

# Dynamic Tree- Based Routing: Applied in Wireless Sensor Network and IOT

Mehdi Khazaei<sup>1\*</sup>

<sup>1</sup>. Department of Computer Engineering, Kermanshah University of Technology

Received: 28 Aug 2021/ Revised: 07 Nov 2021/ Accepted: 07 Dec 2021

## Abstract

The Internet of Things (IOT) has advanced in parallel with the wireless sensor network (WSN) and the WSN is an IOT empowerment. The IOT, through the internet provides the connection between the defined objects in apprehending and supervising the environment. In some applications, the IOT is converted into WSN with the same descriptions and limitations. Working with WSN is limited to energy, memory and computational ability of the sensor nodes. This makes the energy consumption to be wise if protection of network reliability is sought. The newly developed and effective hierarchical and clustering techniques are to overcome these limitations. The method proposed in this article, regarding energy consumption reduction is tree-based hierarchical technique, used clustering based on dynamic structure. In this method, the location-based and time-based properties of the sensor nodes are applied leading to provision of a greedy method as to form the subtree leaves. The rest of the tree structure up to the root, would be formed by applying the centrality concept in the network theory by the base station. The simulation reveals that the scalability and fairness parameter in energy consumption compare to the similar method has improved, thus, prolonged network lifetime and reliability.

**Keywords:** IOT; Wireless Sensor Network; Tree-Based Hierarchy; Clustering; Energy Consumption; Fairness.

## 1- Introduction

The latest advances made in electronics and wireless telecommunication fields has made the design and construction of multipurpose, small, and low energy consumption sensors at low price. These small sensors are capable of receiving information from the environment, and transmit processed data. This fact has initiated the idea of Wireless Sensor Network (WSN).

The WSN consists of many sensor nodes, scattered vastly in an environment which collect information therefrom. The location of the sensors is not necessarily determined, which allows them to be released in risky or inaccessible conditions. Therefore, the WSN algorithms must be self-organized with the cooperation capability. Each sensor has a processor, to first, run a series of simple and initial processing on the information, and next, transmit them. Each sensor has low capabilities, but a combination of them, introduce powerful possibilities. The strength of WSN is applying the sensors, in supervising the environment conditions, structures and facilities' well

performance. WSN is widely used in agriculture, medicine, industry and military [1].

In an environment, the scattered sensors collect information and transmit them to the Base Station (BS), through a multi-hop route, which has no definite infrastructure. WSN designing is subject to many criteria among which fault tolerance, scalability and cost are the essential. Fault tolerance is one of the main challenges in WSN, thus, the researchers investigate to reduce energy consumption in different applications of WSN. Because sensors out of energy directly affect the networks general functionality and fault tolerance, so if the sensors energy are managed well the reliability will be accomplished to a high degree [2].

It is common to apply the hierarchical methods in reducing energy consumption and increasing WSN lifetime. The hierarchical methods are the mechanism to systematize the sensors in a layered structure and assign a specific task to the sensors of each layer before the data is transmitted to the next layer. This structure is a solution for overcoming some of the limitations in WSN.

The tree-based topology is one of the hierarchical structures upon which the method of this article is based on. The advantage of tree-based topology is low energy consumption in data transfer, low complexities and implementation cost.

The disadvantages consist of: non- scalability, load unbalancing, high probability of error, and unequal delays.

Many researches have been done to tree-based, have tried to solve the disadvantages the method. The most renowned tree-based methods the EADAT, BATR, PEDAP, and ETR would be addressed in brief [3,4].

As to EADAT [5], the BS acts as the root and the algorithm begins to run when the control messages are sent by the root. Every node that receives the control message, adjust a proper timer, inversed with the remaining energy. In this interval, a node selects a parent with more energy and low distance, and propagates the controlling messages before the time ends. The outcome of this method is a multicast tree with a BS root which is updated alternatively. Any node, the residue energy of which is below a threshold, sends a help message and enters the standby until its children join under the new subtree. By receiving the new control message, this node joins the new parent, otherwise it sounds alarm. The main advantage of this method is that the nodes with high energy most probably do not become leaves. Therefore, unsuccessful transfers are reduced and approximately achieved load balancing, while, the formation of longer routes is inevitable. The long unequal routes would increase delay, thus, higher energy consumption.

As to the BATR [6], the BS collects all nodes location and forms the routes. The BS forms a minimum spanning tree concerning the energy consumption as to balance out the children under the subtrees. These features can prolong network lifetime and cause load balancing. Since in tree formation, residue energy is not considering, and it is assumed that the nodes generate uniform data, thus, some nodes die earlier, caused unsuccessful transfer.

As to PEDAP [7], the minimum spanning tree is applied to compute the consumed energy. In PEDAP, the residual energy of the nodes is considered in beside distance and data volume to form tree. The tree based on these features, can balance energy consumption and transfer distance, leading to a reduction in delay. In the large and complex networks, energy computation is hard and constructing a tree would consume more energy, indicating lack of scalability in PEDAP.

As to ETR [8], it is the improved version of TR, applied to be balance both the efficiency and cost. In ETR, each node saves a table with low energy, to determine next node with least hop to the root. The selected route hops are less than that of the actual route. Every node, eventually, selects a route with the least hop count, otherwise, considering their parent as the next node. In ETR, the residue energy of the nodes is not concerned in route selection, thus, paths with less residual energy may be selected, which leads to premature death of nodes and no data transmission.

In mentioned methods, data collection is necessary, while, no Clustering is performed in none of them. The EADAT, PEDAP and ETR are fit for small scale networks and

BATR for large scale networks. In the EADAT, BATR and PEDAP, the routs are selected through the BS in a centralized manner, consequently, a high capacity of data should be sent to the BS through the nodes, which is time consuming. In the ETR, the nodes exchange information in forming tables and it is a non-centralized method. The ETR is more flexible, rapid and environment changing adaptable. In the BATR and ETR, the residue energy is not considered in tree formation. Therefore, there exists the possibility of selecting nodes with low energy as the parent, thus, loss of data. In EADAT and PEDAP, the tree is formed by considering the residue energy of nodes, leading to a balance in energy consumption and network lifetime prolongation. In the EADAT, PEDAT and ETR methods, the shortest route is selected for data transfer while in BATR distance is not considered and the tree is balanced based on location and number of nodes. [3,4].

In most studies, the tree-based topologies are independent of clustering, while in this proposed method, to select the parents, the clustering method is applied to balance energy consumption and prevent data loss. To improve scalability of the tree-method, when it is revealed that there is no fairness in energy consumption of sensors, a tree structure will be reformed, to balance the energy consumption of the sensors. With these in mind, the innovation of the proposed method is in overcoming tree-based method disadvantages, specially the lack of scalability and load balancing.

This article is structured as follows: the study design is introduced in Sec. 2; the method is proposed in Sec. 3; the method efficiency is evaluated in Sec. 4 and the conclusion and future works are presented in Sec. 5.

## 2- The Study Design

A WSN consists of sensors, scattered in an environment, collecting data therein. Each sensor has a sense range, through which it can communicate with the available points, consequently, WSN applies the multi-hop communication protocols in information transmission between sensors and BS. In designing WSN attempt is made to have a complete coverage of the area through the sensors. A sensor consists of sense unit, process unit, send/receive unit and power unit and upon their applications, they can contain GPS, power generator, and motion module. The sensors, according to what they perceive, transmit digital signals to the processing unit. This unit with its small memory manages the sensors' cooperation during the assigned task performance. The power unit is highly essential, which is supported by battery or power collection unit like the solar cells.

Some sensors, for any reason may be out of power or dysfunctional and this should not affect the overall network functionality. Therefore, fault tolerance is defined the WSN operation maintaining, regardless of some

sensors dysfunction. A properly designed WSN, upon some sensor dysfunction would immediately adopt itself to the new conditions and continue its operation.

In WSN, the energy limitation of sensors is the main challenge of fault tolerance. Methods like adapting signal strength to distance, sleep scheduling, number of package sent reduction and hierarchical topologies are applied in reducing energy consumption in WSN [9].

The WSN expansion especially in IOT, has made scalability a challenging issue. The hierarchical structure can be a solution for load balancing, fault tolerance, increasing connections, decreasing delay and WSN lifetime prolongation. In this structure, a specified task is defined for the sensors before they transmit data to higher levels. Clustering is the most popular measure in the hierarchical methods [10-14].

In algorithm of clustering the three main elements consist of the sensor, Cluster Head (CH) and BS. The sensors send the collected data to the CH, and the CH sends the data to the BS, usually, is distanced from the sensors. The CH act as the gate between sensors and BS, that is a CH is considered as a sink for sensors and BS is as a sink for the CHs. There exist the two homogenous/ heterogeneous and static/dynamic mechanisms to classify the clustering methods [15].

In the heterogeneous WSN, there exist two groups of sensors: 1) with high processing power and a complex hardware usually applied as the main backbone of the network, and 2) with proportionally lower processing power, which receive information from the environment. In the homogeneous WSN, all sensors have uniform hardware features and processing power, where they can be CH and this role changes periodically among the sensors. In the homogeneous networks, load balancing takes place and energy consumption would be more uniform. The static clusters are usually formed when the network is being formed by heterogeneous sensors and the network designers seek to form the clusters around strong sensors. The cluster size, CH and the number of involved sensors are all static. Applying this type of clusters is limited to the predefined scenarios. In dynamic structure the sensors do not belong to one cluster and it's possible to form different clusters in due time. Dynamic structure is usually to homogeneous networks, but it's applicable in heterogeneous networks as well. Dynamic cluster structure increases efficiency [16].

In clustering, communication is categorized to the inter-cluster and intra-cluster. The intra-cluster communication exchanged messages delivered between the sensors of cluster and CH. The inter-cluster includes messages exchanged between the CHs and BS. When only the CHs transmit information, the collision between the sensors of a cluster is prevented, thus, a saving in energy. The most famous clustering method is the Leach since introduction, some optimizations have been provided.

The dividing and communication manner of the newly introduced hierarchical routing methods differ from that of the traditional clustering methods. These new topologies are: tree-based, chain-based, grid-based and area-based, Fig. (1) [3,4].

- **Tree-based Topology:** This topology consists of branches, leaves and parents. The data are sent to the parents from the children until the data reach the BS. The accumulation and elimination of the repeated data takes place by passing the data through each sensor. Formation of this tree is simple, no need for clustering. It would suffice that each sensor sent the data to the next higher sensor, which is closer to the root, thus, a reduction in energy consumption. The weak point of tree topology is when a parent sensor stopped, the data sent by the children no guided. In tree-based method, the sensors close to the root consume more energy due to data accumulation and there exist no fairness among the sensors. Also, if number of the sensor is high the delay would be varying to data reaching the BS, lacking of scalability in this topology.
- **Chain-based Topology:** The WSN is formed from one or few chains where for each, one leader is selected. Each sensor sends the data to the nearest sensor in the chain until the data reaches the chain leader. During this data transfer the accumulation is made by each sensor, then, aggregated data is sent to BS by leader. Constructing and maintaining this topology is simple, consequently, a saving in energy consumption. If the sensors stop, all the passing data become lost. Considering the sensors position in relation to leader, data transmission delay and the passed data rate from the sensors would be different, leading to an imbalance in the sensors' energy consumption.
- **Grid-based Topology:** The whole network is divided into grids and for each, a leader will be selected. All grid members send their data to the grid leader which sends them to the next leader, to be reach to BS. Grid formation is simple and is based on the geographic position of the network sensors. In this topology the grid size remains the same and only the leader position is changed, this is why routing in grid is local. If in each grid the number of the sensor is high, and the leaders are not selected properly, their energy consumption would be very high.
- **Area-based Topology:** Here, all WSNs are divided in areas of different size. The BS would send data accumulation request to the closest sensor in each area, in a flooding manner to reach source. Then the data is sent to the BS by source. This topology is appropriate for mobile BS which move in determined geographical area.

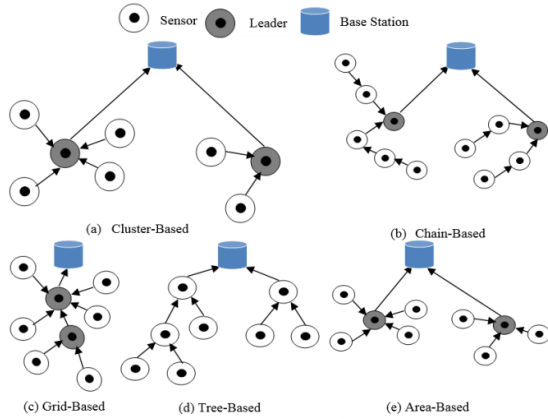


Fig. 1 Hierarchical topologies structure

Because more energy is needed to communicate than to data processing, the objective of hierarchical methods is to reduce sending repeated packets towards BS. (e.g. to send 100-bit at 100 m distance, requires about 10  $\mu$ J energy, while to implement a 32-bit instruction in a loop of one-hundred-thousand, the consumed energy would be about 60 pJ). It is obvious that, one of the best means in reducing sensors energy consumption is reducing the data communication. In the hierarchical structures, this issue is performed through the data compression algorithms, repetitive data elimination and applying the location-based and time-based local properties of the data [17].

### 3- The Proposed Method

The proposed method is hierarchical spanning tree. To construct tree, first, the WSN is modeled through the  $G=\langle V, E \rangle$  graph, Fig. (2), where  $V$  is the sensors and  $E$  is the total edges of the graph. According to Eq. (1), an edge is defined between two sensors when the distance of both is less than  $d$ , and the  $d$  is computed through the free space model [6].

$$\forall x, y \in V, \quad (x, y) \in E \Leftrightarrow dist(x, y) \leq d. \quad (1)$$

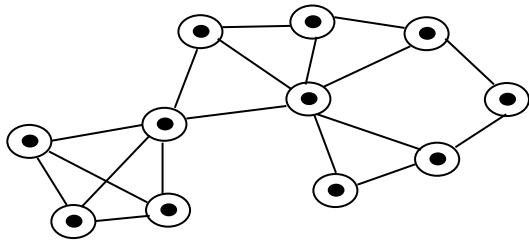


Fig. 2 Modeling WSN through the graph.

The spanning tree construction through the graph consist of two stages: 1) the sensors are clustered through greedy algorithms, and by applying the Leach algorithm a CH is selected for every cluster. These selected CH acts as the sensor (leaves of tree) parents. 2) CHs of stage one are clusters based on network centrality theory, and next renewed CHs are

selected through the Leach algorithm. This process continuous until the tree root, the BS, is achieved. The proposed method is named Dynamic Tree-Based Routing (DTR).

#### 3-1- Constructing Subtree of Leaves

Subtree of leaves is formed based on the communicative properties between the sensors and a greedy algorithm. Because sensors have location-based and time-based properties, attempt is made to allocate the sensors with high dependency under a subtree with one parent. Therefore, for every edge in  $G$ , the edge weight is calculated as Eq. (2), to indicate the sensor dependency.

$$D(A_i, A_j) = \left\lceil \frac{\sum D_s}{\sum D_t} * 100 \right\rceil. \quad (2)$$

In Eq. (2), if  $A_i$  and  $A_j$  are the  $i^{th}$  and  $j^{th}$  sensors, the dependency between them is defined: the sum of similar collected data divided in to the sum of all collected data through them in a specific time. The more similar data is collected, the  $D(A_i, A_j)$  is increased, and vice-versa. If the weight between the two sensors is zero, there exists no dependency. The dependency between the sensors is a symmetric function, therefore, the graph is undirected. In order to allocate sensors with high dependency in a subtree, the quality function is defined through Eq. (3).

$$Q(C) = \frac{1}{|A|} \sum_{c_i \in C} \sum_{e_{xy} \in E^i} Dep(a_x^i, a_y^i). \quad (3)$$

During running the Algorithm 1, the subtree of the leaves is formed in a manner to maximize the quality function at each stage. The obtained values by running the algorithm, would be the guide in forming the subtrees of leaves [18]. After running the algorithm, clusters are formed out of the sensors. Then the Leach algorithm is run for each cluster, and the selected CH acts as the sensors' parent in the cluster. The rest of the stages of connecting parent with the leaves, would be in accordance with the Leach algorithm. The clusters formed at this stage are subtrees that the leaves of the tree (sensor nodes) are members, Fig. (3).

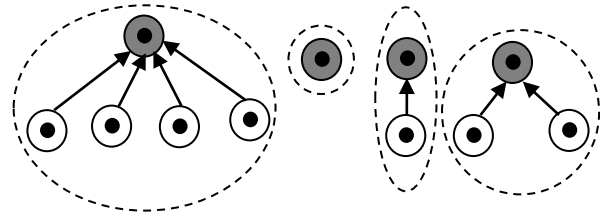


Fig. 3 Formation of clusters each being a subtree of the leaves.

In algorithm 1, the input of the function is the network graph model. The names of the functions are chosen so that they clearly represent their function. In general, functionality of the algorithm is briefed as: the edge with the highest weight is selected so that it is not a member of any cluster; if one of the vertices of this edge has been a cluster member, the

other vertex could join the cluster, in case the quality function is improved. If both the edge vertices are each a member of two different clusters, the two clusters would be merged if the quality function increase. At the end, if any non-cluster member vertex is left, a new cluster is formed for it. After the clustering process, the Leach\_Algorithm () is called to determine the CH of each cluster; following this, the model graph is updated and returned [19].

**Algorithm. 1** The subtrees formation algorithm of leaves

```

1- Subtree Leaves_Clustering(WSN S)
2-
3- G(A, E)=Model_WSN_to_Graph(S);
4- C=φ;
5-
6- while(∃ e ∈ E: Unprocessed_Edge(e))
7-
8-   exy=Find_Edge_with_Max_Weight();
9-   ax, ay=Return_Head_of_Edge(exy);
10-
11-   if (∀ c ∈ C: ax∉ c ∧ ay∉ c)
12-     cnew=Create_Cluster(ax, ay, exy);
13-     C=C+cnew;
14-     Update_Graph();
15-
16-   else if (∀ c ∈ C: (ax ∈ c ∧ ay ∉ c) ∨ (ax ∉ c ∧ ay ∈ c))
17-     Qnew=Q(C);
18-     if (Qnew > Qold)
19-       c=c+Add_Head_not_Belongs_Cluster(ax, ay, exy);
20-       Updating_Graph();
21-
22-     else
23-       Do_nothing();
24-
25-   else if (∃ c, c' ∈ C: (ax ∈ c ∧ ay ∈ c'))
26-     Qnew=Q(C);
27-     if (Qnew > Qold)
28-       cnew=c∪c';
29-       C=C-(c∩c');
30-       C=C+cnew;
31-       Updating_Graph();
32-
33-     else
34-       Do_nothing();
35-
36-   END If
37-
38- END While
39-
40- for(∀ ai ∈ A, ∄ c ∈ C: ai ∈ c)
41-   cnew=Create_Cluster(ai);
42-   C=C+cnew;
43- END For
44-
45- H=Leach_Algorithm(C);
46- Updating_Graph();
47- return G;
48-
49- END Function

```

### 3-2- Constructing the Rest of the Tree up to the Root

To construct the rest of the tree up to the root, the sensors selected as parent in the previews stage are considered, and clustering is repeated. Since at higher levels, there is not dependency between the sensors according to Eq. (2), the algorithm 1 does not work correctly. Hence, the rest of the tree up to the root is formed through the concept of sensor centrality in network theory. The importance of a sensor in a network depends on its neighbor importance, Eq. (4). Where,  $X_i$  indicates the  $i^{\text{th}}$  sensor importance and  $A_{ij}$  is the array of the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column of the adjacency matrix of the graph.

$$X_i = C \sum_{j=1}^n A_{ij} X_j. \quad (4)$$

Equation 4 can be written in it matrix form, Eq. (5) [20].

$$X = CAX \quad \text{or} \quad X(I - CA) = 0. \quad (5)$$

$X$  is the eigenvector of the adjacency matrix  $A$ . Matrix  $A$  determines the relation between the sensors and is defined for the weightless graphs, while as to Eq. (2), for the graph model edges, weight is defined. For this purpose, matrix  $A$  is replaced with matrix  $M$ , and its arrays are obtained through Eq. (6) [21].

$$\forall i, j: \quad m_{ij} = D(A_i, A_j). \quad (6)$$

Because the eigenvector computation of the matrix  $M$  through Eq. (5) is time consuming with the probability of obtaining some negative values, the Perron vector of matrix  $M$  is computed. The Perron vector is a type of the matrix eigenvector the values of which are non-negative with a sum of one. Each entity of Perron vector is computed according to Eq. (7) [22].

$$X^k = \frac{\sum_{i=1}^k \sum_{j=1}^n m_{ij} * X_j^k}{\|\sum_{i=1}^k \sum_{j=1}^n m_{ij} * X_j^k\|_1}. \quad (7)$$

After importance values of sensors is obtained, clustering of the next level begins according to the Algorithm 2. The functionality of this algorithm is in a sense that: the sensors with importance above average importance of the sensors, form a cluster. For a non-cluster sensor like  $P_i$ , a neighbor with the highest values like  $h_j$  is found. If  $h_j$  is the cluster member then,  $P_i$  join this cluster, otherwise,  $p_i$  and  $h_j$  form a new cluster. CH of each cluster is selected as parent and the rest children, by the Leach\_Algorithm (). Following, the updated graph is returned.

In the first call, the input of algorithm 2 is the returned graph of algorithm 1, while, in the next calls, the returned graph of the previous stage of itself. At each call of Algorithm 2, one tree level is constructed, and this pattern continuous up to the root, BS is reached, Fig. (4) [23].

Algorithm. 2 The algorithm of constructing the rest of the tree

```

1- Level_of_Tree_Organizing (Modified_Graph G)
2-
3- H=Return_Heads_of_Clusters (G);
4- C=  $\phi$ ;
5-
6- while ( $\exists h_i \in H$ )
7-   M=M+ Eligibility ( $h_i$ );
8-   Ave =  $\frac{M}{|H|}$ ;
9-   END While
10-
11- While ( $\exists h_i \in H$ )
12-
13-   If (Eligibility ( $h_i$ )  $\geq$  Ave)
14-      $c_{new}=h_i$ ;
15-     C=C+ $c_{new}$ ;
16-     Update_Graph ();
17-   else
18-     Do_nothing ();
19-
20-   END While
21-
22- while ( $\exists h_i \in H: h_i \notin c$ )
23-
24-    $p_i$ =Find_neighbor_with_Max_Eligibility ( $h_i$ );
25-   if ( $\exists c \in C: p_i \in c$ )
26-      $c=c \cup p_i$ ;
27-     Update_Graph();
28-   else
29-      $c_{new}=h_i \cup p_i$ ;
30-     C=C+ $c_{new}$ ;
31-     Update_Graph();
32-
33-   END While
34-
35- H=Leach_Algorithm(C);
36- Updating_Graph();
37- return G;
38-
39- END Function

```

### 3-3- Tree Reformation

If there is no change in the parent nodes close to the root, which consume high energy in sending data of their children, the energy will end rapidly. In a sense, sensors dependency is changed in due time, therefore, reformation of the tree during the network lifetime is inevitable.

In DTR, the sensors next to sending data to their parents, they send the residue energy to let BS become aware of all sensors residue energy. The sending residue energy as piggyback of data, causing energy saving in the sensors. If for a given time no data sent to BS by the sensors, based on the available observation, by applying the Normalized Least Mean Square method (NLMS), the residue energy of these sensors would be predicted through Table 1 [24].

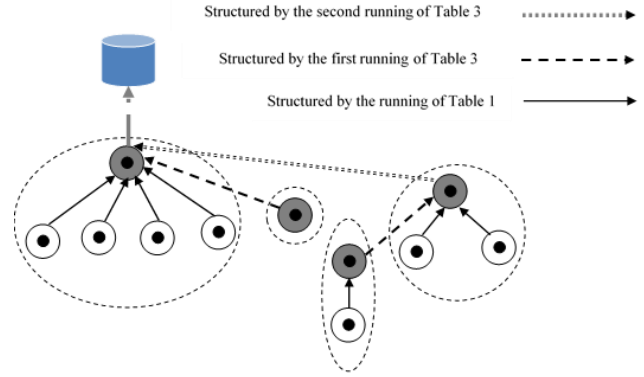


Fig. 4 The tree of network graph model

In Table 1, the underlined variables are vectors and the over-lined ones are to hold the predicted values. The  $W$  is the vector of prediction coefficient filter at size  $q$  and  $E$  is a vector to hold the  $q$  value of process residue energy expressed through Eqs. (8 and 9).

Table.1 The sensors' residue energy value prediction by NLMS

Number	Equations
1	$R_{n-2} = E_{n-1} - \bar{E}_{n-1}$
2	$\underline{W}_n = \underline{W}_{n-1} + \omega * \frac{R_{n-2} * \underline{E}_{n-2}}{\ \underline{E}_{n-2}\ ^2}$
3	$\underline{E}_{n-1} = \text{combine } E_{n-1} \text{ and } \underline{E}_{n-2}$
4	$\bar{E}_n = \underline{W}_n^T * \underline{E}_{n-1}$

$$\underline{W} = [w_1, w_2, \dots, w_{19}]^T. \quad (8)$$

$$\underline{X}_n = [x_n, x_{n-1}, \dots, x_{n-19}]^T. \quad (9)$$

The initial value of  $W$  is zero which can be updated per each new data.  $\bar{E}_n$  is the predicted residue energy of  $E_n$ . BS accumulates all the residue energy of the sensors that is observing fairness in energy consumption. Therefore, the Jain's index, a statistical scattering measure is applied in measuring the uneven energy consumption level in the sensors. This index with a value within 0-1 is obtained through Eq. (10) [25].

$$J_c = \frac{(\sum_{i=1}^m E_i)^2}{k \sum_{i=1}^m E_i^2}. \quad (10)$$

$E_i$  is the residue energy of the  $i^{\text{th}}$  sensor and  $K$  is the total number of WSN sensors. The closer the  $J_c$  rate to 1, the energy consumption is more balanced and vice-versa. The BS computes the determined  $J_c$  rate at different times and if it is below the threshold ( $T_h$ ), the tree reformation is computed and the order of this formation is send to the sensors in the WSN.

### 4- Efficiency Evaluation

The MATLAB is applied to analyze and implement the DTR. The DTR is to be compared with BATR. Similar to

most of the available studies the energy model is considered only for communication, the other consumed energy is ignored. In this model, the consumed energy to send and receive  $L$  bits of data at a distance of  $d$  meters, is computed through Eq. (11 and 12), respectively [6].

$$E_{tx}(L, d) = L * E_c + L * e * d^2. \tag{11}$$

$$E_{rx} = L * E_c. \tag{12}$$

$E_c$  is the base energy to turn on the sender and receiver circuit and  $d$  is the threshold distance defined in Eq. (1). To implement these comparable methods, all homogenous sensors are considered with unequal initial energy, distributed in a  $100 \times 100 \text{ m}^2$  environment in a random manner. The BS is situated outside the  $50 \times 150 \text{ m}^2$  environment. To determine the accuracy of the obtained data and sensors' independence of spread in the environment, tests are repeated to achieve 95% confidence interval. The values of the DTR parameters and the Leach algorithm are according to Table 2 [26].

Table. 2 The applied parameters' value

Parameters	Value
$P_{Leach}$	0.05
$q$	20
$E_c$	50 nJ/bit
$e$	10 PJ/bit.m <sup>2</sup>
$d$	86.2 m
$L$	525 bytes
$r$	10000 round
$T_h$	0.9

The First Node Die (FND), the Last Node Dies (LND), the Standard Deviation of Hops Count (SDHC) and the Fairness Parameter (FP) are considered to evaluate the DTR efficiency. The FND is the time duration until the energy of the first sensor is totally consumed. The LND is the time duration until energy of all sensors is totally consumed. This criterion can be defined based on the past rounds. The more distance between the FND and LND indicates an imbalance in sensors' energy consumption and vice-versa. The SDHC determines standard deviation of Leaves' hop count up to the root. The lower the SDHC, the more balanced the tree, with closer leaves' lengths to the root. The FP is computed through Eq. (10), which determines fairness in the energy consumption for all sensors [27]. The DTR requires global information, that is, the residue energy of the sensors, like most hierarchical tree methods. Because the amount of information is low and is carried with data packages or is predicted through BS, the energy consumption for accumulating global information is low. The complexity of the tree formation algorithm run through BS is linear  $O(n)$ . In algorithm 1,  $n$  is the number of edge or non-processed vertices. In algorithm 3,  $n$  is the number of the previous step CHs. Therefore, the complexity of implementing DTR is almost equal to that

of Leach algorithm. The complexity of implementing BATR algorithm is  $O(n^2)$  [6].

### 4-1- The Results of Evaluation

In the BAR, the purpose is to form a minimum spanning tree with balanced leaves in a specific area, where, the energy consumed for data aggregation by the parents is almost equal. To form BATR, the edges weight is estimating of energy consumption to send data and the residue energy of the sensors are not considered. Therefore, if the energy of the sensors is unequal, the energy of some of them run out faster, shown as to FND and LND in Figs. (5 and 6). In DTR spanning tree structure, in addition to residue energy of the sensors the fairness in energy consumption is considered, and if violation of fairness, the structure would change dynamically. Hence, the DTR provides better FND and LND in Figs. (5 and 6).

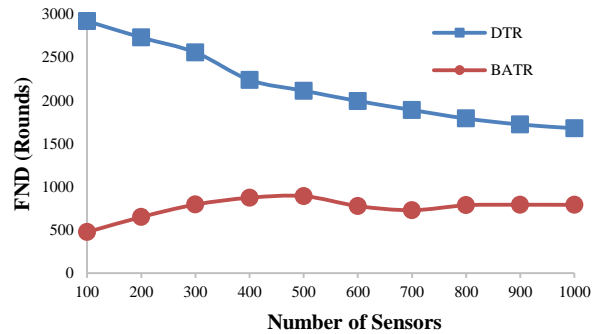


Fig. 5 FND in compared models.

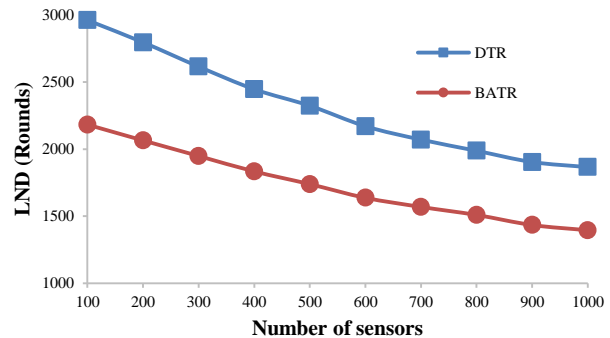


Fig. 6 LND in compared models.

In the DTR, the sensors that accumulate similar data are deployed under the subtree of one parent, thus, the data to be aggregated better by the parents and less energy is consumed to send the data. The less energy consumption between the DTR concerning BATR is shown in Fig. (7).

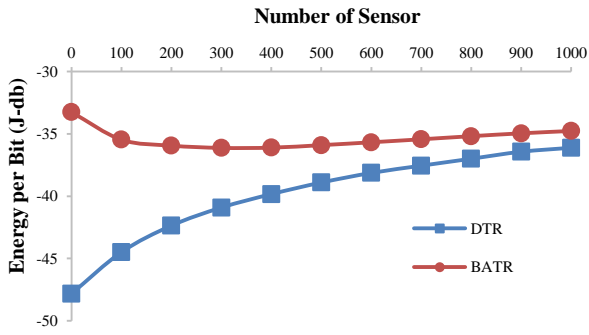


Fig. 7 energy consumed to transfer one bit in the compared methods.

In the DTR, short and long term lifetime of WSN is increased, revealed by the nearness of FND and LND, Fig. (8). Because: 1) less energy consumption to data transmission through clustering with appropriate criterions, and 2) dynamic structure due to justify fairness in energy consuming.

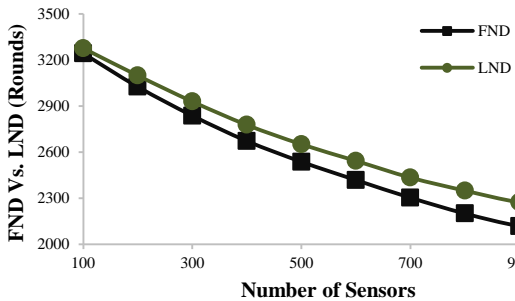


Fig. 8 Comparing FND, LND in DTR.

The comparison of FND and LND in BATR method, reveals that the time difference between the first sensor dysfunction and the whole network dysfunction is high, Fig. (9). This issue leads to premature disconnection of some sensors with the others. Because the residue energy of the sensors is not considered to form minimum spanning tree and energy consumption of the sensors is not fairness, while, in the DTR is contrary.

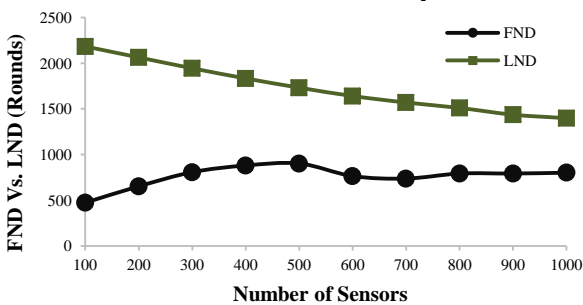


Fig. 9 Comparing FND, LND in BATR.

In BATR and DTR to compare the delay, SDHC is computed at end of each round. As observed in Fig. (10), the BATR standard deviation is lower than that of DTR and this is due to the formation of static balanced-tree. In DTR, the tree is dynamic and changes upon the residue energy, which leads to a change in the leaves' hop count towards the root, thus, more standard deviation into BATR. Because the standard deviation of DTR is close to zero, it can be deduced that delay in DTR is almost moderate and close to the average.

In DTR, the fairness threshold ( $T_h$ ) is 0.9, Table 2. When the  $J_c$  is less than  $T_h$ , the BS releases the control messages for structural changes. As observed in Fig. (11), it is evident that  $J_c$  for different number of nodes is around the  $T_h$ . The slight fluctuation rate around  $T_h$  is due to the energy consumed during the above released order for tree reformation.

In BATR, the constant structure, unequal initial energy, and non-uniform energy consumption by the nodes, causing no fairness in energy consumption, which can become worse as the nodes increase.

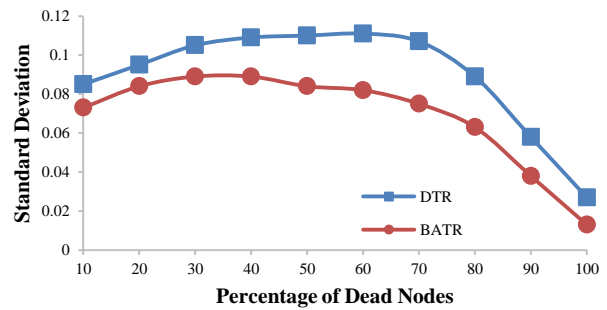


Fig. 10 Comparison of SDHC in the compared models.

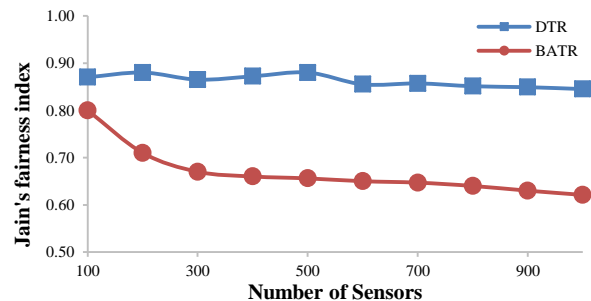


Fig. 11 energy consumption fairness in the compared methods.

### 5- Conclusion and the Future Works

In this paper, a hierarchical spanning tree method is proposed, the strength of which than the available studies are in having a dynamic structure and considering the sensors' dependency during the tree formation. Applying the sensors with the accumulated similar data under one subtree, leads to a reduction in the main load through the non-leaf sensors. These features, next to low implementation cost have reduced the energy consumption, specifically when the initial



energy of the sensors is unequal. The simulation results indicate the short and long term lifetime of the network is increased. Unlike the available methods, DTR is suitable for networks with numerous nodes and IOT. The features of this method are tabulated in Table 3.

In the WSN, the most of the researches have been done based on simulation therefore, the works similar to proposed method could be analytically presented. Applying the machine learning concepts in determining fairness threshold, proper sensors placement and mobile BS can be considered as the future works. Likewise, in the IOT, if every node be considered as an agent, providing a multi-agent would allow most of the IOT challenging issues to be addressed through the multi-agent systems.

Table. 3 The DTR method features

Control Manner	Mobility	Energy Efficiency	Scalability	Delivery Delay	Load Balancing	Algorithm Complexity	Implementation Cost	Global Knowledge
Centralized	No	Yes	Yes	Moderate	Yes	Low	Low	Yes

## References

- [1] M. S. BenSaleh, R. Saida, Y. H. Kacem, and M. Abid, "Wireless Sensor Network Design Methodologies: A Survey," *Journal of Sensors*, vol. 2020, p. 9592836, 2020/01/25 2020.
- [2] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1-48, 2014/04/01 2014.
- [3] X. Liu, "Atypical Hierarchical Routing Protocols for Wireless Sensor Networks: A Review," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5372-5383, 2015.
- [4] L. Chan, K. Gomez Chavez, H. Rudolph, and A. Hourani, "Hierarchical routing protocols for wireless sensor network: a compressive survey," *Wireless Networks*, vol. 26, no. 5, pp. 3291-3314, 2020/07/01 2020.
- [5] M. Ding, X. Cheng, and G. Xue, "Aggregation tree construction in sensor networks," in 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484), 2003, vol. 4, pp. 2168-2172 Vol.4.
- [6] K. Hyun-sook and H. Ki-jun, "A power efficient routing protocol based on balanced tree in wireless sensor networks," in *First International Conference on Distributed Frameworks for Multimedia Applications*, 2005, pp. 138-143.
- [7] H. Ö. Tan and I. Körpeoğlu, "Power efficient data gathering and aggregation in wireless sensor networks," *SIGMOD Rec.*, vol. 32, no. 4, pp. 66-71, 2003.
- [8] W. Qiu, E. Skafidas, and P. Hao, "Enhanced tree routing for wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 3, pp. 638-650, 2009/05/01/ 2009.
- [9] T. Nguyen Duy and V. Nguyen Dinh, "SSTBC: Sleep scheduled and tree-based clustering routing protocol for energy-efficient in wireless sensor networks," in *The 2015 IEEE RIVF International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for Future (RIVF)*, 2015, pp. 180-185.
- [10] Y. Lu and L. D. Xu, "Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2103-2115, 2019.
- [11] M. Kocakulak and I. Butun, "An overview of Wireless Sensor Networks towards internet of things," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, 2017, pp. 1-6.
- [12] M. B. Yassen, S. Aljawaerneh, and R. Abdulraziq, "Secure low energy adaptive clustering hierarchal based on internet of things for wireless sensor network (WSN): Survey," in *2016 International Conference on Engineering & MIS (ICEMIS)*, 2016, pp. 1-9.
- [13] A. Mir and A. Khachane, "Sensing Harmful Gases in Industries Using IOT and WSN," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1-3.
- [14] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-Defined WSN Management System for IoT Applications," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2074-2081, 2018.
- [15] M. M. Afsar and M.-H. Tayarani-N, "Clustering in sensor networks: A literature survey," *Journal of Network and Computer Applications*, vol. 46, pp. 198-226, 2014/11/01/ 2014.
- [16] A. Fahad et al., "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267-279, 2014.
- [17] J. Blanckenstein, J. Klaue, and H. Karl, "A Survey of Low-Power Transceivers and Their Applications," *IEEE Circuits and Systems Magazine*, vol. 15, no. 3, pp. 6-17, 2015.
- [18] M. Abdoos, A. Esmaili, and N. Mozayani, "Holonification of a Network of Agents Based on Graph Theory," Berlin, Heidelberg, 2012, pp. 379-388: Springer Berlin Heidelberg.
- [19] M. Khazaei and N. Mozayani, "Overload management with regard to fairness in session initiation protocol networks by holonic multiagent systems," *International Journal of Network Management*, vol. 27, no. 3, p. e1969, 2017.
- [20] M. Khazaei and N. Mozayani, "A dynamic distributed overload control mechanism in SIP networks with holonic multi-agent systems," *Telecommunication Systems*, vol. 63, no. 3, pp. 437-455, 2016/11/01 2016.
- [21] I. E. Antoniou and E. T. Tsompa, "Statistical Analysis of Weighted Networks," *Discrete Dynamics in Nature and Society*, vol. 2008, p. 375452, 2008/05/06 2008.
- [22] O. Fercoq, "Perron vector optimization applied to search engines," *Applied Numerical Mathematics*, vol. 75, pp. 77-99, 2014/01/01/ 2014.
- [23] A. Esmaili, N. Mozayani, and M. R. J. Motlagh, "Multi-level holonification of multi-agent networks," in *2014 Iranian Conference on Intelligent Systems (ICIS)*, 2014, pp. 1-5.
- [24] J. Wang, J. Liao, T. Li, J. Wang, J. Wang, and Q. Qi, "Probe-based end-to-end overload control for networks of SIP servers," *Journal of Network and Computer Applications*, vol. 41, pp. 114-125, 2014/05/01/ 2014.
- [25] J. Liao, J. Wang, T. Li, J. Wang, J. Wang, and X. Zhu, "A distributed end-to-end overload control mechanism for networks of SIP servers," *Computer Networks*, vol. 56, no. 12, pp. 2847-2868, 2012/08/16/ 2012.
- [26] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660-670, 2002.

- [27] M. Khazaei, "Analytical Model to Create Proxy Server Sessions in Multimedia Networks," *Journal of Information Systems and Telecommunication*, vol. 9, Special Issue, pp. 10-20, 2021.