Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2021 SCPE. Volume 22, Issues 1, pp. 81–92, DOI 10.12694:/scpe.v22i1.1847

IMPROVED LOCALIZED SLEEP SCHEDULING TECHNIQUES TO PROLONG WSN LIFETIME

NACHIKETA TARASIA[*], AMULYA RATNA SWAIN[†], SOHAM ROY[‡] AND UDIT NARAYANA KAR[§]

Abstract. A standard Wireless Sensor Networks(WSNs) comprises of low-cost sensor nodes embedded with small batteries. To enhance the network lifetime of WSN, the number of active nodes among the deployed nodes should be minimum. Along with this, it must be ensured that coverage of the targeted area would not get affected by the currently active nodes. Considering different applications of WSN, there is still a demand for full coverage or partial coverage of the deployed area. Irrespective of the circumstances, a proper sleep scheduling algorithm needs to be followed. Else, the active nodes will be tuckered out of the battery. Random distribution of the sensor nodes in a common area may have multiple active nodes. It is essential to identify the redundant number of active nodes and put them into sleep to conserve energy. This paper has proposed a methodology where the active sensor nodes form a hierarchical structure that heals itself by following a level-wise approach. In the meantime, it also detects the total number of redundant nodes in the coverage area. The performance of the proposed protocol is evaluated using the Castalia simulator. The simulation results show that the proposed level-wise periodic tree construction approach increases the network's durability in conjunction with the level wise approach.

Key words: Wireless Sensor Network, Sleep Scheduling, Coverage, Energy Efficient.

AMS subject classifications. 68M14

1. Introduction. Wireless Sensor Networks (WSNs) have been generally considered as one of the most imperative innovations. A regular Wireless Sensor Network comprises many low-cost devices known as sensor nodes. These multi-practical sensor nodes have limited battery life and are generally used to monitor a region of interest [1] [2]. These sensor nodes have inbuilt miniaturized controllers and radio transceivers. Subsequently, sensor nodes can detect outside events, process the detected information, and transmit it to the sink. WSNs are broadly utilized for ecological condition monitoring, security surveillance of combat zones, monitoring untamed natural life, etc. [3]. Sensor nodes, as a general, are densely deployed in an inhuman domain, where it would be tough to maintain the nodes' battery capacity. To observe and control the physical conditions, WSNs should address the following two requirements: (i) sensing in the intended zone should be appropriately done, and (ii) proper communication should be maintained among the sensor nodes to ensure that the collected data is properly transmitted to the sink node. Else, the overall collection of the data is pointless.

In WSN, a sensor node can sense its detecting range, which is called as sensing coverage of the node. The network coverage [3] [4] could be translated as the aggregate coverage by all the *ACTIVE* sensor nodes. Similarly, an *ACTIVE* node should send the information to another node within its radio coverage area. More is the number of *ACTIVE* nodes; more is the consumption of energy. To boost the lifetime of the WSN, it is logical to limit the number of *ACTIVE* nodes while accomplishing the most extreme conceivable sensing and radio coverage.

Depending upon the necessities of the application, the sensing and radio coverage may be limited [5]. For an application like intrusion detection or movement detection, it must have at least one *ACTIVE* sensor node for each location. So, most of the *ACTIVE* nodes will run out of battery very rapidly. Similarly, for applications like humidity or temperature monitoring, it is required to have fewer numbers of *ACTIVE* nodes, with limited coverage. Nevertheless, if any node is *ACTIVE* for a long duration, it is necessary to have a proper sleep

[*] KIIT Deemed to be University, Bhubaneswar, India (ntarasia06@gmail.com).
[†] KIIT Deemed to be University, Bhubaneswar, India (swainamulya@gmail.com).
[‡] KIIT Deemed to be University, Bhubaneswar, India (skr2538@gmail.com).
[§] KIIT Deemed to be University, Bhubaneswar, India (uditnarayankar@gmail.com).

scheduling mechanism to manage the whole network. Applying a sleep scheduling mechanism over WSN allows the sensor nodes to share their duties among themselves. This mechanism can be constructive in conserving the energy of the sensor nodes. There may be instances where multiple sensor nodes are deployed to cover a common area, which ultimately resulting in wastage of battery. In such scenarios, it is essential to identify the redundant nodes. These redundant nodes can move to sleep mode to enhance the network lifetime. To achieve this, WSNs must follow an appropriate duty cycle mechanism that govern the cycle of slreep and wake-up mode among the sensor nodes.

In this manuscript, we have proposed a sleep scheduling mechanism where redundant nodes are identified and put those nodes into sleep mode to conserve energy. Our proposed protocol ensures full coverage of the desired region. The proposed approach uses a level-wise hierarchical structure, where the sensor nodes mend themselves locally intending to save energy. The rest of the paper is organized as follows. Section II presents state of the art, followed by a proposed approach presented in Section III. The detailed simulation generated results, and analysis of the generated graphs are presented in Section IV. Finally, the manuscript concludes in Section V.

2. State of the Art. Energy conservation is one of the significant taxonomies of WSNs. To enhance the overall lifetime of the WSNs, energy conservation schemes are being consistently investigated and scrutinized by researchers across the globe. Hence, there is an abundant number of research articles available in the background. There exists a considerable different coverage optimization approaches like probing environment and adaptive sleeping (PEAS), probing environment and collaborating adaptive sleeping (PECAS), controlled layer deployment (CLD), random back-off sleep protocol (RBSP), and so on. In PEAS [6], the network lifetime has been extended by embracing a basic 3-mode approach, i.e., sleeping, probing, and active. For each probing region, an active node has remained in charge of coverage, and in the meantime, the other sensor nodes stay in sleep mode to spare energy. The sleep node goes into the probing stage from time to time and checks for the accessibility of the active node's presence by sending a probing message. At that point, it returns to sleep mode to spare energy. The PEAS has some impediment, i.e., a sensor node, which ended up active, must be in a similar state all the time till it dries out which may result in lopsided vitality utilization in the system. PECAS [7] is the extended version of PEAS, where it overcomes few limitations of the prior. The active node in PECAS goes to sleep mode after a stipulated period, but it shares its remaining energy with its neighbors before it goes to sleep mode. This information is used by the probing nodes in the region to decide when to be active again.

The cascading effect is the most commonly faced issues in WSN between the sink and the leaf nodes. The nodes closer to the sink are engaged in transmitting data most of the time compared to far away from leaf nodes. If any event occurs far away from the sink, then more intermediate nodes participate, thus shortening the network's lifetime. The PEAS algorithm has been modified in CLD [8] approach, which uses deterministic node deployment to counter the cascading effect. In CLD, the average distance between two active nodes is maintained as 2r/3, where r is the sensing radius. Sleeping nodes surround the active nodes at a distance of r/6. More number of sleep nodes are placed near the sink node to overcome the cascading effect. CLD can be implemented in those applications where the target area is known beforehand.

RBSP [9] is a probe-based algorithm that uses information regarding the rest of the energy level of the present active node. Here, back-off sleep time is calculated, which is utilized by the currently active node's neighboring nodes to choose when to wake up and examine the currently active node's status. Using this approach, when an active node has high outstanding energy, a neighboring node's chance to turn active is low and the other way around.

The authors [28] recommended sleep scheduling algorithm is intended to improve network administration by reducing power distribution due to the passive listening of nodes. The sleep period is proportionate to the remaining power of sensor nodes and adaptive. A sleep scheduling approach is required to adjust the network administration by utilizing the least energy. A distributed sleep scheduling system allows the sensor node to entirely satisfy the sensing ranges and switch off the node if the conversation doesn't occur or does not have adequate energy [29].

One of the significant challenges in formulating such systems lies within the obliged energy and computational assets accessible to sensor nodes. These constraints must be taken into consideration at all levels of the system progression. The arrangement of sensor nodes is the primary step in setting up a sensor network. Since WSNs contain many sensor nodes, the nodes must be placed in clusters [30], where the area of each specific node cannot be wholly ensured a priori. In this manner, the number of nodes deployed to cover the complete observed zone is mostly higher. The authors in [10] have presented an algorithm that chooses mutually elite sets of sensor nodes, where the union of these sets covers the observed region. The interim actions are same for all collections, and one of the groups is active. This algorithm accomplishes considerable energy savings along with ultimately protecting coverage.

In [11], the authors defined the coverage issue as a choice based problem. Here, the objective is to decide whether each point within the desired region of the sensor network is secured by at least k nodes, where k could be predefined esteem. The sensing ranges of nodes can be unit disks or non-unit disks. This paper displayed polynomial-time calculations in terms of the number of nodes, which can be effectively interpreted in distributed conventions. The simulation result showed that energy could be conserved along with the fault-tolerant model in an area where nodes are deployed randomly. In [12], the authors address the difficulty of choosing the least number of associated sensor nodes to cover a distributed set of interest objects. A centralized algorithm runs by iteratively appending nodes that maximize a measure called k-benefit to an initially empty set of nodes. Though running with the least number of sensor nodes does not singularly signify the system's maximal lifetime. Without global optimization, any nodes that can cover many targets could be recorded to work massively, and they will soon run away from energy.

In [13], the authors disseminated a single step arrangement algorithm that divides the region of intrigued into two equal networks. The nodes are deployed to possess each point in grids to be completely covered and connected. Two strategies have been proposed for the deployment of sensor nodes, i.e., randomized and planned as per the situations. The authors in [14] have proposed an approach in which a moving robot fixes the coverage hole by picking nodes from the coverage area where redundant nodes are present. A carrier-based sensor relocation by robots to mend coverage holes has also been proposed in [15] that uses a virtual force approach in a grid structure of interest. In this approach, full coverage is achieved by placing redundant sensor nodes in coverage holes with the help of robots that randomly move in the network and are restricted in grids.

In [16] and [17], an energy-efficient coverage approach has been proposed where authors consider a large number of sensor nodes were deployed in the area of interest to achieve coverage. In this approach, the number of sensor nodes is highly populated; these nodes are divided into several disjoint sets. The idea of prolonging the network lifetime is by putting the rest of the nodes into sleep mode, whereas one disjoint set is active in a specific area for coverage. This mentioned algorithm follows a centralized approach. A localized and distributed algorithm [18] called Node Scheduling scheme Based on the Eligibility rule(NSBE) that follows a scheduling approach where each node decides to be in either active or in sleep mode. At any instance, one node is active for covering the area of interest, and the redundant nodes are in sleep mode. In each cycle, the active node tries to find an alternative node, which will cover the area of interest. If the alternative node is found, then the active node can go to sleep mode. There is no simulation proof of the stated approach.

Sensor Scheduling for k-Coverage(SSC) [19] monitors a two-dimensional region, where the same locations do not have sensor nodes more than one. K number of sensor nodes must guard each point continuously. The SSC is a centralized algorithm, which is NP-hard. The existence of WSN is determined as the whole span during which the entire region is k-covered. The authors in [20] recommended a solution for domain coverage in a synchronous WSN, where radio strength is equal to the sensing range. This approach considered that each sensor node knows the exact location of its neighbor nodes. The precise location information helps the node to decide autonomously either to be in active or in sleep mode. A node can go to sleep mode if it's sensing range is covered by its neighbor nodes. A backoff algorithm is being followed to avoid coverage gaps while going to sleep mode. However, this may affect the node connectivity.

A centralized algorithm has been proposed by [21], where the sensor nodes are grouped into different sets. The nodes of a particular set are active to cover the required positions. A scheduling algorithm decides the nodes' state of being alternate between active and sleep to extend the whole network's lifespan. An adaptable energy-efficient sensing coverage protocol using a differentiated monitoring service for WSN has been proposed by [22]. Here, each node ensures a certain coverage degree by obeying dynamic scheduling for self. The proposed protocol uses the grid-based approach, where the deployed area is split into several grids, and each grid is allocated with a sensor node. Each active node covering any grid also maintains a list of other nodes that can also cover the same grid. If the network is dense, then the time and space complexity are high as the list contains the sensor node details.

The author in [23] proposed a greedy algorithm that maximizes the network lifespan by following a state scheduling approach. The state scheduling approach gives autonomy to nodes to become active or inactive initially. The proposed approach then tries to cover the target area with the selected active nodes; if failed, the algorithm restarts. However, the algorithm consumes more energy concerning the exchange of messages [24, 25]. It should be noted that all the readings discussed earlier address the scheduling problem for coverage. But not the coverage problem, which ensures connectivity that we concentrate on to achieve. We analyze the sensor nodes' scheduling obstacle to observe the full coverage and connectivity.

3. Proposed Work. After the initial tree construction with currently selected active nodes, the nodes are categorized into two types, viz. the internal nodes($type_1$) and leaf nodes($type_2$). Together they maintain full coverage of the whole deployment area. So, they are also referred to as the coverage nodes. The internal nodes are always active as they receive data packets from their children and send it to the sink node. Nevertheless, the leaf nodes along with $type_1$ nodes do the job of sensing the event and sending messages to their parent. The leaf nodes stay in sleep mode most of the time. These leaf nodes turn active only when they perform the sense operation in case of an event, pass the message to their parent, and again go back to sleep mode.

From the above description, it can be seen that all the sensor nodes participate both in the event detection process as well as message passing. However, in a WSN with high concentration of nodes, either the sensing radius of sensor nodes are overlapping with each other or a combination of nodes entirely coincides with the sensing area of some other nodes. In such scenario, when all these nodes sense their surroundings and send the sensed data to their parent nodes, they perceive the same occurrence of the event and transmit the same data to the sink, which is entirely unnecessary. This leads to ineffective use of energy, where all the efforts by these nodes are useless at the expense of the tree's longevity.

To ensure full coverage of the deployed area, all the sensor nodes do not have to participate in event detection. Some of the nodes can completely remain in sleep mode and do not take part in any operation, and still the full coverage can be ensured. Such nodes remain a part of the tree but do not contribute in providing coverage. Here, those nodes are referred as the $type_0$ nodes. As long as a node is completely in sleep mode, it is equivalent to a dead node.

In WSNs, the sensing radius of most of nodes overlaps with each other. A node is said to be redundant, if its sensing area is within the sensing range of one or more sensor nodes. For area coverage in WSNs, the basic idea is to find the redundant nodes in a required area, which is already under the coverage of some other nodes, and put those redundant nodes into sleep mode. Keeping longevity and coverage in mind, in this paper, we have proposed a novel approach that initially constructs a hierarchical structure and periodically mends itself locally with consultation of nearby nodes based on nodes' level. It finds the redundant nodes in the vicinity of currently changing area only with the involvement of a minimum number of sensor nodes. Thus, full coverage is ensured only by engaging the nodes of the concerned area without involving all the nodes of the whole tree.

3.1. Construction. Based on initial tree construction, the sensor nodes are categorized into three types viz., $type_0$, $type_1$, and $type_2$ nodes. $Type_0$ nodes are the redundant sleep nodes which are currently in sleep mode and do not partake either in event detection or in area coverage. Their behavior is equivalent to a dead node. $Type_1$ nodes are the internal nodes and always remain active for a fixed duration. $Type_2$ nodes are called non-redundant leaf nodes that remains inactive but take part in the coverage of deployed area. Among these three types of nodes, $type_1$ nodes consumes more energy as compared to $type_0$ and $type_2$ nodes. During periodic tree reconstruction, the $type_1$ nodes, which meet the criteria to go to sleep, will be put into sleep mode, i.e., $type_1$ nodes will be turned to either $type_0$ or $type_2$ nodes depending upon the need of coverage of the locale. Moreover, before the reconstruction phase for the $(n+1)^{th}$ round, it is ensured that the tree was under full coverage and connectivity is maintained till the end of the n^{th} round.

We use the different abbreviations to describe the proposed protocol as given in Table 3.1.

For this proposed protocol, the sink node always remains active during the entire lifetime of the network. Before the beginning of $(n+1)^{th}$ round of tree reconstruction phase, a node $N(i)$, which was considered as

Table 3.1: Abbreviation

Short Name	Description
`N(i)`	i^{th} node
`SS(N(i))`	Sleep signal for i^{th} node
`RE(N(i))`	Remaining energy for i^{th} node
`LEVEL(N(i))`	Level of i^{th} node
`PT(N(i))`	No. of packets transfer by i^{th} node in a round
`ALIVE(N(i))`	True if i^{th} node is alive
`CONS(N(i))`	No. of consecutive rounds node i is active
`CRL(n)`	Current level reconstruct for the n^{th} round.
	CRL(n) = (CRL(n-1) +1) % max_level.
	A CRL value of 3 means level 1,2,3 will be reconstructed.
`CHECK_REDUNDANCY(N(k))`	k^{th} node checks for redundancy among its neighbors.

$type_1$ node in the n^{th} tree reconstruction phase and spends a significant amount of energy, can initiate its sleeping process based on satisfying any one of the following criteria.

1. Number of packets transmitted by node $N(i)$ in the current n^{th} round, i.e. represented as $PT(N(i))$, is greater than the threshold value.
2. Number of consecutive rounds the node $N(i)$ is active, i.e. represented as $Cons(N(i))$, is equal to three.
3. The level number of node $N(i)$ is less than or equal to the current level reconstruct for the n^{th} round, which is represented as $CRL(n)$. This $CRL(n)$ value plays an important role as the proposed protocol claims that rather than applying the sleeping process over all the nodes of WSN in each reconstruction phase, it applies on nodes present in certain level of the tree in different reconstruction phase. This is necessary because all the nodes in the different level of the tree do not consume energy in a uniform manner. The nodes which are closer to the sink consume more energy as compared to the nodes which are far away from sink. So, the nodes present in the higher level of the tree need to participate more frequently in tree reconstruction as compared to nodes in lower level. Considering the above necessity, we compute the $CRL(n) = (CRL(n-1)+1)\%max\ level$ that indicates up to which level the nodes will participate in n^{th} round of reconstruction phase. For example, if the CRL value is 3 then it means that nodes in level 1, 2, and 3 will participate in reconstruction phase.

Once, the $type_1$ node N(i) initiates its sleeping process, it sends a sleeping signal called SS(N(i)) to find new parents for all of its children before it is allowed to turn into $type_0$ or $type_2$ node. Then the node $N(i)$ starts with broadcasting a $FIND\_PARENT$ message packet to its children to find their new parent.

As per the algorithm-1, the active node N(i) waits for a random amount of time until all its children get a new parent before it turns into $type_0$ or $type_2$ node depending upon whether it is considered to be a redundant node with the support of its nearby $type_1$ and $type_2$ nodes. Once a node $N(j)$, which is a child of node $N(i)$, receives the $FIND\_PARENT$ packet, it initiates its process to find a new parent by broadcasting a WANT_PARENT packet and waits for a random amount of time to decide its new parent. A node N(k), which receives a WANT_PARENT packet, is eligible to become a parent only when it does not want to sleep, i.e. SS(N(k)) is false, and it is currently alive, i.e. ALIVE(N(k)) is true. Once the node N(k) meets its eligibility criteria, it replies back a PARENT_REPLY packet with including information such as level number and remaining energy which are represented by LEVEL(N(k)) and RE(N(k)) respectively. Now, N(k) is ready to become parent of any node if it is chosen by a node N(j), which is looking for a parent. A node N(j) might receive multiple PARENT_REPLY packets from different $type_0$, $type_1$ or $type_2$ nodes. Out of all the PARENT_REPLY packets from its neighboring nodes, the node N(j) chooses the most suitable node N(k) to be its parent based on the following criteria.

- Node N(j) will give a higher priority to a node N(k) to be its parent if LEVEL(N(k)) is equal to LEVEL(N(j))-1 rather than LEVEL(N(k)) is equal LEVEL(N(j)).

Nevertheless, it will out-rightly reject all nodes N(k) if LEVEL(N(k)) > LEVEL(N(j)).

- Bsed on above criteria, If node N(j) has more than one possible nodes then it selects its parent whose remaining energy is high. The node N(j) chooses N(k1) over N(k2), if LEVEL(N(k1)) is equal to LEVEL(N(k2)) and RE(N(k1)) > RE(N(k2)).

After node N(j) selects its parent from the multiple PARENT_REPLY packets, it sends an ACKNOWLEDGEMENT message to node N(k). When node N(k) receives the ACKNOWLEDGEMENT packet, it acts depending on the type of node it is. If it is a $type_0$ node then it turns to $type_1$ node and sets its CHECK_REDUNDANCY(N(k)) to true. As this node turns active from sleep mode, some of its $type_2$ neighbors might have the possibility to become redundant. So, N(k) performs a redundancy check after a random amount of time among its $type_2$ neighbors, and all nodes found to be redundant are turned to $type_0$ nodes from $type_2$ and put to sleep mode. Once the redundancy check gets over, node N(k) becomes an internal node and turns to $type_1$ node and remains active for the $n + 1^{th}$ round. If N(k) is $type_2$ node then it turns to $type_1$ node as $type_1$ and $type_2$ are both coverage nodes Inter-conversion between these two types of nodes does not require a redundancy check as the coverage was already ensured up to n^{th} round. If node N(k) is a $type_1$ node then it remains a $type_1$ node and does not do any redundancy check as it had already been done before when it turned to $type_1$ node in previous rounds.

After this process gets over, the node N(i), who was waiting for a random amount of time, wants to check whether all of its children have received a new parent or not. It carries out a sequence of events that comprise of message passing to and from its neighbors to determine whether it has been able to get rid of all its children. Still, if N(i) has a child, then it cannot be $type_2$ for the $n + 1^{th}$ round as it has not been able to lose all its child nodes. So N(i) has to remain $type_1$ for the $n + 1^{th}$ round too. N(i) again tries to go to sleep for the $n + 2^{th}$ round after the completion of the $n + 1^{th}$ round. However, if every child N(j) of N(i) can receive a new parent, N(i) gets rid of all its children, and changes its state to $type_2$ node for the $n + 1^{th}$ round.

Once the parent selection process gets completed, all the $type_1$ nodes with parameter CHECK_REDUNDANCY(N(k)) as true check for redundant nodes among its $type_2$ neighbors and the newly found redundant nodes are converted to $type_0$ and put them to sleep mode. When the $n + 1^{th}$ round starts, CONS(N(i)) for all the $type_1$ nodes is increased by 1 to keep track of the no of consecutive rounds a node is active. After each nodes state gets decided, $type_1$ and $type_2$ nodes can resume its task of event detection and send the data packets to the sink through the intermediate nodes. For each packet being sent to the sink node by node N(i), PT(N(i)) is increased by 1.

After a certain number of tree reconstruction phase, when the RE(N(i)) reaches below the threshold energy, then node N(i) needs to die permanently and will not be a part of the tree anymore. However, if N(i) is allowed to die immediately, it might lead to a coverage hole. If node N(i) is a $type_0$ node, it can be allowed to die immediately as it does not provide coverage anyway. If node N(i) is a $type_2$ node and it was instrumental in providing coverage then letting it die instantly, leads to a void in coverage. So, node N(i) broadcasts a TURN_T_2 message among its neighbors, which instructs all its $type_0$ neighbors to turn to $type_2$ to fill the void in coverage. In addition to this, node N(i) broadcasts a CHK_RDNCY message, which is meant for its $type_1$ neighbors. This sets the CHECK_REDUNDANCY(N(k)) as true for the $type_1$ neighbor such as N(k). Then, node N(k) performs redundancy check among its newly turned $type_2$ neighbors, and it turns all the redundant $type_2$ nodes to $type_0$, which was converted because of TURN_T_2 message from node N(i). N(i) does not take part in the redundancy check process of N(k). Once this process gets over, full coverage is maintained without N(i) being a part of it. So ALIVE(N(i)) is set off, i.e., it can die now without any issue.

4. Simulation Results. The proposed protocol performances have been measured using one of the popular simulator exclusive for WSN, called Castalia [31]. Around 1000 number of sensor nodes have been deployed randomly in a deployed area of 250x250m^2 to analyze the proposed approach. As per the universal standard mentioned in the TelosB data sheet [26], the radio transmission power, sensing range, etc., have been considered for this simulation purpose.

To fulfill the primary objective of WSN, i.e., sense the whole deployed area along with maintaining network lifetime, the proposed level-wise tree construction approach includes the area coverage into consideration to fulfill the above two requirements. To validate the efficiency of the proposed algorithm with respect to network lifetime, in figure 4.1, we compare the number of nodes alive after each tree reconstruction round in the proposed

Algorithm 1: In The tree reconstruction phase between two round

```
Function In TIMER_2:
    if N(i) is $type_1$ AND sleep constraint satisfied then
        starts the process to sleep;(set timer of start_process)
    else
        do nothing until next round
    if (ALIVE(N(i))==FALSE AND N(i) is $type_2$ then
        Broadcast TURN_T_2 message;
        Broadcast CHK_RDNCY message
    Trigger TIMER_3 after a random time;
Function In start_process:
    Send FIND_PARENT packet to notify N(j) (their child) that they need to find new parent;
    After a random time interval it triggers "determine_is$type_1$";
if N(j) receives FIND_PARENT packet then
    if ALIVE(N(j)) AND [PFj,1==N(i) OR PFj,2==N(i)] then
        Broadcasts a WANT_PARENT packet;
        Waits a random time while it chooses the best parent reply.
Function In TIMER_3:
    Trigger FIND_REDUNDANT function;
    Send packets to Sink node(if $type_1$ and $type_2$);
    TIMER_2 after a certain time(next round);
if N(k) receives WANT_PARENT packet then
    if (ALIVE(N(k)) == true and SS(N(k)) == false) and (LEVEL(N(k)) == LEVEL(N(j)) or LEVEL(N(k)) +1
      == LEVEL(N(j)) then
        broadcast PARENT_REPLY packet
if N(j) receives PARENT_REPLY packet then
    if if N(j) wants new parent or PFj,1/PFj,2 is -1) then
        if [LEVEL(N(k+1))==LEVEL(N(l))] OR [if(RE(N(k)) >RE(N(l))) AND (LEVEL(N(k))==LEVEL(N(l)))]
          then
            Select node N(k) as parent over N(l);
            Send ACKNOWLEDGEMENT packet to its newly selected parent N(k)
if N(k) receives ACKNOWLEDGEMENT packet then
    if N(k) is $type_1$ node then
        Remains $type_1$
    if N(k) is $type_2$ node then
        Turns to $type_1$
    if N(k) is $type_0$ node then
        CHECK_REDUNDANCY(N(K))=True;
        turns to $type_1$
if N(l) receives TURN_T_2 packet then
    if N(l) is $type_0$ then
        turn to $type_2$
if N(m) receives CHK_RDNCY packet then
    if N(l) is $type_1$ then
        CHECK_REDUNDANCY(N(l))= True
Function In determine_is$type_1$:
    Determine number of child through a sequence of message if no_of_child >0 then
        remains $type_1$
    else
        turns $type_2$
Function In determine_is$type_1$:
    if CHECK_REDUNDANCY(N(i))= True then
        check redundancy among $type_2$ neighbour nodes
```

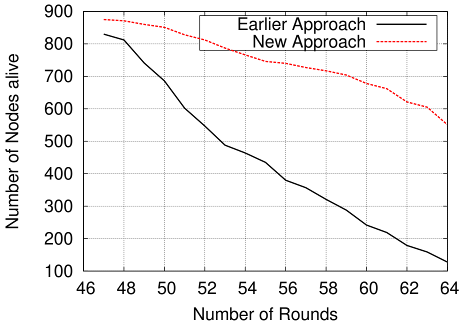

Fig. 4.1: Number of nodes alive after each tree reconstruction round in the normal level-wise tree construction approach [27] vs the proposed level-wise tree construction along with coverage.

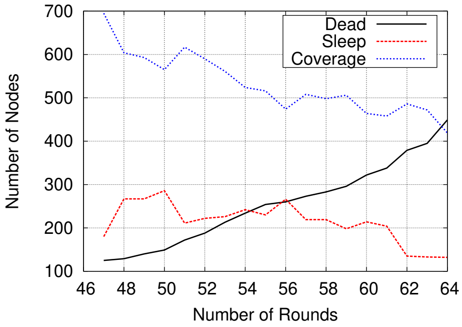

Fig. 4.2: Number of dead nodes, sleep nodes and coverage node after each each tree reconstruction rounds.

approach compared to the normal level-wise approach given [27]. From this figure, it is observed that after 46^{th} round, the nodes start dying out in both the earlier approach and the proposed approach. At 64^{th} round, in the previous approach, almost all nodes have died out, whereas in the case of the proposed method, after the same number of tree reconstruction rounds, the number of nodes dies out is nearly 50%. So, the network lifetime in the proposed approach is almost double as compared to the earlier approach. It indicates that the proposed method not only increases the network lifetime but also achieving area coverage.

Figure 4.2 shows that after the nodes start dying out, how the alive nodes become segregated into completely sleep node and coverage nodes in each tree reconstruction round. Here, the coverage nodes include both intermediate nodes, which are completely active, and the leaf nodes, which are in sleep mode. When an event occurs in their surroundings, those nodes wake up, complete the event detection process, and again go back to sleep mode. From this figure, it is observed that with an increase in the number of dead nodes, both the coverage node and sleep nodes decreases. Here, one interesting observation is that with the rise in the number of dead nodes, the area coverage is even managed with fewer nodes up to a certain tree reconstruction round. As per the figure, it is 64^{th} round. After onwards, even with the total number of alive nodes, the complete coverage could not be achieved.

Figure 4.3 depicts the number of nodes present in each level of the tree after the nodes start dying out as the tree reconstruction round progresses. From this figure, it is observed that the level 2 and 3 have the maximum number of nodes, and the rate of dying out of nodes in level 1 is more as compared to level 2, which is again greater than level 3. The nodes closer to the sink spend more time in data transmission and thus consume more energy than the nodes far away from the sink.

As per the proposed approach, when the $type_1$ nodes want to sleep, the children of those nodes try to select their new parents from the group of nodes, i.e., either a leaf node or a sleep node, or an intermediate node in the previous tree reconstruction round. Figure 4.4 shows that the number of sleep nodes, intermediate nodes,

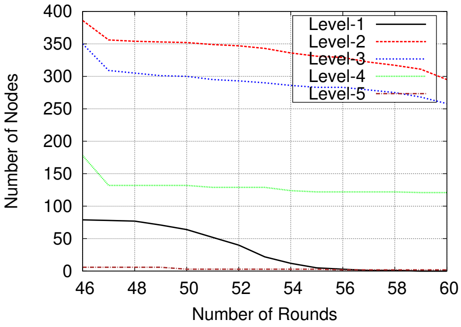

Fig. 4.3: Number of nodes present in each level of the tree in different tree reconstruction rounds.

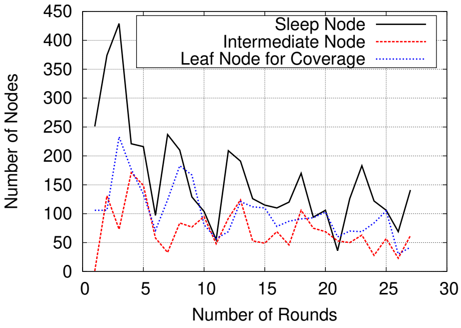

Fig. 4.4: Number of nodes select as parent in the next round of tree reconstruction from different group of nodes (sleep nodes, intermediate node, and leaf nodes).

and active nodes is selected as new parents by the nodes whose current parents are interested to sleep in the next tree reconstruction round. It is also observed that the nodes, which are looking for a parent, always have a higher tendency to choose $type_0$ nodes over $type_1$ & $type_2$ nodes.

Figure 4.5 depicts the number of $type_0$, $type_1$, and $type_2$ nodes in each level of tree at different round of tree reconstruction. In a network size of 1000 nodes, the number of levels comes up to 5. The majority of the nodes are dominated by $type_0$ and $type_2$ nodes. During the initial rounds of tree reconstruction, the number of $type_2$ nodes is more than $type_0$ nodes. As the round progresses, the tree adjusted to show a significant increase in $type_0$ nodes and a decrease in $type_2$ nodes.

In order to evaluate the longevity of the proposed protocol, we simulated both the earlier approach [27] as well as the proposed approach to identify the number of sleep nodes present in different rounds of the tree reconstruction as shown in figure 4.6. This figure ensures that the network lifetime achieved through the proposed approach is far better than the earlier level-wise tree construction approach. This enhancement is achieved as the coverage through the proposed approach is ensured with fewer nodes.

Figure 4.7 compares the number of sleep nodes, coverage leaf nodes, and intermediate nodes in different rounds of tree reconstruction. From this figure, it is observed that in the initial few rounds of tree reconstruction, $type_0$ nodes increase continuously, and coverage nodes decreased. Nevertheless, after the 46th round, more than 150 nodes died at once. So, there is a rise in the coverage nodes and drop in sleep nodes, as many sleep nodes turn to coverage nodes to compensate for the loss of so many coverage nodes.

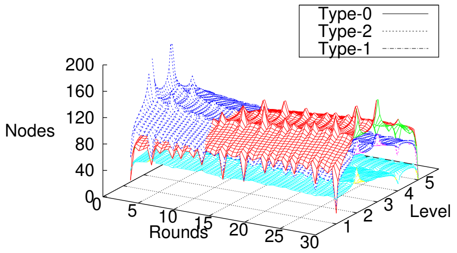

Fig. 4.5: Different types of nodes at different level of tree in different round tree reconstruction

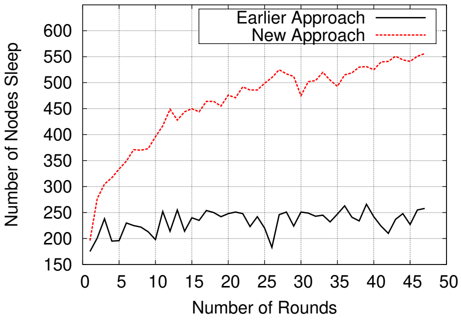

Fig. 4.6: Number of sleep nodes in different round of tree reconstruction in proposed approach vs. normal level-wise approach given in [27].

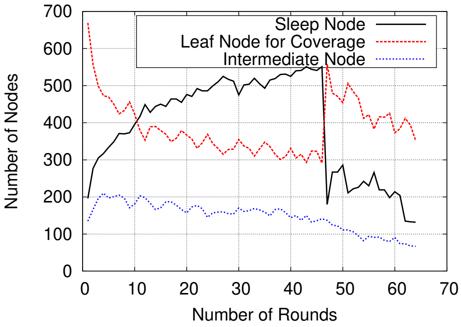

Fig. 4.7: Comparison of number of sleep nodes, coverage leaf nodes, and intermediate node in different round of tree reconstruction.

5. Conclusion. In this paper, we addressed the problem of scheduling sensor activities while having full coverage. Improving the network lifetime by increasing the number of sleep nodes along with maintaining full area coverage is the purpose of this research. We approached the least energy exhaustion provision in WSNs, proposed a level-wise sleep scheduling mechanism, and maintained coverage to achieve the above requirement. In this proposed approach, a routing tree is built that maintains coverage, and at the same time, longevity can be achieved by putting the redundant nodes in the sleep mode. The tree mends itself locally by following a level-wise approach for the tree reconstruction and finds the redundant nodes in the vicinity of the changing area only with the engagement of a minimum number of nodes in the process. Thus, full coverage is ensured only by engaging the nodes of the concerned area without involving all the nodes of the tree. The simulation results also ensured improved network lifetime by making more nodes in the sleep state.

REFERENCES

[1] J. Zheng and A. Jamalipour, Wireless Sensor Networks A Networking Perspective. A JOHN WILEY and SONS, Hoboken, New Jersey, 2009

[2] Z. Chuan, Z. Chunlin, S. Lei, H. Guangjie, A survey on coverage and connectivity issues in wireless sensor networks. J. Network Comput. Appl. 35, 619-632, 2012, Elsevier.

[3] Mulligan, R., Ammari, M.H., Coverage in wireless sensor networks: a survey. Network Protocols Algorithms 2 (2), 2010, 27-53.

[4] Amit, G., Sajal, D., Coverage and connectivity issues in wireless sensor networks: a survey. Sci. Direct, Pervasive Mob. Comput. 4, 303-334, 2008.

[5] Winston, J., Paramasivan, B., 2011. A survey on connectivity maintenance and preserving coverage for wireless sensor networks. Int. J. Res. Rev. Wireless Sensor Networks (IJRRWSN) 1 (2), 11-18.

[6] Fan Ye, F., Zhong, G., Cheng, J., Lu, S., Zhang, L., 2002. PEAS: a robust energy conserving protocol for long-lived sensor networks. In: 23rd International Conference on Distributed Computing Systems(DCS' 02), pp. 28-37.

[7] Gui, C., Mohapatra, P., 2004. Power conservation and quality of surveillance in target tracking sensor networks. In: Proceedings of 10th Annual International Conference on Mobile Computing and Networking (ACM MobiCom' 04), Pennsylvania, USA, pp. 129-143.

[8] Yen, Y., Hong, S., Hang, R., Chao, H., 2007. An energy efficient and coverage guaranteed wireless sensor network. In: Proceedings of IEEE Wireless Communications and Networking Conference, WCNC, pp. 2923-2930.

[9] More, A., Raisinghani, V., 2014. Random backoff sleep protocol for energy efficient coverage in wireless sensor networks. In: Kumar Kundu, M., Mohapatra, P.D., Konar, A., Chakraborty, A. (Eds.), Advanced Computing, Networking and Informatics, Wireless Networks and Security Proceedings of the Second International Conference on Advanced Computing, Networking and Informatics (ICACNI-2014), vol. 2. Springer, pp. 123-131.

[10] S. Slijepcevic and M. Potkonjak, "Power efficient organization of Wireless Sensor Networks", in Proc. of International Conference on Communications (ICC'01). IEEE, June 2001, pp. 472–476.

[11] C. F. Huang and Y. C. Tseng, "The coverage problem in a wireless sensor network," in Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '03), pp. 115–121, San Diego, Calif, USA, September 2003.

[12] Z. Zhou, S. Das, H. Gupta. "Connected K-Coverage Problem". Proceed- ings of the International Conference on Computer Communications and Networks (IC3N), 2004.

[13] H. Mousavi, A. Nayyeri, N. Yazdani and C. Lucas. "Energy conserving movement assisted deployment of Ad hoc sensor networks". IEEE Comm. Lett., 10(4): 269-271, 2006.

[14] G. Fletcher, X. Li, A. Nayak, and I. Stojmenovic. "Randomized Robot- assisted Relocation of Sensors for Coverage Repair in Wireless Sensor Networks". IEEE Comm. Lett, 2010.

[15] X. Li, G. Fletcher, A. Nayak, and I. Stojmenovic. "Randomized carrier based sensor relocation in wireless sensor and robot networks". Ad Hoc Networks. 11: 1951-1962, 2013.

[16] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sen- sor networks," in Proc. of IEEE International Conference on Communications (ICC), 2001, vol. 2, pp. 472–476.

[17] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li and D.-Z. Du, "Wireless sensor networks with energy efficient organization," Journal of Inter- connection Networks, vol. 3, no. 3-4, pp. 213–229, 2002.

[18] D. Tian and N. D. Georganas, "A coverage preserving node scheduling scheme for large wireless sensor networks," in Proc. of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), 2002.

[19] Gao, Shan, Chinh T. Vu, and Yingshu Li. "Sensor scheduling for k-coverage in wireless sensor networks." International Conference on Mobile Ad-Hoc and Sensor Networks. Springer, Berlin, Heidelberg, 2006.

[20] Tian, Di, and Nicolas D. Georganas. "A coverage-preserving node scheduling scheme for large wireless sensor networks." Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications. 2002.

[21] Cardei, Mihaela, et al. "Energy-efficient target coverage in wireless sensor networks." Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.. Vol. 3. IEEE, 2005.

[22] Yan, Ting, Tian He, and John A. Stankovic. "Differentiated surveillance for sensor networks." Proceedings of the 1st international conference on Embedded networked sensor systems. 2003.

[23] Alfieri, Arianna, et al. ”Exploiting sensor spatial redundancy to improve network lifetime [wireless sensor networks].” IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.. Vol. 5. IEEE, 2004.

[24] Pottie, Gregory J., and William J. Kaiser. ”Wireless integrated network sensors.” Communications of the ACM 43.5 (2000): 51-58.

[25] Raghunathan, Vijay, et al. ”Energy-aware wireless microsensor networks.” IEEE Signal processing magazine 19.2 (2002): 40-50.

[26] Telosb data sheet. http://www.xbow.com/Products/ Product pdf files/Wireless pdf/TelosB Datasheet.pdf

[27] N. Tarasia, A. R. Swain, S. Roy and U. N. Kar, “A Level-Wise Periodic Tree ConstructionMechanism for Sleep Scheduling in WSN,” Journal of Communications Software and Systems 16, no. 2 (2020): 113-121.

[28] Mhatre, Kavita Prashant, and Uday Pandit Khot. ”Energy Efficient Opportunistic Routing with Sleep Scheduling in Wireless Sensor Networks.” Wireless Personal Communications 112.2 (2020): 1243-1263.

[29] Fang, Wei, Mithun Mukherjee, Lei Shu, Zhangbing Zhou, and Gerhard P. Hancke. ”Energy utilization concerned sleep scheduling in wireless powered communication networks,” In 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 558-563. IEEE, 2017.

[30] Guruprakash, B., C. Balasubramanian, and R. Sukumar. ”An approach by adopting multi-objective clustering and data collection along with node sleep scheduling for energy efficient and delay aware WSN.” Peer-to-Peer Networking and Applications 13.1 (2020): 304-319.

[31] *Castalia a simulator for wireless sensor networks*, http://castalia.npc.nicta.com.au/pdfs/Castalia User Manual.pdf.

Edited by: Dana Petcu
Received: Dec 21, 2020
Accepted: Jan 21, 2021