all vertices in corresponding graph should be connected. In the undirected graph G, the two vertices u and v are called connected if graph G has the path from u to v, otherwise, they are called non-connected. If two vertices can be connected to each other with a path of length 1 or one edge, the vertices are called adjacent or neighbor. The graph is connected if any two desired vertices can be connected to each other through at least one path. Here, to check the network connectivity, we use graph traversal algorithms. The purpose of graph traversal is to obtain all vertices that are available through one vertex. The idea is very simple: if the graph traversal algorithm indicated that at least one vertex in a WSN graph is not reachable by any other node, we can conclude that the network is not connected. Starting from a vertex using Breadth-First Search (BFS) or Depth-First Search (DFS) algorithm [30], all vertices that can be accessed are counted. After the graph has been entirely traversed, if the numbers of counted vertices are equal to or smaller than the number of vertices in graph G, the graph and thus the intended network are connected. Otherwise the network is non-connected. Here, we examined the network connectivity using BFS algorithm. This algorithm starts the traversal from root (in graphs without root, the desired vertex is selected as a root) and puts it at level 1, then in each step examines all the neighbors of the vertices which have been accessed in the last level, and have not been accessed before, and puts them to the next level. This process stopps when all the neighbors of the vertices in the last level have been seen. Thus the algorithm each time visits all neighbors of a vertex and then goes to the next vertex. This process continues until traversing all vertices in the graph.

3.3 Checking network coverage

As previously mentioned, another important concern of a sensor network, is network coverage. Here, we consider all sensors have the same sensing area. As a result, we divide the intended area by the flat square equal to a sensing area of a sensor node. If there is at least one node in each square, the coverage is achieved. We assume that sensors are densely deployed and the sensor nodes have overlapping sensor ranges so that every part of the sensing area is covered by at least one sensor.

3.4 Identifying redundant nodes

After obtaining the graph of the intended network which has was explained earlier, the redundant nodes should be determined. The redundant nodes can be turned off or put into sleep mode periodically to save energy and increase the network lifetime. The main issue is how and which nodes are selected as redundant nodes. Choosing the proper nodes as redundant can affect the lifetime of the network. In the following, the two proposed algorithms to determine the redundant nodes are explained. The first one has low complexity and can detect lateral redundant nodes in the non-dense networks. The other one is the improvement of the first method to cover dense and non-dense networks. Here by a bit more complexity, the second method could better detect redundant nodes in dense network.

3.4.1 The first method to identify redundant nodes

In this section, the first proposed method is explained. We use the network graph which is created in the previous sections. First, the minimum spanning tree (MST) should be defined in the target graph. MST is a subset of the edges of a weighted graph that connects all the vertices together without any cycles and with the minimum possible total edge weight [31]. Actually, MST defines the edges which form the best routes for nodes to communicate with each other. The main idea of the first proposed method is simple: since MST defines minimum edges which forms the best route for any communications between all vertices of the graph, each vertex (or node) which has no more than one edge belonging to the MST could be removed without affecting the connectivity of the other vertices in the graph, and so may considered as a redundant node.

To obtain the minimum spanning tree of the connected graph, different algorithms can be used. Two common algorithms to find minimum spanning tree are Prime and Kruskal algorithms [31]. We modified the Prime algorithm to meet a WSN graph the pseudo code of which is indicated in Figure 2. Starting from one vertex, the Prime algorithm selects edges with the lowest weight which passes from it. In the next step, the edge that has the minimum weight between edges that passes from two available nodes is selected. In the same manner, through next step, the edge that has the minimum weight between edges that passes from three available nodes is selected and this procedure is repeated until the minimum spanning tree is obtained.

After obtaining the network graph and recognizing the minimum spanning tree, the MST degree of each vertex is calculated by equation (2). The MST degree of vertex i says MSTdegr(i) is the number of edges connected to the vertex and belonging to the MST.

$$\text{MSTdegr}(i) = \#\left((\text{edges connected to } i) \cap \text{MST}\right) \quad (2)$$

Then, we assign a MST index say *MstIndx(i)* to each vertex i of the network nodes according to the equation (3).

$$\text{MstIndx}(i) = \begin{cases} 1 & if \ \text{MSTdegr}(i) == 1 \\ 0 & else \end{cases} \quad (3)$$

Finally, the nodes that have a MST index equal to 1 are redundant and are candidates to be turned off. In fact, MST index equal to 1 indicates that this node is not in communication route between any other two nodes, so its removal does not affect connectivity of the network. Beside connectivity consideration, we are looking to remove the nodes where turning off them do not affect the coverage. For this reason, the sensing area of the candidate node is checked. If there is any other node which covers the candidate node sensing area, the candidate node can be removed without affecting the coverage. The pseudo code of the proposed algorithm to select redundant nodes which could be removed including all steps is presented in Figure 3.

Due to MST property, the proposed method presented in this section selects redundant nodes which are not through the main communication route of the network. These nodes are mainly lateral nodes that exist in non-dense networks. To improve the proposed method and delete more redundant nodes from network, we merged our proposed method with a previous method to create a new method which has the advantage(s) of both methods. The second method also maintains the network connectivity and coverage which is presented in the next section.

Input: n equal to number of nodes in the network graph, W[i][j] is edges weight between node i and j in the network graph, Vnear: The index of vertex with the smallest distance.

Output **:** Edges set F

```
Start
 F = Φ; (F is  an edge set)
For node i=2 to n
nearest[i] =1;
distance[i] = W[1][i];
End   For
Repeat (n-1 times)
min = ∞;
For node  i=2 to  n
If   (0 ≤distance[i] ≤min) then
min = distance[i];
vnear = i;
end If
End For
Add edge (vnear , nearest[vnear]) To F;
distance [vnear] = -1;
For  i=2 to  n
If (W[i][vnear] ≤ distance[i])
distance[i] = W[i][vnear];
nearest[i] = vnear;
End If
End Repeat
Return F as MST;
```

Fig. 2. Pseudo code to obtain the MST of a network graph

```
Assume the number of nodes equal to N.
Obtain MST according to Figure 2
For i =1 to N
Calculate MSTdegr(i)  of each node i based on formula in (1);
If (MstIndx(i) = = 1) && ( there exists another node covering the node i sensing area)
Turn off node i;
End if
End for
```

Fig. 3. Pseudo code of the first node deletion (turning it off) algorithm

```
Input: n number of nodes or vertices in the network graph
For i=0 to n
For j=0 to n
If dist(i,j) ≤ r
Add j to NB(i);
 End   For
 End   For
For i=0 to  n
RD = {i};
For j=0 to n
If   NB(i) ==  NB(j)
Add node j to RD set;
Run BFS to calculate eccentricity of node j say eccen(j);
End If
End For
For node j in RD set
Calculate IF(j);      (based on equation (4))
End for
Extract two nodes with the maximum IF() in RD set say m and n;
Exclude node m or n which has the least eccentricity from RD set;
Turn off all nodes in RD set;
End For
Implement the first proposed algorithm in Figure 3 nd delete some other nodes;
```

Fig. 4. Pseudo code of the second node deletion algorithm.

3.4.2 The second method to remove nodes

In the previous section, a method was presented based on basic graph theories to remove nodes in large-scale WSN. In a dense network many intermediate redundant nodes exist through central area of the network. Since the first method can effectively detect lateral redundant nodes which are mainly exist in non-dense networks, in this section, another method which is the improvement of the first one is presented. Besides MST for detecting lateral redundant nodes, the second method relies on neighborhood area to detect intermediate redundant nodes, and furthermore, considers energy of the nodes. For this purpose, we identify the redundant neighbors' nodes. The idea is that the nodes with the same neighbors may have the same role in the network and may be considered redundant. As mentioned before, two nodes are adjacent or neighbor if the distance between them is less than the transmission range of the nodes. Then we compare the nodes with each other. All nodes which have the same set of neighbors are put in series named redundant nodes (RD). Here, the method presented by Wang et al. [29] is accepted to make some preferences to the redundant neighbor nodes. They considered a parameter which represents amount of nodes energy among its neighbors, relative to each of the members in a neighbor set. This parameter was named IF which its value obtains according to the equation (4).

$$IF(i) = \frac{Er_i}{|NB(i)-i|} \prod_{j \in NB(i)} L(i,j) \qquad (4)$$

In equation (4), i is the intended node and j specifies the neighbor of node i. Er_i is the remaining energy of node i.

NB(i) is the neighbors set of node i. L(i,j) expresses the link quality between two nodes i and j. п specifies the cardinality of the relationship. Yet in IF factor, the role of a node among the entire network is not considered. Here, in addition to the amount of IF, we calculate the amount of eccentricity [32] for each node in neighbor RD set. Therefore, to remove nodes in addition to the amount of IF for each node, the amount of eccentricity for each node is considered. Eccentricity is a graph based factor for a node which represents the node effect on network diameter. The higher the eccentricity, the longer the route imposed by the node in the network. The maximum amount of eccentricity is the diameter of the graph. As a result, among nodes in RD sets that have the highest IF, the node with less eccentricity would be kept on and the rest of the nodes could be removed or turned off in the same RD set. The Breadth-First Search (BFS) and graph traversal algorithms can be used to obtain the amount of eccentricity for each node. After selecting nodes via the above algorithms, we delete or turn them off to reduce energy consumption in network. In the second stage, after removing these nodes from network RD sets, we apply the minimum spanning tree on this graph, and based on method 1, specify the nodes with degree 1. If there are any other nodes in that region that maintain coverage condition, we will delete or turn off the node. The pseudo code of this algorithm is described in Figure 4.

4. Simulations and Results

In this section, the proposed methods to remove nodes in large-scale WSNs are applied to some networks and the results are compared. For this purpose, we use the NS2 simulator. Table 1 shows the simulation parameters.

Table 1. Simulation Parameters

Parameter	value
Area	100
Initial Energy	Max
NS2 Version	2.28

For simulation, first we built the network with different nodes. Then we executed three experiments. In each one, a method was applied to identify redundant nodes in the network. The methods in each experiment are the two proposed methods and also the previous redundant nodes selection by Wang et al. [29]. After identifying the redundant nodes, they are turned off periodically, and hence the network performance is evaluated by energy consumption. For evaluation of the managed network in which redundant nodes are turned off, the managed network works for a period of time and the energy of nodes and the whole energy of the network are checked. As a result, the performances of the two proposed methods in distinguishing redundant nodes are compared with the method presented by Wang et al. [29]. At the first stage, we form a network with 100 nodes which are randomly speared over a selected area such that connectivity and coverage were preserved. Then, the three methods to remove nodes were applied. We checked the energy of the remaining network while working. This experiment was repeated several times and average energy consumption was calculated and reported. Figure 5 shows the graph of average consumed energy of overall networks during the time and Figure 6 shows the bar chart of average network energy consumption in different methods with 100 nodes network.

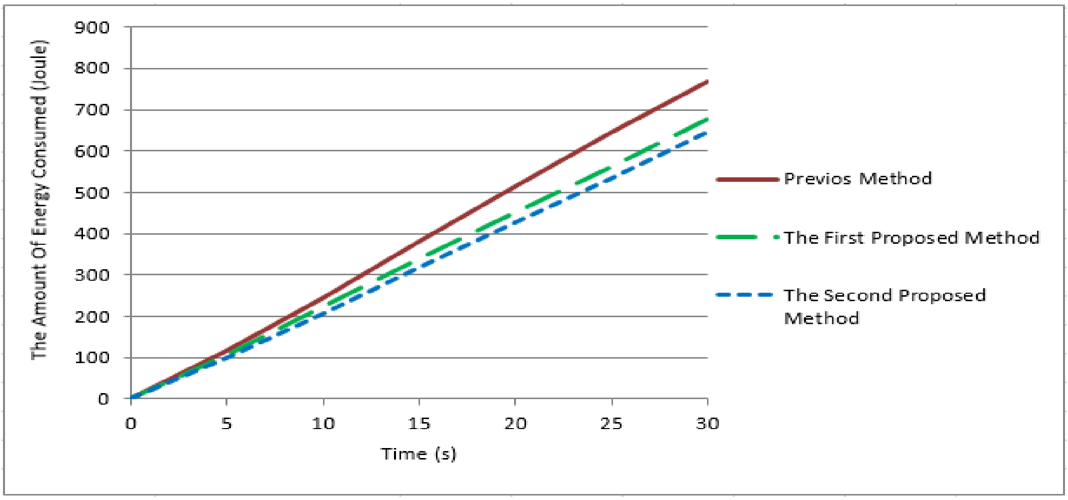

Fig. 5. Graph of average energy consumption of networks with 100 nodes network in different methods

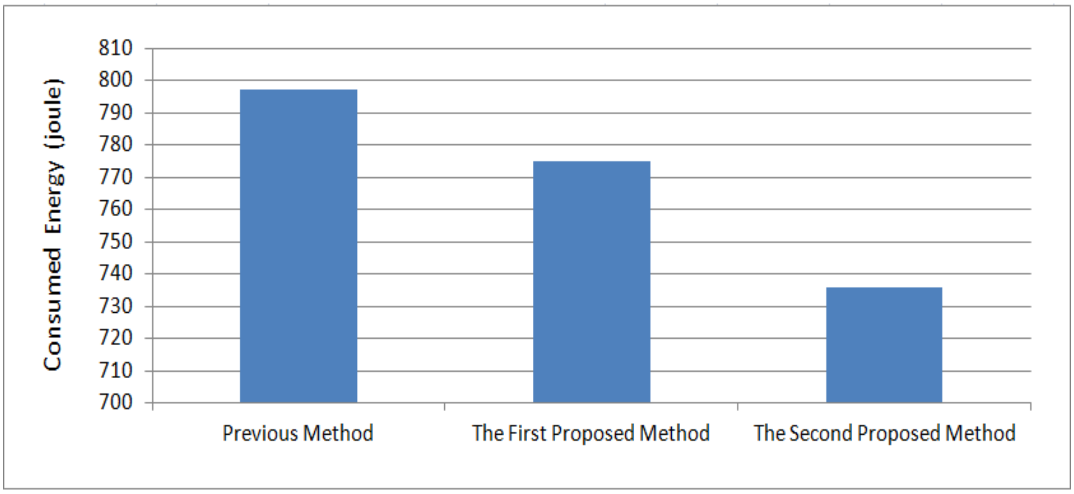

Fig. 6. Average energy consumption of networks with 100 nodes in different methods

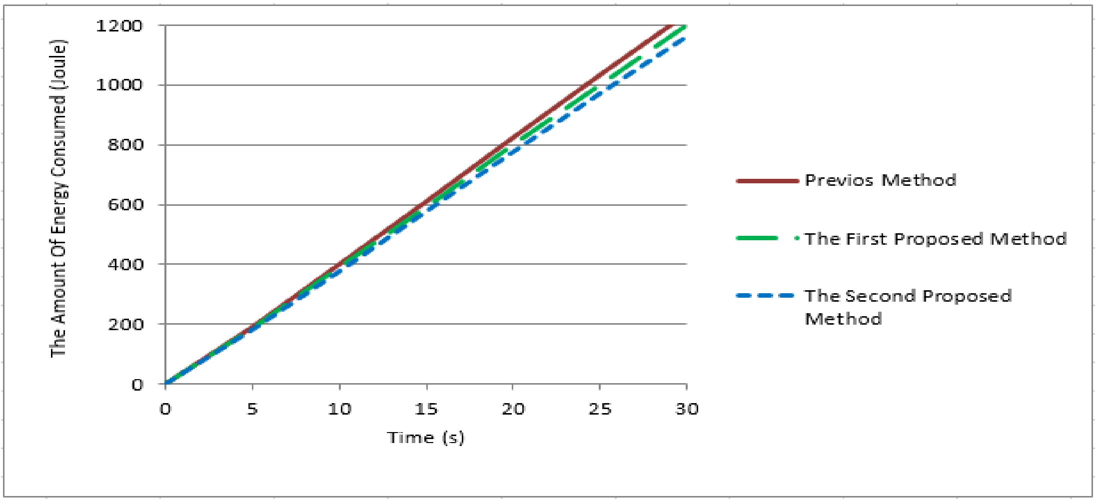

Fig. 7. Graph of average energy consumption of networks with 150 nodes in different methods

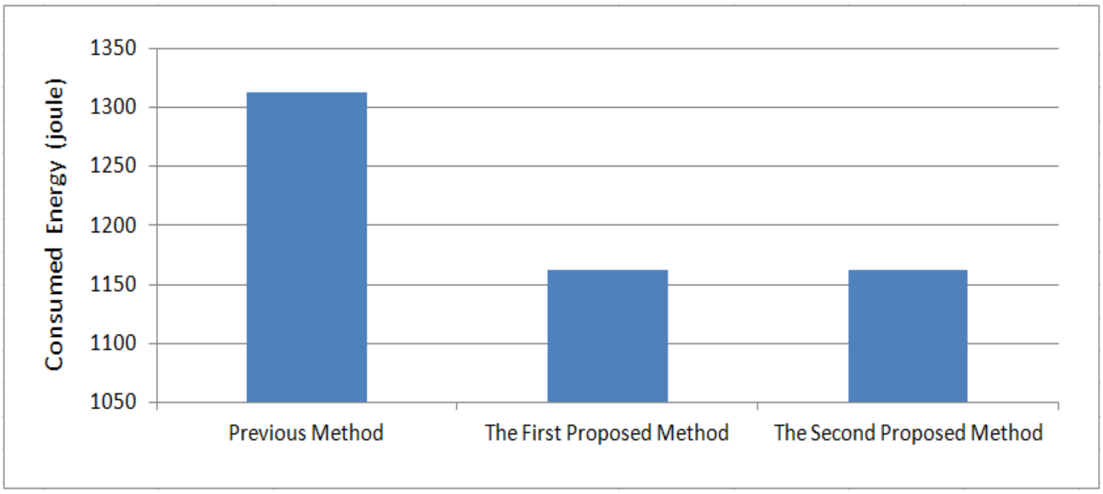

Fig. 8. Average energy consumption of networks with 150 nodes in different methods

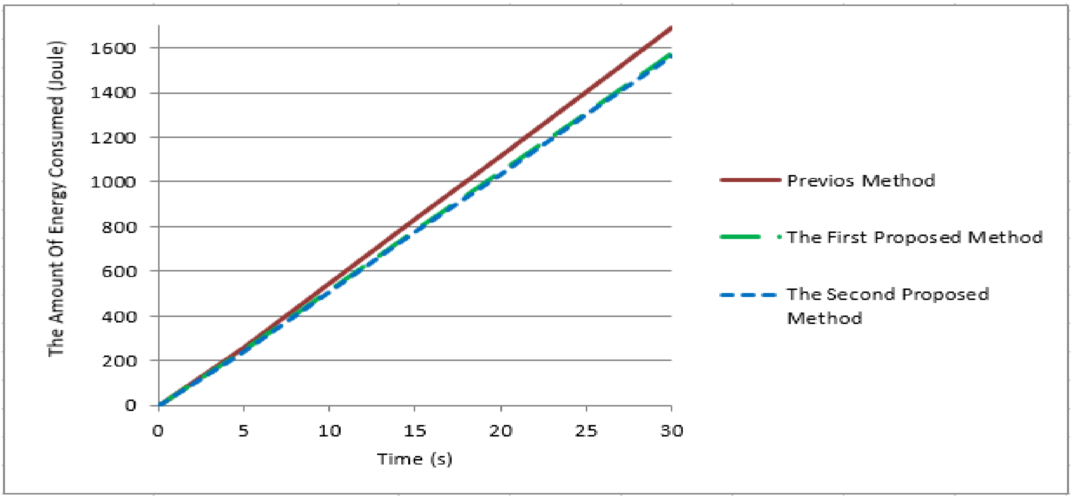

Fig. 9. Graph of average energy consumption of networks with 200 nodes in different methods

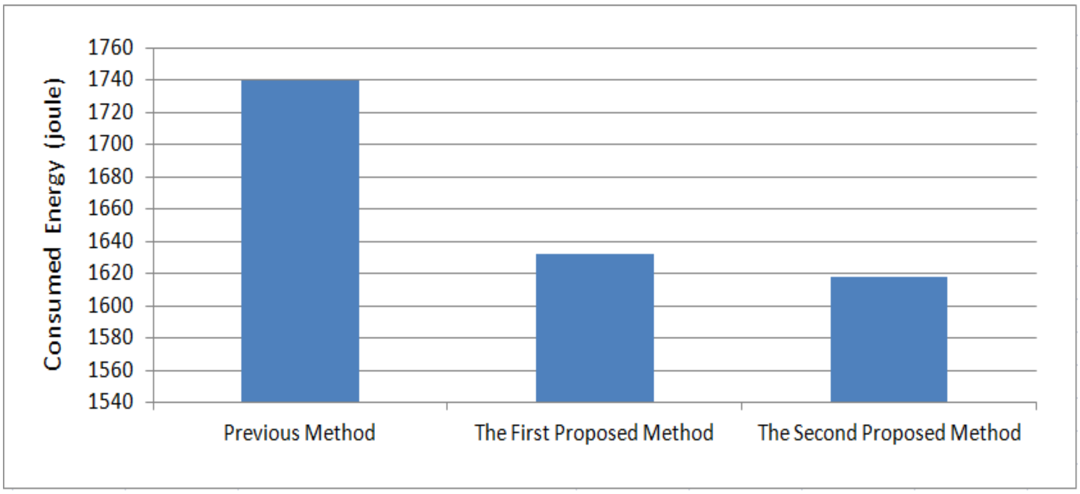

Fig. 10. Average energy consumption of networks with 200 nodes in different methods

In the second stage, we repeated the previous experiment on networks with 150 nodes in the same area to examine the impact of the proposed methods. Figures 7 and 8 depict the linear and column charts of the network energy consumption. Furthermore, we tested these proposed methods on networks with 200 nodes and repeated the previous experiment. The amount of average energy consumption of the networks is shown in Figures 9 and 10.

The simulation results through Figures 5 to 10 indicate that the amount of overall network energy consumption of the proposed methods in comparison with existing methods is decreased. The simulation results also indicate that the proposed methods, with effective selection of redundant nodes can reduce the amount of network energy consumption to a greater extent. Furthermore, as the number of the nodes increases in the network that makes a dense network, the second proposed method work better.

5. Conclusion

In this paper, two methods were presented using graph theories to identify and select redundant nodes in large-scale WSNs. The proposed methods dealt with connection and coverage in the large-scale WSN and identified redundant nodes to remove accordingly, which in turn preserved the overall network connection and coverage. In the proposed methods, the minimum spanning tree, eccentricity, and also the introduction of neighborhood parameters are used to identify and select the redundant nodes that removing or turning them off doesn't affect the network connection and coverage. The first method could detect lateral redundant nodes that mainly exist in non-dense networks. The second method improves the first method by a little more complexity to detect intermediate and lateral redundant nodes. The simulation results indicate that the two proposed methods are more effective than the previous method and work well in dense and non-dense large-scale WSNs. So, the first method is proper for non-dense networks and the second for the dense large-scale WSNs.

References

[1] J. Zheng, and A. Jamalipour, "Wireless sensor networks: A networking perspective". *John Wiley & Sons*, United States, 2009.

[2] S. J. Ramson, and D. J. Moni, "Applications of wireless sensor networks - A survey". Proceedings of *International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, pp. 325-329, 2017.

[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey". *Computer Networks*, vol. 52, pp. 2292-2330, 2008.

[4] Z. Jiao, H. Wang, and K. Ma, "The connectivity measurement in complex directed networks by motif structure". *International Journal of Sensor Networks*, vol. 21, 2016.

[5] X. Bai, Z. Yun, D. Xuan, T. H. Lai, and W. Jia, "Optimal patterns for four-connectivity and full coverage in wireless sensor networks" *IEEE Transactions on Mobile Computing*, vol. 9, pp. 435-448, 2010.

[6] M. Cardei, and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks". *Computer communications*, vol. 29, pp. 413-420, 2006.

[7] S. He, J. Chen, and Y. Sun, "Coverage and connectivity in duty-cycled wireless sensor networks for event monitoring". *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 475-482, 2012.

[8] M. Rebai, H. Snoussi, F. Hnaien, and L. Khoukhi, "Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks". *Computers & Operations Research*, vol. 59, pp. 11-21, 2015.

[9] H. Lin, L. Wang, and R. Kong, "Energy efficient clustering protocol for large-scale sensor networks". *IEEE Sensors Journal*, vol. 15, pp. 7150-7160, 2015.

[10] M. Rebai, H. M. Afsar, and H. Snoussi, "Exact methods for sensor deployment problem with connectivity constraint in wireless sensor networks". *International Journal of Sensor Networks*, vol. 21, 2016.

[11] L. Li, J. Y. Halpern, P. Bahl, Y. M. Wang, and R. Wattenhofer, "A cone-based distributed topology-control algorithm for wireless multi-hop networks". *IEEE/ACM Transactions on Networking*, vol. 13, pp. 147-159, 2005.

[12] S. Misra, S. D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements". *The 27th Conference on Computer Communications*, 2008.

[13] H. Teng, K. Zhang, M. Dong, K. Ota, A. Liu, M. Zhao, and T. Wang, "Adaptive Transmission Range Based Topology Control Scheme for Fast and Reliable Data Collection", *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[14] N. Xu, A. Huang, T. W. Hou, and H. H. Chen, "Coverage and connectivity guaranteed topology control algorithm for cluster-based wireless sensor networks", *Wireless Communications and Mobile Computing*, vol. 12, pp. 23-32, 2012.

[15] J. S. Leu, T. H. Chiang, M. C. Yu, and K. W. Su, "Energy efficient clustering scheme for prolonging the lifetime of wireless sensor network with isolated nodes". *IEEE communications letters*, vol. 19, pp. 259-262, 2015.

[16] D. Feng, C. Jiang, C. Lim, L. J. Cimini, G. Feng, and G. Y. Li, "A Survey of Energy-Efficient Wireless Communications". *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 167-178, 2012

[17] M. Garcia, S. Sendra, and J. Lloret, "Saving energy and improving communications using cooperative group-based wireless sensor networks". *Telecommunication Systems*, vol. 52, pp. 2489-2502, 2013.

[18] Z. Chen, M. Ma, X. Liu, A. Liu, and M. Zhao, "Reliability Improved Cooperative Communication over Wireless Sensor Networks". *Symmetry*, vol. 9, 2017.

[19] P. Kumar, M. P. Singh, and U. S. Triar, "A review of routing protocols in Wireless Sensor Network". *International Journal of Engineering Research & Technology*, vol. 1, 2012.

[20] Singh, Santar Pal, and S. C. Sharma. "A survey on cluster based routing protocols in wireless sensor networks." *Procedia computer science* 45, pp. 687-695, 2015.

[21] M. H. Anisi, A. H. Abdullah, S. A. Razak, and M. A. Ngadi, "Overview of data routing approaches for wireless sensor networks". *Sensors*, vol. 12, pp. 3964-3996, 2012.

[22] L. Malathi, and R. K. Gnanamurthy, "Cluster based hierarchical routing protocol for WSN with Energy Efficiency". *International Journal of Machine Learning and Computing*, vol. 4, 2014.

[23] T. Amgoth, and P. K. Jana, "Energy-aware routing algorithm for wireless sensor networks". *Computers and Electrical Engineering*, vol. 41, pp. 357-367, 2015.

[24] G. S. Brar, S. Rani, V. Chopra, R. Malhotra, H. Song, and S. H. Ahmed, "Energy efficient direction-based PDORP routing protocol for WSN". *IEEE Access*, vol. 4, pp. 3182-3194, 2016.

[25] G. J. Horng, T. Y. Chang, and S.T. Cheng, "An effective node-selection scheme for the energy efficiency of solar-powered WSNs in a stream environment". *Expert Systems with Applications*, vol. 41, pp. 3143-3156, 2014.

[26] L. M. Kaplan, "Global node selection for localization in a distributed sensor network". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, pp. 113-135, 2006.

[27] O. Demigha, W. K. Hidouci, and T. Ahmed, "On energy efficiency in collaborative target tracking in wireless sensor network: a review". *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 1210-1222, 2013.

[28] F. R. Armaghani, I. Gondal, J. Kamruzzaman, and D. G. Green, "Sensor selection for tracking multiple groups of targets". *Journal of Network and Computer Applications*, vol. 46, pp. 36-47, 2014.

[29] H. Wang, H. E. Roman, L. Yuan, Y. Huang, and R. Wang, "Connectivity, coverage and power consumption in large-scale wireless sensor networks". *Computer Networks*, vol. 75, pp. 212-225, 2014.

[30] T. H. Cormen, "Introduction to algorithms", *MIT press*, United States, 2009.

[31] R. E. Neapolitan, and K. Naimipour, "Foundations of algorithms", *Jones & Bartlett Learning*, 2010.

[32] G. Chartrand, and P. Zhang, "A first course in graph theory". *Courier Corporation*, 2012.