IET Networks

Research Article

Efficient approach to maximise WSN lifetime using weighted optimum storage-node placement, efficient and energetic wireless recharging, efficient rule-based node rotation and critical-state-data-passing methods

ISSN 2047-4954
Received on 22nd February 2017
Revised 22nd July 2017
Accepted on 15th August 2017
E-First on 16th October 2017
doi: 10.1049/iet-net.2017.0035
www.ietdl.org

Labisha R.V.[1] ✉, *Baburaj E.*[1]
[1]*Department of CSE, Sun College of Engineering and Technology, Nagercoil, Tamil Nadu, India*
✉ *E-mail: labisharv@yahoo.com*

Abstract: The sensor nodes of wireless sensor networks (WSNs) are usually positioned in remote or inaccessible areas and hence the physical maintenance such as battery replacement is more difficult. One of the core challenges of WSN is to increase the network life time, meaning that the efficiency of power utilisation should be the maximum. The major reasons for high power consumption are drawn out transmission path to reach sink, work load add-ons on nodes closer to storage medium, and the energy necessary for storage. This study proposes a novel method of data storage and retrieval for WSN environment to improve the network lifetime by minimising the energy consumption. The proposed method accomplishes this milestone using four novel algorithms, namely weighted optimum storage-node placement(SNP), efficient and energetic wireless recharging, efficient rule-based node rotation and critical-state-data passing. The areas of influence of this proposed method are SNP, wireless recharging, node rotation and cooperative multi-input–multi-output clustering/routing. Simulation results claim that the proposed method multiplies the WSN topology lifetime ratio by a significant level and outperforms the earlier versions significantly.

1 Introduction

Wireless sensor networks (WSNs) are wont to construct pervasive computing applications such as sensing environmental conditions, monitoring people's behaviour over a course of time. These applications generate data continuously which are huge for an epoch. A huge amount of data should be stored somewhere for upcoming retrieval and data analysis. The major challenges of these applications are: an efficient mechanism for data storage, an effective method to retrieve the data (query processing), and bringing efficient methods to stretch the limited lifetime of SN [1]. The assorted data-driven applications of WSN usually require higher-energy consumption due to complex sensors employed in it. Hence, the energy conservation is one of the major issues in WSN-based research to maximise network lifetime. This necessitates inventing new technologies and rational algorithms to effectively utilise the power source, so that the network lifetime could be elongated significantly.

The arrangement of SNs can be done in such a way to reduce the data transmission path between sensor nodes and storage medium. The author Ganesan *et al.* [2] presented a storage-and-search system which describes a lossy, gracefully degrading storage model. The author Ahn and Krishnamachar [3] presented a paper to support energy-efficient storage by providing scaling laws to perform less energy occupied storage. In this paper [1], Sheng *et al.* provided a solution to improve network lifetime based on tree such as architecture, in which a better strategy for positioning the storage devices in an optimised way is addressed. The paper [4] proposes an alternative to external storage based on the estimation of real network distribution for efficient data storage and retrieval by using geometric hash table. The paper [5] presented the concept of reducing the SNs in an optimised way, so that the energy consumption would be minimised. Mane *et al.* [6] described a method for optimal positioning of SNs by concentrating the raw data transfer, query diffusion and query reply and by defining the best location of SNs in an SN.

An alternate technique used to increase the network lifetime is to *recharge* the sensor nodes. The paper [7] used multiple charger nodes to solve the recharging issue of SN and Afzal *et al.* [8] used *modified corner cube retro-reflector* and a device *Thin-film corner cube retro-reflector* for recharging sensor nodes. The paper [9] described the impact of wireless charging technology on SN deployment and routing arrangement. The paper [10] defined a wireless recharging infrastructure using three components such as mobile charger (MC), sensor nodes equipped with wireless power receivers and an energy station (ES). The paper [11] presented a recharging method using actor nodes and power. In the paper [12], an radio-frequency identification-based wireless rechargeable sensor node (WRSN) charged its onboard energy storage above a threshold in order to power its sensing, computation and communication components. The paper [13] explained about the scalability of networks with wireless recharging. The paper [14] suggested a velocity control algorithm for mobile charging in WRSNs. The paper [15] concentrated on joint energy replenishment and scheduling mechanism for recharging to maximise the lifetime of WRSN.

Another significant way to extend the network lifetime is achieved by exchanging the positions of sensor nodes through rotation process. Jain *et al.* [16] provided a lifetime maximisation of WSN by the concept of exploiting mobility for energy-efficient data collection using mobile nodes which placed in the sensor field, as forwarding agents. Luo provided a lifetime elongation in WSNs by using joint mobility-and-routing and the concentration of data traffic toward a small number of base stations (with mobility) [17]. Kansal *et al.* [18] proposed a controlled mobile infrastructure for low-energy embedded networks to reduce the communication energy consumption at the energy constrained nodes. The paper [19] extended WSN lifetime by the huddling behaviour of emperor penguins where the penguins took turns on the cold extremities of a penguin huddle and a mobile node rotation method was suggested for low-cost mobile sensor nodes to increase the WSN lifetime. Daniel *et al.* described a node rotation scheme to extend network lifetime [20].

Appropriate clustering and cluster-head (CH) node election can radically reduce the energy consumption of sensor nodes and a lot

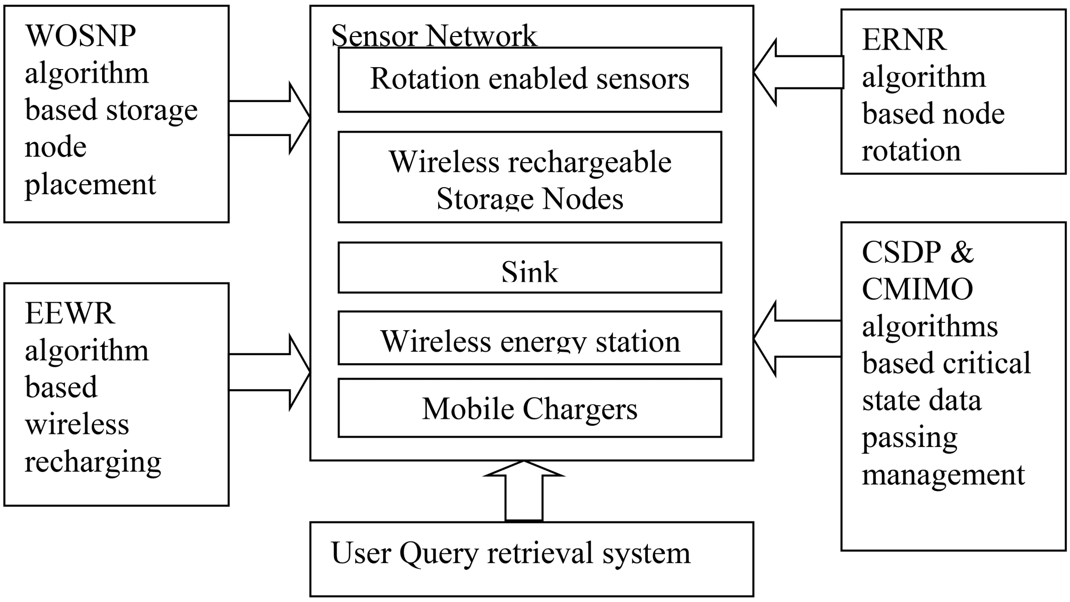

Fig. 1 *Architecture diagram of the proposed data-storage and data-retrieval system in WSN*

of research works are undertaken in this area. The paper [21] presented the famous low energy adaptive clustering hierarchy (LEACH) method for clustering and Liang [22] presented an alternative method for cluster-based lifetime improvement. The paper [23] expressed the cooperative multi-input–multi-output (CMIMO)-based clustering/routing method to extend the lifetime of network by selecting two CHs per cluster. The paper [24] determined the responsible sensing region of each sensor according to the remaining energy using a weighted Voronoi diagram. The paper [25] designed a self-organising and adaptive dynamic clustering (DCMDC) solution to maintain MDC-relay networks.

According with the literature, the existing methods are suffering by insufficient performance with respect to the network lifetime improvement due to their own demerits such as lengthy transmission path, inefficient battery recharging algorithms and inefficient battery power management. The proposed method is aimed to overcome the drawbacks of the existing versions and to elongate the WSN lifetime by applying techniques in the areas of storage-node placement, wireless recharging in sensor nodes, sensor node rotation and critical-state handling (with clustering). The proposed method extends the network lifetime than previous methods by using binary tree such as architecture and four new algorithms/methods. The novel weighted optimum SN placement (WOSNP) method distributes the SNs in optimum positions in the network tree structure to reduce the lengthy storage path and to conserve the energy of sensor nodes. The novel efficient and energetic wireless recharging (EEWR) algorithm is designed to recharge the SNs using the wireless recharging technology. The novel efficient rule-based node rotation (ERNR) algorithm swaps the sensor nodes to boost the network lifetime. The novel critical-state-data-passing (CSDP) algorithm reduces the data loss when the node faces critical state in its data transmission path and in addition the network lifetime is also maintained.

The remainder of this paper is organised as follows: in the next section, the facts of the proposed algorithms are enlightened and the experimental results are discussed and analysed in Section 3 with tables and charts. Finally, Section 4 concludes this paper.

2 Proposed method

This paper proposes a new energy-efficient method for data-storage and data-retrieval system in WSN to extend the network lifetime. Fig. 1 describes the architecture of the proposed method.

The conservation of energy and network lifetime improvement are achieved based on the following four novel methods:

- WOSNP.
- EEWR.
- ERNR.
- CSDP.

2.1 WOSNP algorithm

In the WSN-based data-storage and query-retrieval system, the primary challenge of storing the sensed data efficiently is addressed in such a way that either the collected data are stored in network sensors or the data is transmitted directly to the sink. The approach of data storage at sensors meets failure because of the high production cost of sensor nodes and their memory limitations. The prolonged data storage at sensors drains their battery power quickly and hence this is not viable. The other solution for WSN data storage reduces the battery power drastically as there are huge data transmissions between sensors and the sink. The sensors around the sink too drain their battery in a higher rate as they act as sensing unit as well as data transmission medium and accelerate toward network's death.

The quicker death of sink's neighbours can be avoided by introducing a minimum quantity of sensor nodes with permanent storage medium such as flash memory. These special nodes (called as SNs with optional add-on batteries) are positioned in the network to store the sensed data of the nearby sensors and to reply to users' queries, so that the data-storage process will be done at the nearby nodes and not at the sink. Moreover, the energy saved by the data traversal path count reduction for both sensor nodes and the intermediate nodes improves network lifetime. This idea of introducing SNs addresses the challenges in storage, communication capacity and network lifetime [1].

The existing methods exhibit a poor lifetime of WSN as the placement of SNs is not optimal. Therefore, a new and novel method is needed to minimise the total energy cost in data accumulation and data query by judicious placement of SNs. This paper proposes a new and novel WOSNP method which constructs the network such as a tree model rooted from the sink by placing SNs brilliantly. By and large, sensor nodes are distributed as per the requirements of mine or petroleum pipes, where the dynamic tree model for communication is built after the sensor nodes deployment. Fig. 2 illustrates the working method of the SN-based data storage in which the total network is separated by left-hand side nodes and right-hand side nodes.

Usually, WSN is constructed by several sensor nodes and one sink. The WSN can be represented by a set Z

$$\begin{aligned} Z &= \{\{\mathrm{SSN}\}, \mathrm{SNK}\} \\ N &= k + 1 \end{aligned} \qquad (1)$$

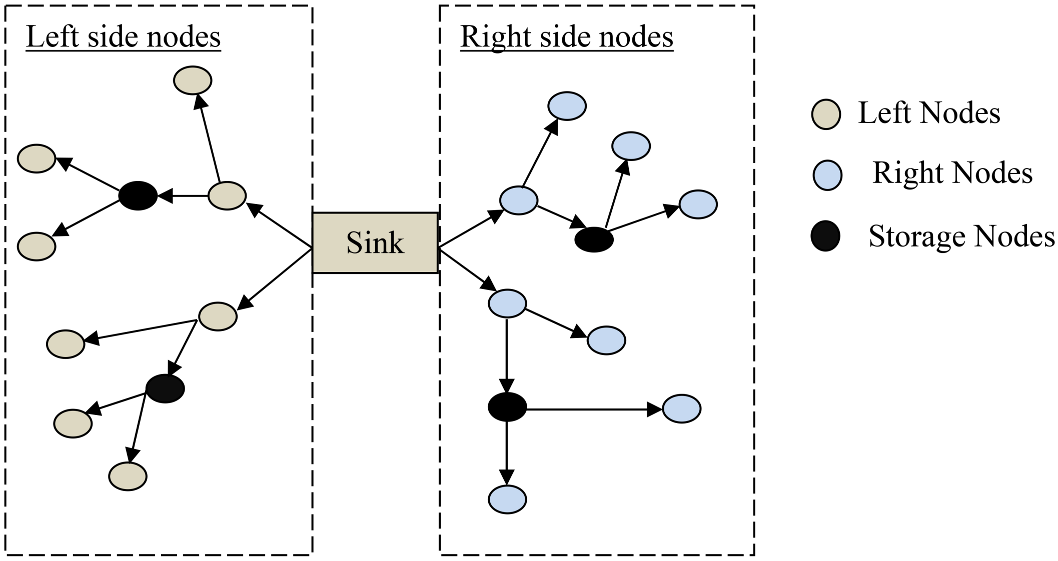

Fig. 2 *Data-storage model by SN placement*

$$\begin{aligned} &\mathrm{SSN} = \{\mathrm{SS}_1, \mathrm{SS}_2, \mathrm{SS}_3, \ldots, \mathrm{SS}_k\} \\ &\mathrm{SSN} \in Z \end{aligned} \tag{2}$$

where SSN is the sensor nodes set; SS is the Sensor node; SNK is the sink node; Z is the WSN elements set; k is the sensor node count; and N is the wireless network elements count.

In the SN, k sensor nodes are assumed and a sink is placed in the middle of the network. These constraints are presented in (1) and (2). The total network is divided as two parts and the elements are coded as Z_{LEFT} as left-hand side elements and Z_{RIGHT} as right-hand side elements and both are collectively represented in (3)

$$\mathrm{SS}_m \in \begin{cases} Z_{\mathrm{LEFT}} & \text{if} \quad \mathrm{SS}_m^x < \mathrm{SNK}^x \\ Z_{\mathrm{RIGHT}} & \text{otherwise} \end{cases} \tag{3}$$

$\mathrm{SS}_m \in \mathrm{SSN}$ and $m \in [1, k]$, where SS^x is the horizontal x location of the sensor node and SNK^x is the horizontal x location of the sink.

Then binary tree-like structure is constructed individually for both left-hand and right-hand sides and the nodes are positioned in a way that at least one node should be in conduct for communication. Some of the sensor nodes are designated as SNs and the others are forwarding nodes. The total network Z can be represented by (4)–(12)

$$Z = \{\{Z_{\mathrm{LEFT}}\}, \{Z_{\mathrm{RIGHT}}\}, \mathrm{SNK}\} \tag{4}$$

$$Z_{\mathrm{LEFT}} = \{\{\mathrm{FSR}_{\mathrm{LEFT}}\}\ \{\mathrm{ST}_{\mathrm{LEFT}}\}\} \tag{5}$$

$$Z_{\mathrm{RIGHT}} = \{\{\mathrm{FSR}_{\mathrm{RIGHT}}\}\ \{\mathrm{ST}_{\mathrm{RIGHT}}\}\} \tag{6}$$

$$\mathrm{FSR}_{\mathrm{LEFT}} = \left\{\mathrm{FSR}_{\mathrm{LEFT}}^1, \mathrm{FSR}_{\mathrm{LEFT}}^2, \mathrm{FSR}_{\mathrm{LEFT}}^3, \ldots, \mathrm{FSR}_{\mathrm{LEFT}}^m\right\} \tag{7}$$

$$\mathrm{FSR}_{\mathrm{RIGHT}} = \left\{\mathrm{FSR}_{\mathrm{RIGHT}}^1, \mathrm{FSR}_{\mathrm{RIGHT}}^2, \mathrm{FSR}_{\mathrm{RIGHT}}^3, \ldots, \mathrm{FSR}_{\mathrm{RIGHT}}^n\right\} \tag{8}$$

$$\mathrm{ST}_{\mathrm{LEFT}} = \left\{\mathrm{ST}_{\mathrm{LEFT}}^1, \mathrm{ST}_{\mathrm{LEFT}}^2, \mathrm{ST}_{\mathrm{LEFT}}^3, \ldots, \mathrm{ST}_{\mathrm{LEFT}}^p\right\} \tag{9}$$

$$\mathrm{ST}_{\mathrm{RIGHT}} = \left\{\mathrm{ST}_{\mathrm{RIGHT}}^1, \mathrm{ST}_{\mathrm{RIGHT}}^2, \mathrm{ST}_{\mathrm{RIGHT}}^3, \ldots, \mathrm{ST}_{\mathrm{RIGHT}}^q\right\} \tag{10}$$

$$M = m + n \tag{11}$$

$$P = p + q \tag{12}$$

where $\{\mathrm{FSR}_{\mathrm{LEFT}}\}$ is the forward sensor node set at left-hand side, $\{\mathrm{FSR}_{\mathrm{RIGHT}}\}$ is the forward sensor node set at right-hand side, $\{\mathrm{ST}_{\mathrm{LEFT}}\}$ is the SN set at left-hand side, $\{\mathrm{ST}_{\mathrm{RIGHT}}\}$ is the SN set at right-hand side, m is the count of left-hand side forward sensor nodes, n is the count of right-hand side forward sensor nodes, p is the count of left-hand side SNs, q is the count of right-hand side SNs, M is the total forward sensor nodes, and P is the total SNs.

The proposed method of positioning the SNs works based on the weight values of four properties of each sensor node. Suppose, P nodes are decided to be placed as SNs then, the nodes are placed alternatively on both left-hand and right-hand sides till there is no position to place the nodes and the equilibrium is maintained in this process. The four properties are listed below:

- Neighbour-based SN eligibility (ELIGIBILITY).
- Storage path length (SPL).
- Descendent nodes count (DCs).
- Siblings occurrence (SO).

The architecture diagram of the proposed WOSNP method is depicted in Fig. 3. According to Fig. 3, WSN is separated into left-hand and right-hand side nodes and the properties SPL, ELIGIBILITY, DC and SO are computed to place the SNs optimally. These four properties and the weight value assignments are explained below.

2.1.1 Neighbour-based SN eligibility (ELIGIBILITY): The eligibility constraint is described as two successive nodes cannot be SNs. Thus, if a node X is designated as an SN, then none of its adjacent nodes can be an SN. Consecutive SNs accumulate the cost which is unwanted.

The weight value assignments for left-hand side nodes are performed using (13)

$$W_{\mathrm{ELIG}}^X = \begin{cases} 0 & \text{if} \quad \mathrm{Parent\_Node}(X) \in \mathrm{ST}_{\mathrm{LEFT}} \\ 0 & \text{else if} \quad \mathrm{ChildNode}(X) \in \mathrm{ST}_{\mathrm{LEFT}} \\ 0 & \text{else if} \quad X \in \mathrm{ST}_{\mathrm{LEFT}} \\ 1 & \text{else} \end{cases} \tag{13}$$

where W_{ELIG}^X is the eligibility-based weight value, X is the sensor node to be checked for eligibility for SN, Parent_Node(X) is the parent node of the node X and Child_Node(X) is the child node of the node X.

Finally, the neighbour-based eligibility property returns 0 for non-eligible and 1 for eligible as an SN.

2.1.2 Storage PL: The SPL property for a sensor node Y can be calculated using the distance between node Y and the nearest

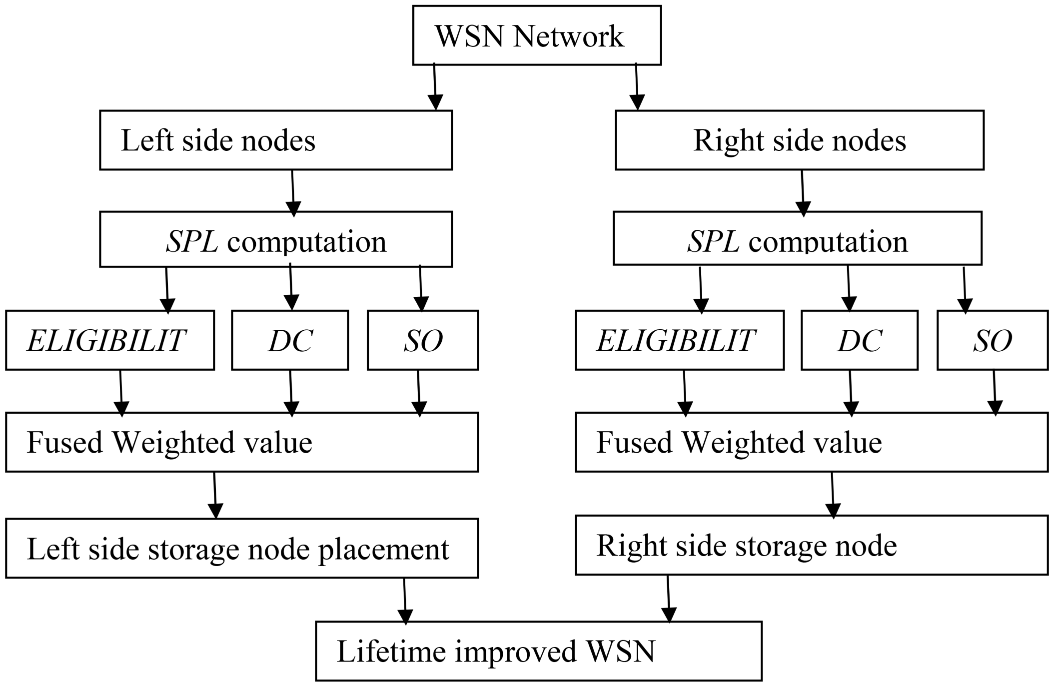

Fig. 3 *Block diagram of the proposed WOSNP method*

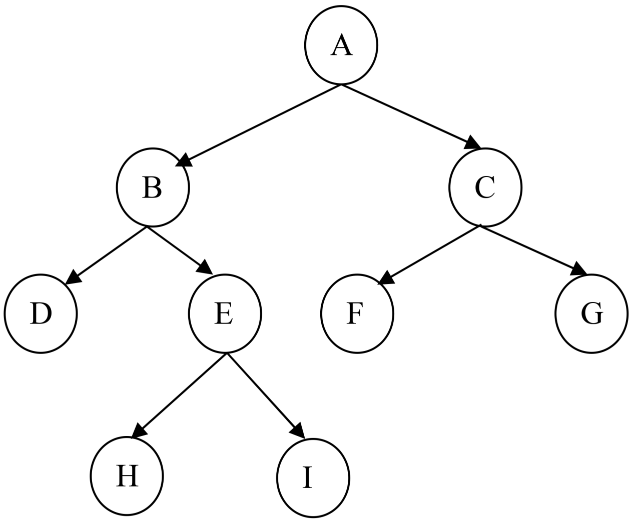

Fig. 4 *Descendent nodes of a sample binary tree*

storage medium which could be a node or the sink. The *SPL* property computation is performed based on (14) and (15)

$$\mathrm{ST_{medium}} = \{\{\mathrm{ST_{LEFT}}\}, \mathrm{SNK}\} \tag{14}$$

$$\mathrm{SPL}^{Y} = \mathrm{PathCount}(Y, \mathrm{Nearest\_Of\_ST_{medium}}) \tag{15}$$

where Y is any sensor node, $\mathrm{ST_{medium}}$ is a set which contains allocated SNs and sink, SPL^{Y} is the SPL for the Y sensor node, PathCount is the function to compute PL of Y and the nearest $\mathrm{ST_{medium}}$.

The weight value assignment procedure for the SPL property is described in (16) and (17)

$$\begin{aligned} &\mathrm{MAX_{SPL}} = \mathrm{Max}(\mathrm{SPL}^{i}) \\ &i \quad \in \{\mathrm{SR_{LEFT}}\} \end{aligned} \tag{16}$$

$$W_{\mathrm{SPL}}^{Y} = \begin{cases} w_{\mathrm{SPC}} & \text{if} \quad \mathrm{SPL}^{Y} = \mathrm{MAX_{SPL}} \\ 0 & \text{else} \end{cases} \tag{17}$$

where SPL^{i} is the SPL for each sensor nodes, Max is the function to find the maximum value, $\mathrm{MAX_{SPL}}$ is the maximum SPL value and w_{SPC} is the weight constant for SPL computation. It can be assigned as 0.60. W_{SPL}^{Y} is the weighted computation of SPL for sensor node Y.

The weight value of 0.60 is assigned to W_{SPL}^{Y} if the computed SPL for a given sensor node is matched with the maximum SPL of the left-hand side; otherwise, it is assigned to 0. The SPL is the heavy dominated property than DC or SO and hence the higher priority constant 0.60 is assigned as the weight value to SPL. The network lifetime is significantly affected by the length of storage paths. So, if an SN is positioned in between this lengthy transmission path, the power usage will be significantly reduced for data storage.

2.1.3 Descendent node count: Descendent nodes refer to the children of a parent node. Fig. 4 illustrates the descendent nodes of a binary tree rooted from node A. All the nodes except A are A's descendent nodes, nodes D, E, H and I are the descendent nodes of B and so on.

The DC property computation needs two parameters as input, namely a sensor node to be proposed as SN (say X), a set of sensor nodes $\{s\}$ that are associated with high SPL for a 'maximum value computation'. The DC computation is performed using (18)

$$\mathrm{DC}^{X} = \mathrm{func\_DC}(X) \tag{18}$$

where DC^{X} is the computed value of DC property for the node X andfunc_DC is the function to compute DC.

The weight value of DC property is computed using (20)

$$\begin{aligned} &\mathrm{MAX} = \mathrm{Max}(\mathrm{DC}^{i}) \\ &i \quad \in \quad \{s\} \end{aligned} \tag{19}$$

$$W_{\mathrm{DC}}^{X} = \begin{cases} w_{\mathrm{dc}} & \text{if} \quad \mathrm{DC}^{X} = \mathrm{MAX} \\ 0 & \text{else} \end{cases} \tag{20}$$

where MAX is the maximum DC value among DCs of given set $\{s\}$, DC^{i} is the DC value of each and every element of the set $\{s\}$ and w_{dc} is the weight constant for DC computation. It can be

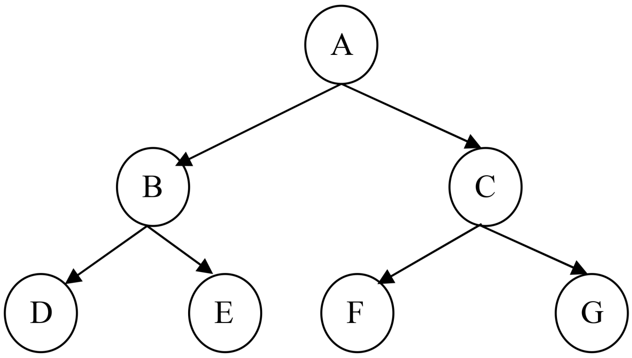

Fig. 5 *Illustration of sibling nodes*

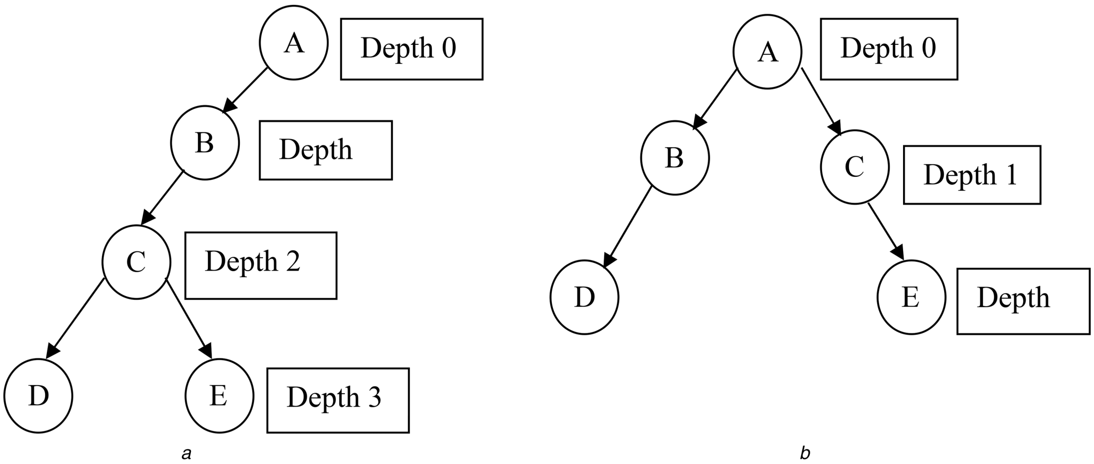

Fig. 6 *Illustration of domination of sibling property*
(a) Tree with descendent node count four and non-siblings property, ***(b)*** Tree with DC four and sibling property

assigned as 0.30. $W_{\rm DC}^{X}$ is the weighted computation of DC for X sensor node.

The DC is the second dominative property to find the SN position priority and its weight value is assigned as 0.30. The network lifetime is significantly reduced when more descendents are placed under a sensor node. If an SN is positioned amongst the densely populated sensor nodes, the power requirements will be reduced as the lengthy data transmission path is avoided.

2.1.4 Sibling occurrence: In binary tree, the terminology *sibling* is mentioned as the two nodes that are children of the same parent and this SO property is computed for positioning of SN. This sibling terminology is illustrated in Fig. 5. In Fig. 5, the B&C, D&E and F&G node pairs are sibling nodes. The node pair B&C is formed from a single parent A and the node pair D&E is formed from a single parent B. The node parent F&G is formed from a single parent C. In this paper, a parent node is checked against the property of sibling in their child nodes.

If a node X is proposed for an SN, it should be checked whether it has siblings. Fig. 6 expresses the two cases of tree structure.

The DC of root A is 4 in both Figs. 6*a* and *b*. However, the tree depth varies as 3 for Figs. 6*a* and 2 for Fig. 6*b*. The root A does not bring forth sibling property in Fig. 6*a* but in Fig. 6*b*, it sires siblings with B&C. The child nodes D&E drop more energy in Fig. 6*a* than in Fig. 6*b* since the passing path count for node D is 3 in Fig. 6*a*, whereas the passing path count is 2 in Fig. 6*b*. In Fig. 6*b*, the battery energy drop is reduced to reach node D as the passing path count is reduced. From this illustration, it can be concluded that the parent node with non-sibling property decreases the network lifetime when compared with the parent node with siblings for same DC. In general, single child parents are more suitable than two children parents for SN placement to increase the network lifetime. The node A of Fig. 6*a* is better suitable for SN placement and the SO computation is done using (21)

$$\mathrm{SIBL}^{X} = \begin{cases} 1 & \text{if} \quad \text{Left\_child\_node\_available\_for\_}X\text{\_ and} \\ & \quad \text{Right\_child\_node\_not\_available\_for\_}X \\ 1 & \text{else} \quad \text{if Right\_child\_node\_available\_for\_}X\text{\_ and} \\ & \quad \text{Left\_child\_node\_not\_available\_for\_}X \\ 1 & \text{else} \quad \text{if else if Left\_child\_node\_available\_for\_}X\text{\_ and} \\ & \quad \text{Right\_child\_node\_available\_for\_}X \\ 0 & \text{else} \end{cases} \tag{21}$$

The weighted computation of sibling property for the node X is computed using (22)

$$W_{\rm SIBL}^{X} = \begin{cases} w_{\rm sibl} & \text{if} \quad \mathrm{SIBL}^{X} = 1 \\ 0 & \text{else} \end{cases} \tag{22}$$

where $W_{\rm SIBL}^{X}$ is the weighted value from sibling property for node X and $w_{\rm sibl}$ is the weight constant for sibling computation. It can be assigned as 0.10.

The sibling behaviour is the third dominating criteria of affecting the network lifetime and to schedule its dominating range the weight constant $w_{\rm sibl}$ is settled as 0.10. Over all, the single child node parent gets more privilege than the two children node parents. In this way, the four properties such as ELIGIBILITY, SPL, DC and SO are computed and the fusion of these four properties is used to compute the priority level for SN placement.

The SPL property is computed for entire forward sensor nodes of left-hand side and if only one storage path produces the weighted SPL W_{SPL}^{Y} as 0.6, then the middle node of the storage path is planned to settle as the SN. If the ELIGIBILITY property value is also 1, then that node is permanently settled as SN; otherwise, not. In some cases, there is a chance for multi-instances of same length storage paths available, i.e. multiple sensor nodes are returning the W_{SPL}^{Y} value as 0.6. In this case, the multi-instances of storage paths are collected as a set $\{Z_{\mathrm{SPL}}\}$. The first storage path is taken and the middle element of the storage path is considered as the node X and it is checked to the possibility of placement as SN through the DC property computation. The DC computation uses the node X and the sensor node which are taken from the set $\{Z_{\mathrm{SPL}}\}$. Finally, the W_{DC}^{X} value for the node X is computed and in this way the other storage paths lying in the set $\{Z_{\mathrm{SPL}}\}$ are also computed their W_{DC} value. The multi-instances of Max-length storage paths from the set $\{Z_{\mathrm{SPL}}\}$ are taken one by one and the middle element is assumed as dummy SN X. Then, the dummy SN X is checked for the sibling property and the weighted sibling value W_{SIBL}^{X} is computed. If the W_{SIBL}^{X} value is settled by 0.10, then that node X contains one child node; otherwise, it contains two children nodes. The remaining storage paths from $\{Z_{\mathrm{SPL}}\}$ are also considered and the W_{SIBL}^{X} value is computed. The ELIGIBILITY property is checked and the W_{ELIG} value is obtained for each dummy SN X. The fused weight value for each storage path of $\{Z_{\mathrm{SPL}}\}$ is formed using (23)

$$W_{\mathrm{Fused}} = W_{\mathrm{ELIG}} * (W_{\mathrm{SPL}} + W_{\mathrm{DC}} + W_{\mathrm{SIBL}}) \tag{23}$$

where W_{Fused} is the final weighted value to compute the priority to settle SN in the middle of storage path set $\{Z_{\mathrm{SPL}}\}$.

The higher W_{Fused} value producing storage path from set $\{Z_{\mathrm{SPL}}\}$ is found and the middle node of that path is settled as real SN. The new SPL, DC and SO computations are performed iteratively. On the basis of these results, the fusion process is performed and the new SN is located at that middle element of selected storage path.

Fig. 7 explains how the SNs are positioned based on the proposed WOSNP method. In this example, four SNs are decided to be placed in the total network. The SPL for each left-hand side sensor node is computed and the W_{SPL}^{9} value is returned as 0.60 because the node 9 provides a storage path in 10 unit length. There is only one *MaxStoragePath* instance occurred in this case. The storage path formed by the set of nodes $\{9, 8, 7, 6, 5, 4, 3, 2, 1, \mathrm{sink}\}$ is selected for SN placement. The middle element 5 is chosen as dummy SN and the ELIGIBILITY property is checked for this node. The ELIGIBILITY property gives 1 because there are no other neighbour SNs. So, the node 5 is fixed as SN permanently. In Fig. 3*b*, the node 5 is marked as SN. In the next iteration, there are three storage paths constituting *MaxStoragePath* length and the storage path 1 is generated by the set of nodes $\{16, 15, 14, 13, 12, 11, 10, 1, \mathrm{sink}\}$. In this case, the SPL is 8. The storage path 2 is generated by the set of nodes $\{24, 23, 22, 21, 20, 19, 18, 17, \mathrm{sink}\}$ and the SPL is 8. The storage path 3 is generated by the set of nodes $\{31, 20, 29, 28, 27, 26, 25, 17, \mathrm{sink}\}$ and the SPL is 8. In these three storage paths, the lengths of the storage paths are maintained as same and three storage paths holds the W_{SPL} value as 0.60. The DC property is computed for the storage path 1 using the middle dummy SN 12 and the three storage path information. There are 6 descendent nodes under the node 12. According to (20), the W_{DC} value is settled as 0. The sibling property is computed for the node 12. The sibling property is assigned by 0.10 because it has only one child node. The eligibility of the node 12 is computed and it is assigned as 1 because there are no other SNs nearby it. The fused value of the storage path 1 is computed using (23) and the computation is such as $1.0 \times (0.60 + 0.0 + 0.10) = 0.7$. The fused value of storage path 2 is computed by $1.0 \times (0.60 + 0.30 + 0.10) = 1.0$. The fused value of storage path 3 is computed by $1.0 \times (0.60 + 0.30 + 0.0) = 0.90$. Among the three storage paths, the storage path 2 gains the fused value 1. This is the higher fused value among these three storage paths. Then, the middle node 20 on the storage path 2 is selected as the position of SN and the SN is positioned in it permanently. The storage path 1 loses its priority, because it has less DC property value. The storage path 3 loses its priority because the sibling behaviour is settled in the child nodes of node 27.

The next iteration is started and two storage paths are selected for the SN positioning competition. They are

$$\{16, 15, 14, 13, 12, 11, 10, 1, \mathrm{sink}\}$$
$$\{31, 30, 29, 28, 27, 26, 25, 17, \mathrm{sink}\}$$

The fusion value for storage path 1 is computed as $1.0\times(0.60 + 0.0 + 0.10) = 0.7$ and the fused value for storage path 2 is computed as $1.0\times(0.60 + 0.30 + 0.0) = 0.90$. So, the middle node 27 of the storage path 2 is assigned as SN. In the next iteration, there is only one storage path with the length 8 and the ELIGIBILITY property for the middle node 12 is also settled as 1. So, the middle node 12 is settled as SN (Fig. 8).

2.2 EEWR method

Battery powered sensor nodes networks have deployment challenges due to limitations of their own power source. The interior networks such as the one used in coal mine or petroleum tube monitoring do not possess the possibility of energy harvesting but the wireless recharging technique provides a novel, unique and reliable way to provide power for the interior sensors. The wireless recharging technology can be used to recharge sensor devices in distance without wires or plugs. The paper [26] explains that energy can be transferred between magnetically coupled coils in excess of 2 m. The mobile vehicles equipped with resonant coils and high-density battery packs can move toward nodes in very close distance and distribute wireless energy to the nodes with high efficiency. The wireless recharging technique can be adopted in consumer electronics, health care and electrical vehicles.

The three components such as MC, sensor nodes with wireless power receivers and an ES together contribute to the wireless recharging system. The MC is a mobile robot which carries a wireless power charger for sensor nodes. The ES monitors the energy status of the network and directs the MC to charge sensor nodes [10]. This is a more cost expensive system because of the manufacturing of all sensor nodes with wireless power receiver.

A novel EEWR algorithm is developed in this paper to control and link MCs, ES and the sink for recharging on WSN network in an optimal way. By and large, SNs lose their energy quicker than the normal sensor nodes. The proposed storage and query-retrieval system which are equipped with SNs is planned to handle only the designated SNs with wireless recharging facility, so that their lifetime is elongated to a greater extent and at the same time this facility is affordable since only a limited quantity of nodes are used for wireless recharging.

Fig. 9 explains how the MCs and ES are positioned. One ES and two MCs are used here, for example. The MC_1 and MC_2 are recharging the left-hand and right-hand side nodes, respectively. The SNs which possess recharge facility bear the power receiver antennas and they communicate their energy status to the sink periodically. The sink then sends this report to the ES which is seated nearby the sink to maintain common distance for both side nodes and the ES interprets the information from sink and executes the algorithm for recharging to ignite charging activities and then the command messages are sent to the specified MC. On receiving the command messages, the MC makes mobility as per the instructions (on how to handle shortest path to reach the nodes) [9, 10, 13] and charges the specific SN. For detailed configuration of MCs, ES and power receivers, see paper [10] (Fig. 10).

The EEWR algorithm is designed to complete the following five criteria:

- If the MC keeps on moving, its energy gets exhausted quickly and hence it must recharge the critical-stage SN at least for s seconds. This time is known as *CriticalServiceTime*. On completion of this, it can recharge any SN.

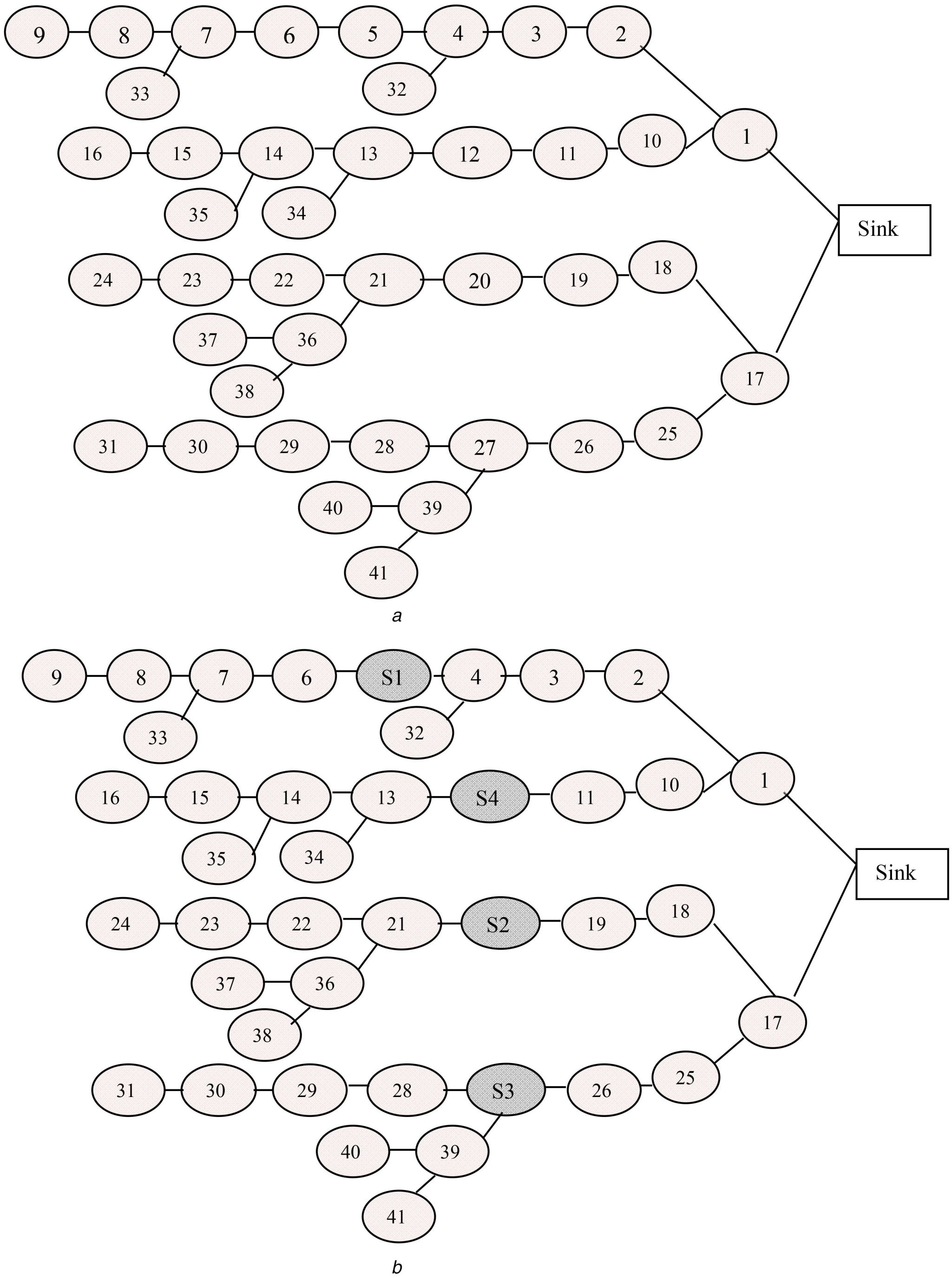

Fig. 7 *Illustration of the proposed WOSNP method*
(a) WSN before storage placement, ***(b)*** WSN after SN placement

- To reduce the energy consumption of MC, the duration for the recharge of *NonCriticalNode* is set as 2 s at least, which is known as *BasicServiceTime.*
- The EEWR algorithm is applied for left-hand side SNs by the ES using the energy report of left-hand side SNs.
- The EEWR algorithm is applied for right-hand side SNs by the ES using the energy report of right-hand side SNs.
- Both left-hand and right-hand side recharging is performed simultaneously.

The SNs periodically pass their energy requests to sink in predefined interval *s*. The sink resends those requests to ES and kept in ES's queue. The ES serves all the entries of its queue. The critical energy status of an SN is confirmed when it meets 10% average storage energy. The types of energy requests of an SN are *CriticalEnergyStatus* and *NonCriticalEnergyStatus*.

The ES checks for *CriticalEnergyStatus* of SNs in its queue. Suppose a *CriticalEnergyStatus* of an SN (say the *k*th *node*) is found, it checks weather the *CurrentAllocation* is equal to *CriticalType*. If it is true, the ES completes the recharging of currently recharging critical node for *CriticalServiceTime* and sends commands to corresponding MC to recharge the *k*th SN. If *CurrentAllocation* is equal to *NonCriticalType* the ES sends commands to corresponding MC to recharge the *k*th SN.

The ES searches for any *CriticalEnergyStatus* in its queue and if it finds false match, it checks for the trueness of *CurrentAllocation* = *CriticalType.* If it is true, the ES completes the recharging of currently recharging critical node for *CriticalServiceTime* and after completion it recharges the same node for *AdditionalTime* because there is none of urgent recharge. Then, the ES finds the *MinimumEnergyStorageNode* (say *r*) from energy status report and it signals the corresponding MC to recharge the SN *r* for *BasicServiceTime.* Here, *BasicServiceTime* is considered because there is no urgent recharge request in the queue.

The ES makes verification for CurrentAllocation is equal to *NonCriticalType* and availability of any *CriticalEnergyStatus* is false. If the output is true, then it detects *MinimumEnergyStorageNode* from queue. This SN can be called as updated *MinimumEnergyStorageNode.* This node may be differing from the old *MinimumEnergyStorageNode.* If they are different, the ES completes the recharge of old node for *BasicServiceTime* and after that it sends recharging commands to MC to recharge updated node. Suppose the updated node and old node are same, the additional *BasicServiceTime* recharge is allocated to old node (or updated node).

Input: WSN network, storage node count u.
Output: WSN network with storage node placement to improve network lifetime.
Step_1: DO network separation as Z_{LEFT} and Z_{RIGHT}.
SET *Positioned_Storage_Node_Count=0*
Step_2: DO *SPL* $\forall\ Z_{LEFT}$.
Step_3: DO *Max_length_storage_path* computation
Step_4: IF *MAX_Len_Storage_Path_Count*=1
DO *ELIGIBILITY computation*
IF W_{ELIG}=1
SET *Storage_node=Middle_node*
END
ELSE
FOR EACH *instance_of_multi_path*
DO *DC* computation
DO *SO computation*
DO W_{FUSED} computation
END
SET *Storage_node=*
Middle_node_of_*Max*(W_{FUSED})_given_path
END
Set *Positioned_Storage_Node_Count=1*
Step_5: REPEAT steps 3 to 4 for Z_{RIGHT} elements.
SET *Positioned_Storage_Node_Count=*
Positioned_Storage_Node_Count+1
Step_6: DO Iterative_process
IF *Left_Side_Storage_Node_Placement =Possible*
Storage_node=
Middle of selected_path using steps 2,3 and 4.
Positioned_Storage_Node_Count=
Positioned_Storage_Node_Count+1
END
IF *Right_Side_Storage_Node_Placement=*
Possible
Storage_node=Middle of selected_path using steps 2,3 and 4.
Positioned_Storage_Node_Count=
Positioned_Storage_Node_Count+1
END
WHILE *Positioned_Storage_Node_Count<u*

Fig. 8 *WOSNP algorithm*

The ES may face the empty queue problem, and to solve that the status of MC is set as idle. The proposed EEWR algorithm performs an intelligent recharging at the situations of *CriticalType* and *NonCriticalType* energy statuses of SNs. This algorithm concentrates both the energy issues of MCs and SNs.

2.3 ERNR algorithm

One of the prominent issues met by WSNs is to manage the power consumption of nodes for a better lifetime of the network as sensors are energy hungry devices. In arctic regions, the temperature may vary from −50° and the wind may blow 180 km/h. To tackle this situation, the emperor penguins form tightly packed huddles and make swapping actions at a speed of 12 cm/s, so that all members of the huddle can get a chance to warm up and this technology can be used to save the lifetime of sensor nodes in WSN [20]. The mobile sensor swapping approach is the technique that a wireless sensor exchanges position of itself with less-energy-consumption sensor and is one of the key solutions to improve the network lifetime [19, 20] (Fig. 11).

This paper proposes a new and novel ERNR algorithm to sensor node rotation for the rechargeable-SN integrated WSN network in contrast to the EEWR method which is used for SNs not for the sensor nodes. Here, SNs do not make any movement. The ERNR algorithm is applied individually for sensor nodes on both sides.

The ERNR algorithm makes rotation with sensor nodes using three parameters such as *Energy*, *Distance* and *Rotation count*. Here, *Energy* means the battery energy of sensor node and *Distance* means distance between *CriticalEnergyNode* and related node. The term rotation means the swapping process of lower-energy nodes with higher-energy nodes. The rotation process of a node may consider the related nodes such as parent node NODE_i^P, left leaf node $\mathrm{NODE}_i^{\mathrm{LL}}$ and right leaf node $\mathrm{NODE}_i^{\mathrm{LR}}$. The *Rotation count* means total rotations made by the specified node.

The NODE_i is considered as *CriticalEnergyNode* using (24) and (25). The NODE_i is marked as *CriticalEnergyNode* (by setting $CN_i = 1$) when its energy is less than the mean energy E_{MEAN} of left-hand side sensors. If the NODE_i is categorised as *CriticalEnergyNode*, then only the rotation process is continued; otherwise, there is no need of node rotations. If the NODE_i is *CriticalEnergyNode* and the parent node NODE_i^P is not belonging

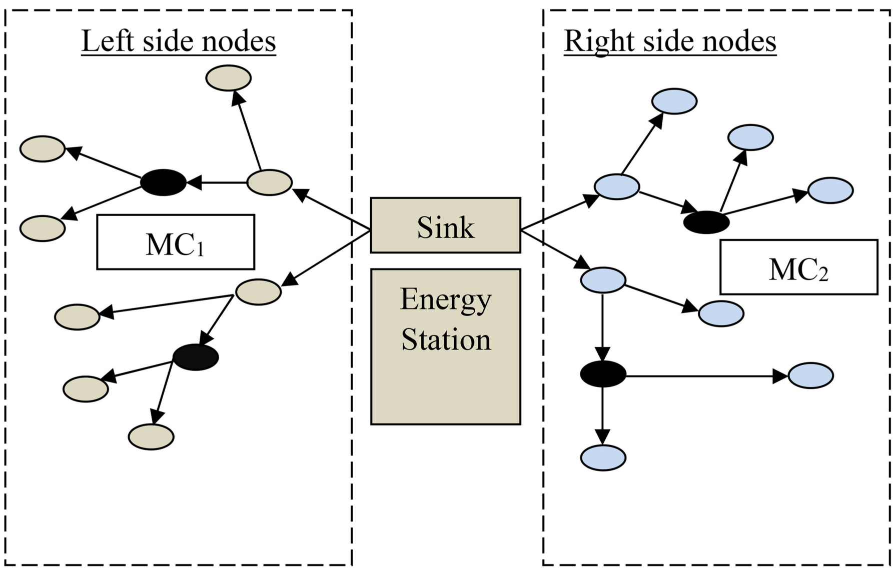

Fig. 9 *Diagram for placement of MCs and ES*

```
Input: Binary tree type WSN with storage nodes SNs, Energy status report, Energy Station, MC1 and MC2
Output: Lifetime improved storage nodes and network
Step_1: SET CriticalServiceTime = s secs
Step_2: SET BasicServiceTime = 2s secs
        SET PeriodicalTime = s secs
        SET AdditionalTime = s secs
        SET CriticalEnergy =10%_of_average_storage_node_Energy
Step_3: DO WHILE  PeriodicalTime=true
            IF CurrentAllocation=CriticalType and IsAnyCriticalEnergyStatusInQueue=true
                DO Check completion of CriticalServiceTime for current_recharging_node
                IF not completed then
                    Wait for completion of CriticalServiceTime
                    After completion
                        SET CurrentAllocation =CriticalType
                        Recharge next coming Critical_stage_node
                ELSE
                    SET CurrentAllocation=CriticalType
                    Recharge next coming Critical_stage_node
                END
            END
            IF CurrentAllocation=NonCriticalType and IsAnyCriticalEnergyStatusInQueue=true
                SET CurrentAllocation=CriticalType
                Recharge next coming Critical_stage_node immediately
            END
            IF CurrentAllocation=CriticalType and IsAnyCriticalEnergyStatusInQueue=false
                Wait for completion of CriticalServiceTime
                After completion
                    Allocate AdditionalTime to currently_recharging_critical_node
                    After completion of  AdditionalTime
                        SET CurrentAllocation=NonCriticalType
                        Find MinimumEnergyStorageNode from energy_status_report
                        Recharge  MinimumEnergyStorageNode for BasicServiceTime
            END
            IF CurrentAllocation=NonCriticalType and IsAnyCriticalEnergyStatusInQueue=false
                Find Updated_MinimumEnergyStorageNode from the energy status report
                IF  Old_ MinimumEnergyStorageNode <>Updated_ MinimumEnergyStorageNode
                    Wait for completion_of_ BasicServiceTime_criteria
                    After completion
                        SET CurrentAllocation=NonCriticalType
                        Allocate BasicServiceTime to
                                        Updated_MinimumEnergyStorageNode
                ELSE    IF  Old_ MinimumEnergyStorageNode =
                                        Updated_MinimumEnergyStorageNode
                        SET CurrentAllocation=NonCriticalType
                        Allocate BasicServiceTime to
                            Updated_MinimumEnergyStorageNode (or
                            Old_MinimumEnergyStorageNode)
                    ELSE
                        Set MC_Status=Idle
                    END
                END
            END
        END WHILE
```

Fig. 10 *EEWR algorithm*

to any SN or sink $\mathrm{NODE}^{\mathrm{SINK}}$, then NODE_i is swapped with the highest energy node among NODE_i^P, $\mathrm{NODE}_i^{\mathrm{LL}}$ and $\mathrm{NODE}_i^{\mathrm{LR}}$.

If NODE_i is not belonging to any SN or sink and the three energies E_i^P, E_i^{LL} and E_i^{LR} are same, then the NODE_i is swapped with the minimum distance node $\mathrm{NODE}_{\mathrm{MIN}}^{\mathrm{DIST}}$ using the *Distance* parameters D_i^P, D_i^{LL} and D_i^{LR}. If NODE_i is not belonging to any SN or sink and the three *Energies* are same and the three *Distances* are same then the *MinimumRotationCount* node $\mathrm{NODE}_{\mathrm{MIN}}^{\mathrm{ROT}}$ is used for the swapping of NODE_i.

If the parent node NODE_i^P belongs to any one of SN or sink, then NODE_i is swapped with the *HighestEnergy* node among left leaf node $\mathrm{NODE}_i^{\mathrm{LL}}$ and right leaf node $\mathrm{NODE}_i^{\mathrm{LR}}$. If the energies of left and right leaf nodes are same, then the *MinimumDistance* node among left leaf node $\mathrm{NODE}_i^{\mathrm{LL}}$ and right leaf node $\mathrm{NODE}_i^{\mathrm{LR}}$ is used for swapping. If the energies and distances of left and right

Input: Binary tree shape WSN network with Storage nodes
Output: Lifetime improved sensors by Node rotation
Step 1: Find Mean energy of Left side sensor nodes

$$\mathrm{E_{MEAN}} = \frac{1}{n} \sum_{i \in n} E_i \qquad (24)$$

Where

$\mathrm{E_{MEAN}}$ - Mean energy of left sensor nodes
n - Total left side sensor nodes
E_i Energy of i[th] sensor node

Step_2: Mark sensor nodes which are less than the energy $\mathrm{E_{MEAN}}$ as Critical_node CN_i

$$CN_i = \begin{cases} 1 & if \quad E_i < \mathrm{E_{MEAN}} \\ 0 & else \end{cases} \qquad (25)$$

Step_3:
$NODE^{SINK} = Sink$
$NODE_i^P = Parent(NODE_i)$
$NODE_i^{LL} = LeafNodeLeft(NODE_i)$
$NODE_i^{LR} = LeafNodeRight(NODE_i)$
$E_i^P = Energy(NODE_i^P)$
$E_i^{LL} = Energy(NODE_i^{LL})$
$E_i^{LR} = Energy(NODE_i^{LR})$
$D_i^P = Distance(NODE_i^P, NODE_i)$
$D_i^{LL} = Distance(NODE_i^{LL}, NODE_i)$
$D_i^{LR} = Distance(NODE_i^{LR}, NODE_i)$
$R_i^P = RotationCount(NODE_i^P)$
$R_i^{LL} = RotationCount(NODE_i^{LL})$
$R_i^{LR} = RotationCount(NODE_i^{LR})$

Rule_1: IF CN_i==1 && $NODE_i^P$!= *Any_StorageNode* && $NODE_i^P$!= $NODE^{SINK}$ &&
$E_i^P > E_i^{LL}$ && $E_i^P > E_i^{LR}$ then
$Swap(NODE_i, NODE_i^P)$
END
Rule_2: IF CN_i==1 && $NODE_i^P$!= *Any_StorageNode* && $NODE_i^P$!= $NODE^{SINK}$ &&
$E_i^{LL} > E_i^{LR}$ && $E_i^{LL} > E_i^P$ then
$Swap(NODE_i, NODE_i^{LL})$
END
Rule_3: IF CN_i==1 && $NODE_i^P$!=*Any_StorageNode* && $NODE_i^P$!= $NODE^{SINK}$ &&
$E_i^{LR} > E_i^P$ && $E_i^{LR} > E_i^{LL}$ then
$Swap(NODE_i, NODE_i^{LR})$
END
Rule_4: IF CN_i==1 && $NODE_i^P$!=*Any_StorageNode* && $NODE_i^P$!= $NODE^{SINK}$ && E_i^P==E_i^{LL}==E_i^{LR} then
D_{MIN}=$Min(D_i^P, D_i^{LL}, D_i^{LR})$
$NODE_{MIN}^{DIST} = MinDistGivenNode$
$Swap(NODE_i, NODE_{MIN}^{DIST})$
END
Rule_5: IF CN_i==1 && $NODE_i^P$!=*Any_StorageNode* and $NODE_i^P$!= $NODE^{SINK}$ && (E_i^P==E_i^{LL}==E_i^{LR}) &&
(D_i^P==D_i^{LL}==D_i^{LR}) then
$R_{MIN} = Min(R_i^P, R_i^{LL}, R_i^{LR})$
$NODE_{MIN}^{ROT} = MinRotationCountGivenNode$
$Swap(NODE_i, NODE_{MIN}^{ROT})$

END
Rule_6: IF CN_i==1 && ($NODE_i^P$ == *Any_StorageNode* || $NODE_i^P$ ==$NODE^{SINK}$) && $E_i^{LL} > E_i^{LR}$ then
$Swap(NODE_i, NODE_i^{LL})$
END
Rule_7: IF CN_i==1 && ($NODE_i^P$ == *Any_StorageNode* || $NODE_i^P$ ==$NODE^{SINK}$) && $E_i^{LR} > E_i^{LL}$ then
$Swap(NODE_i, NODE_i^{LR})$
END
Rule_8: IF CN_i==1 && ($NODE_i^P$ == *Any_StorageNode* || $NODE_i^P$ ==$NODE^{SINK}$) && E_i^{LR}==E_i^{LL} then
D_{MIN}=$Min(D_i^{LL}, D_i^{LR})$
$NODE_{MIN}^{DIST} = MinDistGivenNode$
$Swap(NODE_i, NODE_{MIN}^{DIST})$
END
Rule_9: IF CN_i==1 && ($NODE_i^P$ == *Any_StorageNode* || $NODE_i^P$==$NODE^{SINK}$) && E_i^{LR}==E_i^{LL} &&
D_i^{LR}==D_i^{LL} then
R_{MIN} =$Min(R_i^{LL}, R_i^{LR})$
$NODE_{MIN}^{ROT} = MinRotationCountGivenNode$
$Swap(NODE_i, NODE_{MIN}^{ROT})$
END

Fig. 11 *ERNR algorithm*

leaf nodes are same, then the *MinimumRotationCount* node $\mathrm{NODE}_{\mathrm{MIN}}^{\mathrm{ROT}}$ is used for swapping purpose. The same process is simultaneously performed in the right-hand side nodes also.

2.4 CSDP algorithm

In SNs, the participation of parent nodes is categorised into two kinds as assisting-for-data passing and sensing sensitive data. Often data loss occurs due to the breakage of network by the death of parent-data-passing nodes in tree structure. There is no existing paper provides solution to eliminate this issue. This paper presents a novel solution using an algorithm CSDP. Sensor nodes make data passing through the parent nodes. The parent node may be at the brink of death and leads to disconnected network when it survives in the critical-low-energy level. To avoid the death of a parent node, parent node should be obliged only to sensing the data and avoids passing the data at critical-low-energy level. Since the work load of the parent node is reduced, its lifetime gets improved and hence the network lifetime too. However, this phenomenon blocks the data passing to child nodes of the critical parent node. To resume the transmission passage, the neighbour nodes of the disconnected child node ignore the tree structure and find alternative path for data passing using the proposed algorithm CSDP (Fig. 12).

The binary tree SN constructed using ‘SN placement’, ‘recharging of SNs’ and ‘rotation’ is used as input for the CSDP algorithm. The CSDP algorithm is used to handle the critical state in data passing for left-hand side and right-hand side nodes one after another. This implementation enhances the data-passing capability of the SN and improves network lifetime. The $D_{\mathrm{Threshold}}$ parameter can be set based on the broadcasting capacity of sensor nodes. The term NODE_F can be expanded as *FirstOccurranceMaxEnergyGivenNode* and NODE_S can be expanded as *SecondOccurranceMaxEnergyGivenNode*.

The CSDP algorithm extends the network lifetime by reducing the load of critical energy nodes. The mean energy is computed using (26) and the critical state of a parent node NODE_i^P is confirmed if the condition $E_i^P < E_{\mathrm{MEAN}}$ is true. When the parent node NODE_i^P of a child node NODE_i meets critical state, the child node NODE_i collects the neighbour nodes in the $\mathrm{LIST}^{\mathrm{NEIG}}$. If any $\mathrm{LIST}^{\mathrm{NEIG}}$ node falls under the critical state or matches with SN, the corresponding neighbour is removed from the list. If a neighbour contains critical-state node in its communication path, the corresponding neighbour is also removed from the $\mathrm{LIST}^{\mathrm{NEIG}}$.

The highest energy node in $\mathrm{LIST}^{\mathrm{NEIG}}$ is found and it is called as NODE_F. The second highest node is found and it is known as NODE_S. Suppose the first highest energy and second higher energy are same, then $\mathrm{FLAG}_{\mathrm{TwiceMax}}$ is assigned as true. If neighbour count C_i^{NEIG} is equal to 0, the NODE_i will stop communication soon because there are no neighbours around it. If C_i^{NEIG} is equal to 1, then NODE_F is used for communication. If neighbour count is 2, NODE_F is used for communication.

If the C_i^{NEIG} is >2 and $\mathrm{FLAG}_{\mathrm{TwiceMax}} = \mathrm{true}$, then the communication node (CN) is found using (27)

$$\mathrm{CN} = \begin{cases} \mathrm{NODE}_F & \text{if } D_F < D_S \\ \mathrm{NODE}_S & \text{if } D_S < D_F \end{cases} \tag{27}$$

Suppose the neighbour count is >2 and the $\mathrm{FLAG}_{\mathrm{TwiceMax}} = \mathrm{false}$, theCN is detected using (28)

$$\mathrm{CN} = \begin{cases} \mathrm{NODE}_F & \text{if } \quad D_F == D_S \\ \mathrm{NODE}_F & \text{if } \quad D_F < D_{\mathrm{Threshold}} \\ \mathrm{NODE}_S & \text{else} \end{cases} \tag{28}$$

The CMIMO [23] is combined with the CSDP algorithm to improve the energy efficiency of the proposed method and the CMIMO is a distributed MIMO-adaptive energy-efficient clustering/routing scheme which is proposed by Mohammad *et al.* in [23]. The CSDP algorithm only selects the neighbourhood node for the critical-state management and the clustering/routing are decided by CMIMO method which is also used at critical state. The CMIMO is constructed by allocating two CHs which are master CH (MCH) and slave CH (SCH) and the cooperation of these two CHs with single-input–single-output system, construct an advanced MIMO architecture. The SNs and sink can be preferred for the selection of MCH and SCH in a cluster. The proposed CSDP method reduces the data loss occurrence in data storage.

2.5 Query data-retrieval system

The sensor nodes store their sensed data in the SNs or sink using the proposed method which includes the techniques such as SN placement, recharging, rotation and critical-state management to increase the lifetime of SN. The user may place a demand on sink to retrieve the stored data of a particular node, then the sink tries to checkup the availability of the required node's data in SN or sink, and if available the sink puts forward that data to the user (query can be a range of information or a single information).

3 Simulation results

The simulation of the proposed data storage and retrieval method is implemented using Network Simulator 2.0. The channel type used in this simulation is *WirelessChannel* and the radio-propagation model used here is *TwoRayGround*. The network interface type is configured using *WirelessPhy* and media access control (MAC) type is 802_11. The interface queue type used is *DropTail/PriQueue* and link layer type is *LL*. The antenna model used to configure the NS2 is OmniAntenna and the maximum packet in interface queue is settled as *2*. Total node parameter used in this simulation is 154 and node movement speed is configured as 1 m/s. The *X* dimension of topography is fixed as 1000 and *Y* dimension of topography is fixed as 1000.

The NS2 implementation of the proposed WOSNP algorithm includes SNs count as 8. The 154 nodes are configured as below:

- One sink.
- 142 Sensor nodes.
- Eight SNs.
- One ES.
- Two MCs.

The entire sensor nodes are deployed with binary tree structure based on Fig. 2. The nodes are separated and grouped as left-hand side nodes and right-hand side nodes. In the NS2 simulation, left-hand side part contains four SNs and right-hand side part contains four SNs. The SN arrangement is performed based on WOSNP algorithm. Sink is connected with permanent power source. SNs are manufactured with recharging facility. Sensor nodes are assumed as homogenous and equipped with battery power sources. The NS2 simulation stores the descendent nodes data in SNs and it delivers that data to sink when sink requests that data through query process.

This simulation implements the proposed EEWR algorithm which is built with the node recharging facility. The NS2 simulator is configured with one ES and two MCs to achieve the simulation. In real-time practise, the MC is a mobile robot which contains a wireless power charger for sensor nodes. In the simulation, left-hand side of network contains one MC and other MC is positioned at the right-hand side. The ES is equipped with heavy power pack and it is poisoned near sink in the NS2 topography. The SNs are manufactured with wireless power receivers to adopt with wireless recharging. In the EEWR simulation, all sensor nodes are initially having same quantity power and entire SNs are also powered with equal measure initially. At the execution time, the charge of SNs is diminished and it sends recharge request to sink. Sink delivers that message to ES and it signals the corresponding MC to make recharge of exact SN. The communication range of sensor nodes is decided as 100. In the simulation, the initial energy assigned for each sensor node is 100 J and for SNs is 200 J. An SN spends 1 J energy per storage. Assumptions are made that a single hop passing for a transmission occupies 1 J energy and if five hops are required

Input: Sensor nodes based on binary tree structure (including storage nodes)
Output: Lifetime improved sensor network
Step_1: DO Compute Mean energy

$$\mathrm{E_{MEAN}} = \frac{1}{n} \sum_{i \in n} E_i \tag{26}$$

Where
$\mathrm{E_{MEAN}}$ - Mean energy of nodes
n - Total nodes
E_i Energy of i[th] node

Step_2: FOR EACH *ChildNode* $NODE_i$
$NODE_i^P = Parent(NODE_i)$
$E_i^P = Energy(NODE_i^p)$
IF $E_i^P < E_{MEAN}$
$FLAG_P^{CRIT} = true$ // *CriticalStateFlag*
Collect $NeighborNodes(NODE_i)$ in $LIST^{NEIG}$
$k = 0$
FOR EACH $LIST^{NEIG}$
$Node = LIST^{NEIG}(k)$
IF
$FLAG_{Node}^{CRIT} == true$ ||
$TypeOfSensorNode(LIST^{NEIG}(k)) == StorageNode$ ||
$IsDataPassingPathHasCriticalStateNode(Node) == true$ *Then*
DO Remove $LIST^{NEIG}(k)$ from $LIST^{NEIG}$
ELSE
$k = k + 1$
END
END FOR
E_{MAX1}=*FirstOccurranceMaxEnergy(*$LIST^{NEIG}$*)*
$NODE_F$ =*FirstOccurranceMaxEnergyGivenNode*
E_{MAX2} =*SecondOccurranceMaxEnergy(*$LIST^{NEIG}$*)*
$NODE_S$ =*SecondOccurranceMaxEnergyGivenNode*
IF $E_{MAX1} == E_{MAX2}$ THEN
$FLAG_{TwiceMax} = true$ // *Twice Max Occurrence*
END
D_{MIN} =*MinDist(*$NODE_i$, $LIST^{NEIG}$*)*
$NODE_{MD} = MinDistGivenNode$
$D_F = Dist(NODE_i\ ,\ NODE_F)$
$D_S = Dist(NODE_i\ ,\ NODE_S)$
$C_i^{NEIG} = NeighborCount(NODE_i)$
RULE_1:
IF $C_i^{NEIG} == 0$ THEN
DO: $Data\ Sending\ of\ NODE_i\ will\ soon\ stop\ because\ of\ non-availability\ of\ Neighbors$
END
RULE_2:
IF $C_i^{NEIG} == 1$ THEN
DO: $NODE_i$ uses $NODE_F$ for communication
END
RULE_3:
IF $C_i^{NEIG} == 2$ THEN
DO: $NODE_i$ uses $NODE_F$ for communication
END
RULE_4:
IF $C_i^{NEIG} > 2$ && $FLAG_{TwiceMax} == true$ && $D_F < D_S$ THEN
DO: $NODE_i$ uses $NODE_F$ for communication
END
RULE_5:
IF $C_i^{NEIG} > 2$ && $FLAG_{TwiceMax} == true$ && $D_S < D_F$ THEN
DO: $NODE_i$ uses $NODE_S$ for communication
END
RULE_6:
IF $C_i^{NEIG} > 2$ && $FLAG_{TwiceMax} == false$ && $D_F == D_{MIN}$ THEN
DO: $NODE_i$ uses $NODE_F$ for communication
END
RULE_7:
IF $C_i^{NEIG} > 2$ && $FLAG_{TwiceMax} == false$ && $D_F < D_{Threshold}$ THEN
DO: $NODE_i$ uses $NODE_F$ for communication
END IF
RULE_8:
IF $C_i^{NEIG} > 2$ && $FLAG_{TwiceMax} == false$ && $D_F \geq D_{Threshold}$ THEN
DO: $NODE_i$ uses $NODE_S$ for communication
END IF
END IF
END FOR

Fig. 12 *CSDP algorithm*

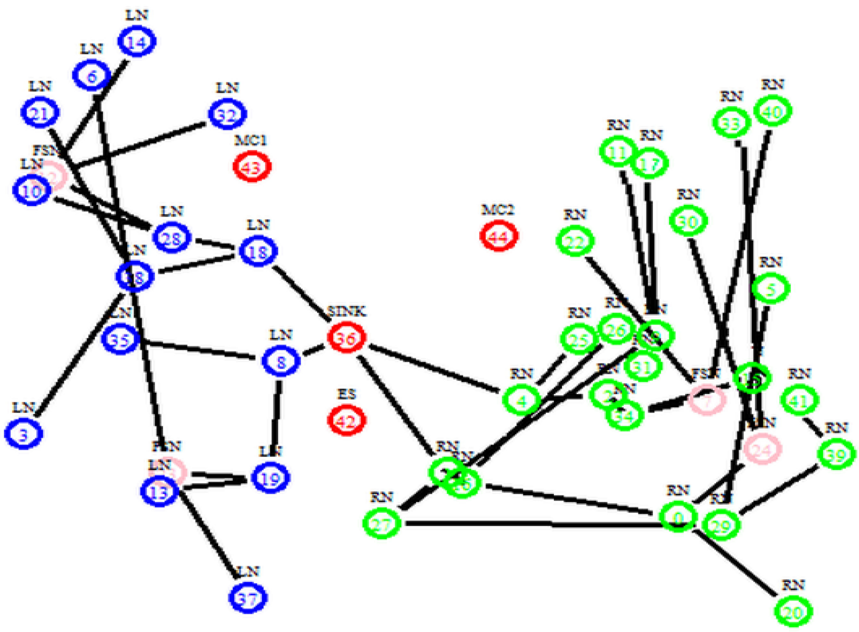

Fig. 13 *Simulation screenshot of NS2 (Network Simulator 2.0) implementation*

Table 1 NLR analysis for 100 nodes with 12 SNs

Methods	NLR
Bo Sheng	2.4
Yang Peng	2.5
Fatme El	6.0
proposed	24.0

Table 2 Average NLR analysis

Methods	Average NLR
Bo Sheng	1.6
Yang Peng	2.7
Fatme El	7.0
proposed	15.0

for a data transmission path, then 5 J energy will be spent out. In the EEWR algorithm simulation, the *CriticalServiceTime* parameter is assigned as 5 s and *BasicServiceTime* is fixed as 10 s. The *PeriodicTime* is set as 5 s and *AdditionalTime* is maintained as 5 s. The *CriticalEnergy* reaching parameter is fixed as 10%. The simulation process checks the queue for *PeriodicTime* interval and it recharges the SNs up to *CriticalServiceTime* or *BasicServiceTime*. Some time *AdditionalTime* recharge is also maintained based on the algorithm. The *CriticalServiceTime* recharging charges the SN up to 20 J energy and the *BasicServiceTime* recharging adds 40 J energy. The parameters such as *CriticalServiceTime*, *BasicServiceTime*, *PeriodicTime*, *AdditionalTime*, *Initial Energy*, *CriticalEnergy reaching Percentage* and *MCs count* depend on user configuration based on their own network design. In this simulation, we adopt with these configuration values.

The proposed ERNR algorithm is implemented in NS2 with swapping or rotation process. For each rotation, this simulation assumes 1 J energy spending for a single node. In real applications, these sensor nodes make rotations with the help of parallel cables but here we do only simulation. Swapping is allowed only for sensor nodes and not for SNs and sink. The node swapping speed is pointed as 1.0 m/s. The distance computations such as D_i^P, D_i^{LL}, D_i^{LR} are calculated using the general formula which is described in (29)

$$\text{distance} = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \tag{29}$$

where $x1-$ x coordinate of node 1, $y1-y$ coordinate of node 1, $x2-x$ coordinate of node 2 and $y2-y$ coordinate of node 2.

The proposed CSDP algorithm is simulated with the distance calculation using (29). The $D_{\text{Threshold}}$ parameter is assumed as 75. It increases the lifetime of sensor nodes particularly so the network stability is highly maintained.

Fig. 13 shows the NS2 implementation screen for 36 nodes with 4 SNs. It also shows one ES, one sink and two MCs. The sink, ES and MCs are marked in red colour and the SNs are marked by pink colour. The left-hand side nodes are marked by blue, whereas the right-hand side nodes are marked by green colour.

The proposed method is compared against the three papers such as Sheng *et al.* [1], Peng *et al.* [10] and El-Moukaddem *et al.* [19] to evaluate the performance level of it. The experimental analysis is obtained by varying sensor nodes count as 50, 100 and 150. The SN count parameter is also varied by 4 and 8. The node swapping movement speed range is also changed in between 0.1 and 1.0 m/s to get better analysis.

Table 1 describes the network lifetime ratio (NLR) analysis using 12 SNs. This paper generates 20 networks each consisting of 100 nodes placed uniformly at random. The NLR is computed using general max life method [19]. If A is any network improvement algorithm and I is input instance, then the NLR can be computed using (30)

$$\text{NLR} = \frac{L(A, I)}{L(I)} + \text{EV}(I) \tag{30}$$

Here, $L(A, I)$ described the lifetime achieved using the algorithm A on I and $L(I)$ described the lifetime without any algorithm. The network lifetime is measured using the number of rounds of simulation time till all the nodes drain their energy completely. The $\text{EV}(I)$ means initial variance of I. In this research, each sensor node possesses same initial energy so that here $\text{EV}(I) = 0$. So the term $\text{EV}(I)$ can be neglected. The proposed method reaches the NLR up to 24.0 because of the implementation with 12 SNs and 4 proposed algorithms.

Table 2 describes the average NLR analysis with the configuration of four SNs. The NLR calculation is computed separately for 50 nodes, 100 nodes and 150 nodes using (30). The average of these three NLRs are computed and listed in table. From this analysis, it can be shown clearly that the proposed method is better one than previous versions.

The proposed method improves the network lifetime by 24 times (from Table 1) using 12 SNs. The proposed method provides the average NLR as 15.0 (from Table 2 and Fig. 14*a*), whereas other methods generate less improvement, because the proposed

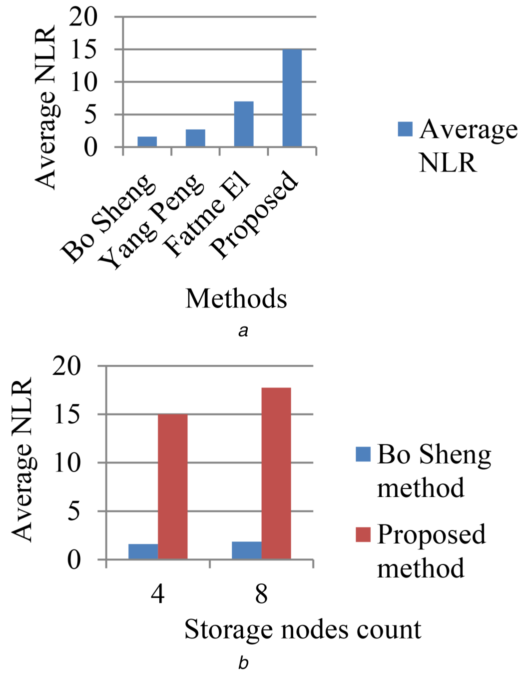

Fig. 14 *Average NLR analysis for 50 sensor nodes*
(a) With four SNs, ***(b)*** With different SNs count

Table 3 Average lifetime improvement using 50 nodes at different speeds for Fatme El method and the proposed method (four SNs included)

Node movement speed, (m/s)	Average NLR	
	Fatme El method	Proposed method
0.1	7.19	14.97
0.5	7.22	15.0
1.0	7.25	15.04

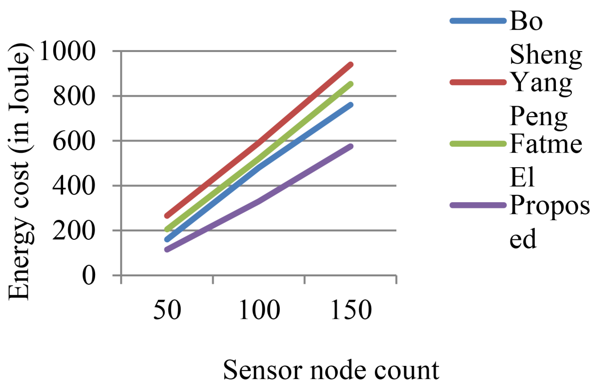

Fig. 15 *Chart for energy cost analysis for various sensor node counts*

method is enriched with the algorithms such as WOSNP, EEWR, ERNR, CSDP and CMIMO. The data-passing PL for each sensor node is reduced because WOSNP algorithm places SNs, so that the energy consumption of sensor nodes is reduced. The EEWR algorithm charges the SNs through wireless recharging, so that SNs are maintained in living mode for a long period to slow down the death of WSN. The fast death of *closer nodes of storage medium* is minimised using the node rotation by ERNR algorithm. The critical-state-energy nodes are relieved from the *assisting-for-data-passing* duty using CSDP and CMIMO algorithms and they are only permitted to sense the data, so that critical energy nodes extend their lifetime periods. From Table 3, it can be concluded that when the node swap movement speed is increased, then the lifetime is increased for the Fatme El and proposed methods, so that the 1 m/s speed is advisable one. From Fig. 14*b*, it is proved that the increment of SN count improves the network lifetime at

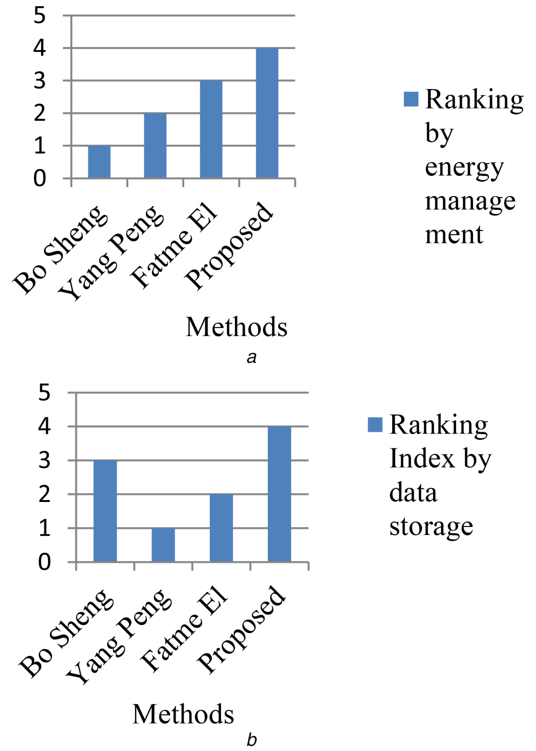

Fig. 16 *Performance-based rank analysis*
(a) Ranking by energy management, ***(b)*** Ranking based on data-storage performance without data loss

considerable level. The proposed method provides better improvement in network lifetime than Bo Sheng method.

The energy cost of a single node to make storage in a storage medium is computed by assuming that each hop in its storage path spends 1 J energy, so if the storage path hops length is 5, then that node needs 5 J energy to make storage. From Fig. 15, it can be noted that when the sensor node count increases, the energy cost also increases. The Bo Sheng method reduces the energy cost than the Yang Peng method because of the SN placement in Bo Sheng method. The proposed method makes extra reduction in the energy cost than the Bo Sheng method due to the brilliant placement of SNs using WOSNP algorithm.

Fig. 16*a* expresses the ranking of the four methods in case of energy management and the higher value rank means better method and vice versa. Fig. 16*b* illustrate the performance of four methods in the case of data storage without data loss and the proposed method wins credit because of the critical-state management using CSDP algorithm. Here, higher ranking index means better method in data-storage performance.

Fig. 17*a* depicts that the proposed method takes less time for data storage because it is supported by the brilliant placement of SNs through the WOSNP algorithm for the network which is configured of four SNs. Here, time taken is computed by assuming 0.01 s for passing a single hop to data storage. Fig. 17*b* depicts that the proposed method takes only reasonable time for data retrieval (for 50 nodes including 4 SNs) because the data should be extracted from the SNs and sink.

4 Conclusion

This paper presents a new methodology for data storage and retrieval system in WSN with four novel algorithms such as WOSNP, EENR, ERNR and CSDP to improve the lifetime of WSN. These algorithms enhance the network lifetime by conservation, management and renewal of energy. The WOSNP algorithm positions the SNs in an intelligent way to hike NLR and EENR algorithm provides solution for emergency recharging. The ERNR algorithm stabilises the network through node swapping. The CSDP algorithm diminishes the response level of critical parent node to raise NLR. We show detailed illustration for the designs in simulation. Simulations are made to analyse the

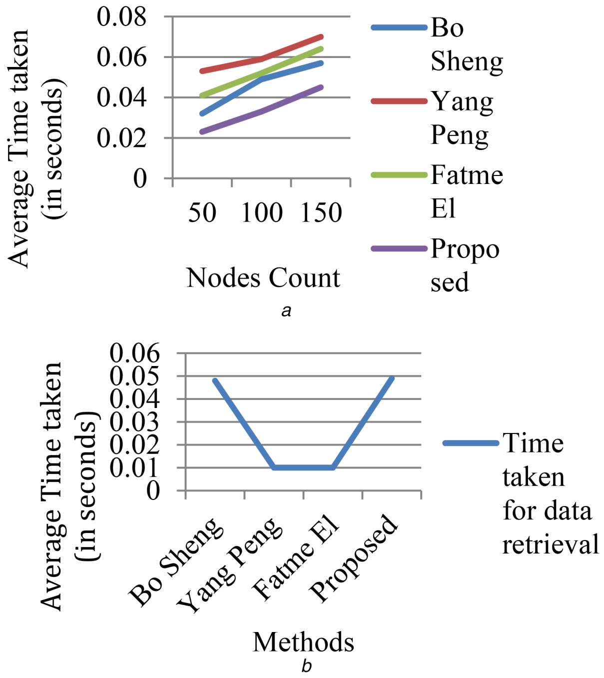

Fig. 17 *Average time taken analysis*
(a) Average time taken for data storage, ***(b)*** Average time taken for data retrieval

performance of the proposed method using different parameters and the experimental results depict that the proposed method can improve the average lifetime of network by more than a factor of 24 (using 12 SNs). By increasing SNs count, the network lifetime can elongate than the factor of 24. The average NLR of the proposed method is 15 with support of 4 SNs. The extensive simulation results affirm that the proposed system can boost the SN lifetime significantly. The average time taken analysis proves that the proposed method occupies only very reasonable quantity of time for data storage (for 100 nodes only 0.033 s) and retrieval (for 50 nodes only 0.049 s). The energy cast is also minimum for the proposed method than existing versions. We believe that the proposed comparative analysis will be very helpful for future research activities in this field. Once queries are given by user, the sink searches data from SN and itself, to deliver data to user. The overall benefits of the proposed data storage and retrieval method concludes that the proposed method is the best than other three existing versions. The implementation of data compression on heavy size data can be left as future enhancement to further improve the network life period.

5 References

[1] Sheng, B., Li, Q., Mao, W.: 'Optimized storage placement in sensor networks', *IEEE Trans. Mob. Comput.*, 2010, **9**, (10), pp. 1437–1450

[2] Ganesan, D., Greenstein, B., Heidemann, J., *et al.*: 'Multi resolution storage and search in sensor networks', *ACM Trans. Storage*, 2005, **V**, (N), pp. 277–315

[3] Ahn, J., Krishnamachar, B.: 'Fundamental scaling laws for energy efficient storage and querying in wireless sensor networks', *ACM MobiHoc*, 2006, pp. 22–25

[4] Albano, M., Chessa, S., Nidito, F., *et al.*: 'Dealing with non uniformity in data centric storage for wireless sensor networks', *IEEE Trans. Parallel Distrib. Syst.*, 2011, **22**, (8), pp. 1398–1406

[5] Apte, S.S., Mane, S.M.: 'Improvement of limited storage placement in wireless sensor networks', *IOSR J. Comput. Eng. (IOSR-JCE)*, 2013, **12**, (2), pp. 107–111

[6] Mane, S.M., Bhanwase, V.V., Vadhoot, A., *et al.*: 'Efficient an optimal storage placement in sensor network', *Int. J. Sci. Eng. Res.*, 2014, **5**, (2), pp. 54–59

[7] Yao, W., Li, M., Wu, M.: '*Inductive charging with multiple charger nodes in wireless sensor networks*' (APWeb Workshops Springer, 2006)

[8] Afzal, M.I., Mahmood, W., Sajid, S.M., *et al.*: 'Optical wireless communication and recharging mechanism of wireless sensor network by using CCRs', *Int. J. Adv. Sci. Technol.*, 2009, **13**, pp. 49–59

[9] Tong, B., Li, Z., Wang, G., *et al.*: 'How wireless power charging technology affects sensor network deployment and routing'. IEEE ICDCS, 2010

[10] Peng, Y., Li, Z., Zhang, W., *et al.*: 'Prolonging sensor network lifetime through wireless charging'. IEEE Real-Time Systems Symp. (RTSS), 2010

[11] Sharifi, M., Sedighian, S., Kamali, M.: 'Recharging sensor nodes using implicit actor coordination in wireless sensor actor network', *Wirel. Sens. Netw.*, 2010

[12] Fu, L., Cheng, P., Gu, Y., *et al.*: 'Minimizing charging delay in wireless rechargeable sensor networks'. IEEE Proc. INFOCOM, 2013

[13] Yang, Y., Wang, C., Li, J.: 'Wireless rechargeable sensor networks – current status and future trends', *J. Commun.*, 2015, **10**, (9), pp. 696–706

[14] Yuanchao, S., Hamed, Y., Peng, C., *et al.*: 'Near-optimal velocity control for mobile charging in wireless rechargeable sensor networks', *IEEE Trans. Mob. Comput.*, 2016, **15**, (7), pp. 1699–1713

[15] Yuanchao, S., Kang, G.S., Jiming, C., *et al.*: 'Joint energy replenishment and operation scheduling in wireless rechargeable sensor networks', *IEEE Trans. Ind. Inf.*, 2017, **13**, (1), pp. 125–134

[16] Jain, S., Shah, R., Brunette, W., *et al.*: 'Exploiting mobility for energy efficient data collection in wireless sensor networks', *Mob. Netw. Appl.*, 2006, **11**, (3), pp. 327–339

[17] Luo, J.J., Hubaux, P.: 'Joint mobility and routing for lifetime elongation in wireless sensor networks'. IEEE 24th Annual Joint Conf., IEEE Computer Communication Society, 2005, pp. 1735–1746

[18] Kansal, A.D., Jea, D., Estrin, D., *et al.*: 'Controllably mobile infrastructure for low energy embedded networks', *IEEE Trans. Mob. Comput.*, 2006, **5**, (8), pp. 958–973

[19] El-Moukaddem, F., Torng, E., Xing, G.: 'Maximizing network topology lifetime using mobile node rotation', *IEEE Trans. Parallel Distrib. Syst.*, 2015, **26**, (7), pp. 1958–1970

[20] Zitterbart, D.P., Wienecke, B., Butler, J.P., *et al.*: 'Coordinated movements prevent jamming in an emperor penguin huddle', *PLoS ONE*, 2011, **6**, (6), p. e20260, doi: 10.1371/journal.pone.0020260

[21] Handy, M.J., Haase, M., Timmermann, D.: 'Low energy adaptive clustering hierarchy with deterministic cluster-head selection'. Proc. Fourth Int. Workshop on Mobile and Wireless Communications Network, 2002, pp. 368–372

[22] Liang, Q.: 'Cluster head election for mobile *ad hoc* wireless network'. Proc. 14th IEEE Int. Symp. Personal, Indoor and Mobile Radio Comm.(PIMRC), 2003, pp. 1623–1628

[23] Mohammad, Z., Siam, K.M., Younis, O.: 'Energy-efficient clustering/routing for cooperative MIMO operation in sensor networks', TR-UA-ECE, 2008

[24] Chao-Tsun, C., Chih-Yung, C., Shenghui, Z.: 'SRA: a sensing radius adaptation mechanism for maximizing network lifetime in WSNs', *IEEE Trans. Veh. Technol.*, 2016, **65**, (12), pp. 9817–9833

[25] Abdelrahman, A., Mohammad, H., Bamidele, A., *et al.*: 'Dynamic clustering and management of mobile wireless sensor networks', *Elsevier, Comput. Netw.*, 2017, **117**, pp. 62–75

[26] Kurs, A., Karalis, M.R., Joannopoulos, J.D., *et al.*: 'Wireless power transfer via strongly coupled magnetic resonances', *Science*, 2007, **317**, p. 83