CHAPTER 4

EXTENDING THE NETWORK LIFETIME OF WIRELESS SENSOR NETWORK USING RESIDUAL ENERGY EXTRACTION - HYBRID SCHEDULING ALGORITHM

4.1 INTRODUCTION

Wireless sensor networks (WSNs) are mostly deployed in a remote working environment, since sensor nodes are small in size, cost-efficient, low-power devices, and have limited battery power supply. The network lifetime mainly depends on the battery lifetime of the node. The main concern is to increase the lifetime with respect to energy constraints. One way of doing this is by turning off redundant nodes to sleep mode to conserve energy while active nodes can provide essential $k$-coverage, which improves fault-tolerance. Hence, we use scheduling algorithms that turn off redundant nodes after providing the required coverage level $k$. The scheduling algorithms can be implemented in centralized or localized schemes, which have their own advantages and disadvantages. To exploit the advantages of both schemes, we employ both schemes on the network according to a threshold value. This threshold value is estimated on the performance of WSN based on network lifetime comparison using centralized and localized algorithms. To extend the network lifetime and to extract the useful energy from the network further, we go for compromise in the area covered by nodes.
4.2 PROPOSED ALGORITHMS

The algorithms used here assume a two dimensional area with randomly deployed set of sensors \( S = \{s_1, s_2, \ldots, s_n\} \) with a fixed sensing range \( r \). In this proposed work we used three parts to extend the network lifetime for the sensor networks. The first part deals with finding out the coverage of each sensor. The second part deals with the disjoint set formation/node scheduling algorithm development. Third part deals with the employment of the centralized algorithm in a given topology with different percentage of area covered.

4.2.1 Algorithm to Find \( k \)-Coverage

Consider a set of sensors, \( S = \{s_1, s_2, \ldots, s_n\} \), in a two dimensional area \( A \). Each sensor \( s_i, i = 1,2, \ldots, n \) is located at coordinate \((x_i, y_i)\) inside \( A \) and has a sensing range of \( r_i \), i.e., it can monitor any object that is within a distance of \( r_i \) from \( s_i \). A location in \( A \) is said to be covered by \( s_i \) if it is within \( s_i \)'s sensing range. A location in \( A \) is said to be \( j \)-covered if it is within at least \( j \) sensors’ sensing ranges. Below, we use an algorithm to determine whether a sensor is \( k \)-perimeter-covered or not. Consider two sensors \( s_i \) and \( s_j \) located in positions \((x_i, y_i)\) and \((x_j, y_j)\) respectively. The distance between two sensors is calculated using the distance equation (4.1)

\[
d(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4.1}
\]

If \( d(s_i, s_j) > 2r \) then \( s_j \) does not contribute any coverage to \( s_i \)'s area coverage. The fraction of coverage is given by finding the segment of \( s_i \), overlapping with \( s_j \) using the central angle. The central angle can be noted
from Figure 4.1 which can be derived by considering the incremental perimeter of sensor $s_i s_j$ ($\Delta ps_i s_j$). Using the cosine rule the central angle can be found which can be obtained from equation (4.2).

Where $\Delta ps_i s_j$ - incremental perimeter of sensor $s_i s_j$

\[
\cos \alpha = \frac{ps_i + s_j - ps_j}{2(ps_i) (s_i s_j)} \tag{4.2}
\]

\[
\alpha = \arccos \left( \frac{d(s_i s_j)}{2r} \right) \tag{4.3}
\]

**Figure 4.1 Central Angle**

This can be repeated for all sensors that overlap with $s_i$ as follows.

1. For all neighboring sensors $s_j$ of $s_i$ such that $d(s_i s_j)>2r$, place the points $\alpha_j, L$ and $\alpha_j, R$, were $R$ on the line segment $[0, 2\pi]$ and sort all these points in an ascending order into a list $L$. Properly mark each point as a left or right boundary of a coverage range.

2. Traverse the line segment $[0, 2\pi]$ by visiting each element in the sorted list $L$ from left to right and if $\alpha_j, L$ comes add the
coverage by 1 and if $\alpha_j, R$ comes decrement the coverage by 1 and thus determine the coverage of $s_j$.

## 4.2.2 Scheduling Algorithm

Energy is a restricted resource for sensor nodes, which determines how long and how well sensor networks can work. Most energy efficient approaches use the same way to save energy, reducing the number of sensors working simultaneously. By scheduling redundant sensors to go to sleep while the essential coverage has been satisfied, the network lifetime can be significantly prolonged.

### 4.2.2.1 Centralized Scheduling Algorithm

The network lifetime is defined as the total duration during which the whole area is $k$-covered. Sensor Scheduling for $k$-Coverage (SSC) can be defined as a sensor network with $n$ sensors that can provide $k$-coverage for the monitored area, schedule the activities of the sensors such that at any time the whole area can be $k$-covered and the network lifetime is maximized.

The scheduling decisions can be made at the Base Station (BS). The BS broadcasts the schedule to all the sensors so that each sensor can know when it should be active to monitor the area. Hence this scheduling is called the Centralized Scheduling for $k$-coverage Algorithm (CSKA). To solve the SSC problem, divide the sensors into disjoint or non-disjoint subsets and each subset $k$-covers the whole area, where $k$-cover indicates for every point in the area, it is covered (monitored) by at least $k$ sensors. These subsets can be scheduled to be active successively. For each subset, its lifetime is determined by the sensor which has the least power.
Consider sensor \( s_i \) is \( k \)-perimeter covered if all points on the perimeter of \( s_i \) are covered by at least \( k \) sensors, which are in the same set with \( s_i \), other than \( s_i \) itself. Similarly, a segment of \( s_i \)'s perimeter is \( k \)-perimeter-covered if all the points on the segment are covered by at least \( k \) sensors, which are in the same set with \( s_i \), other than \( s_i \) itself. The Perimeter Coverage Level (PCL) of a sensor \( s_i \) is defined as the number of sensors in the same set with \( s_i \), which cover any point on \( s_i \)'s perimeter of the sensing area. The lower the PCL, the smaller will be the node density.

Considering the case where each sensor has a fixed sensing range and all the sensors are divided into disjoint cover sets, the objective of the proposed work is to construct as many subsets as possible such that (i) each subset can \( k \)-cover the whole monitored area; (ii) the network lifetime is maximized.

![Figure 4.2 PCL Greedy Selection Algorithm](image-url)
Figure 4.2 shows the explanation of PCL Greedy Selection Algorithm which is used to form disjoint subsets. The inputs of this algorithm are the required coverage level $k$, the sensor set $\{S\}$. The main idea is to iteratively construct subsets $C_i$ by choosing sensors from the region with the lowest sensor density. When constructing an individual $C_i$, at each step, the sensor with the smallest PCL value is added to $C_i$. In this way, include as few sensors as possible in $C_i$ and these sensors are distributed in the area as widely as possible because they are from the regions with the lowest sensor density, such that more sensors can be left to join other subsequent subsets and the overlapped sensing regions in each subset are reduced as much as possible. This also indicates that when constructing a subset $C_i$, the region with smaller node density is taken care of with higher priority. The output of this algorithm is the number of disjoint subsets $\{C\}$ which provide essential $k$-coverage.

The input of this algorithm includes $k$ the user-specified coverage level, and the set of all the sensors $\{S\}$. The output is $C_i$, a set of subsets, and each subset can $k$-cover the whole area. To verify whether a subset $C_i$ can $k$-cover the entire surveillance area, the method used is proposed in Chi-Fu Huang et al. (2007) and is called as get Coverage Level $C_i$.

Figure 4.3 Perimeter Coverage of a Sensor $s_i$
Figure 4.3 shows the perimeter coverage of sensor $s_i$. For each sensor $s_i$, it tries to identify all neighbouring sensors which can contribute some coverage to $s_i$'s perimeter. Specifically, for each neighbouring sensor $s_j$, the angle of $s_i$'s arch is determined and is denoted by $(\alpha_j, L, \alpha_j, R)$, which is perimeter-covered by $s_j$. Placing all the angles $(\alpha_j, L, \alpha_j, R)$ on $(0, 2\pi)$ for all $j$'s, it is easy to determine the level of perimeter coverage of $s_i$. For example, Figure 4.3(a) shows how $s_i$ is covered by its neighbour nodes is shown in dashed circles. Mapping these covered angles decide that $s_i$ is perimeter covered is shown in Figure 4.3(b).

After coverage verification, all the sensors in $\{S\}$ are sorted in non-decreasing order based on their PCL values. Then sensors are added into a subset in a greedy manner. After some iteration, the current subset $C_i$ can provide $k$-coverage, a new subset $C_{i+1}$ will construct in the same manner. PCL Greedy selection stops when we can no longer construct subsets which can $k$-cover the whole surveillance area.

The subsets formed using PCL algorithm will have redundant sensors since each node is added to subset in a greedy manner. These redundant sensors are identified and removed and added back to $\{S\}$, so that they are still available to be added to the subsequent subsets. This operation is done using the subroutine called Prune Greedy Selection Algorithm which is shown in Figure 4.4. In this algorithm, given a subset $C_i$ each sensor in $C_i$ is checked to see whether sensor coverage $s_{cov}$ is smaller than the user specified $k$ value by removing each and every sensor in that subset.
If a sensor is redundant it will be removed from $C_i$ and added back to $\{S\}$ so that it can be used to form other subsets. This PCL Greedy Selection algorithm and Prune Greedy Selection algorithm together constitute the Centralized Scheduling Algorithm (CSA). This CSA is first employed to form disjoint subsets such that every subset covers 95% of the surveillance area. When the network fails, some /most of the nodes may have enough energy to work more.
4.2.2.2 Localized Scheduling for K Coverage Algorithm (LSKA)

LSKA works in a rounding fashion as in Figure 4.5 with the round length of $dRound$, that is each sensor runs this algorithm every $dRound$ unit of time. At the beginning of each round is a decision phase with the duration of $W$. There is several advantages of working in rounds.

(i) $k$ can be dynamically changed, for some applications, such as forest fire, the value of $k$ need to be changed while the network is running. For example, in the dried season, there is more chance of fire happening, thus the value of $k$ needs to be high. However, in the rainy season, that chance is small, so the value of $k$ needs to be small to save network energy. Also, the operation of the network does not need to be interrupted while $k$ is being changed.

(ii) LSKA supports robustness at each round, there is exactly one set cover, in responsible for sensing task. In a situation that some sensors in that set cover are out of service (may die), then the sensing data will be affected and network may temporarily not provide $k$-coverage for some interval of time. However, this problem will not affect prolong since the new set cover will be discovered at the next round to take charge of sensing task.

(iii) Energy Efficient algorithm Besides, LSKA is an energy-efficient distributed algorithm which requires only 1-sensing hop neighbour information and it also provides $k$-coverage for the whole network (which is a kind of fault-tolerance). Thus, LSK algorithm satisfies all the requirements of a sensor network protocol.
All the sensors have to decide its status in the decision phase. The decision time $W$ is the time taken to compute the status once. The status of the sensors can be $ON$ or $OFF$ and each sensor decides its status sequentially and informs the status through the "HELLO" messages to its neighbours. Each neighbour finds its neighbours and updates the value using the already received "HELLO" messages.

![Decision Phase and Sensing Phase](image)

**Figure 4.5 Localized Scheduling Rounds**

In the sensing phase, the sensors that decided to remain $ON$ begin to monitor the area and sends the updated data messages. The "HELLO" messages consume a considerable amount of energy. Hence to eliminate this, it is assumed to be $dRound \gg W$. Here $dRound$ is assumed to be 15 times of $W$.

The Radio energy model that is being used in our analysis is shown in Figure 1.1. A simple model is assumed where the transmitter dissipates energy to run the radio electronics and the power amplifier and the receiver dissipates energy to run the radio electronics.
4.3 SIMULATION SETUP

Simulation setup for this algorithm is variable size network both of grid and random topologies are used where nodes were distributed between \((x = 0, y = 0)\) and \((x = 100, y = 100)\) with the Base Station at location \((x = 50, y = 50)\). Each data message is 64 bytes long and hello messages are 200 bits long. The power attenuation is dependent on the distance between the transmitter and receiver. For relatively short distances, the propagation loss can be modeled as inversely proportional to \(d^2\), whereas for longer distances, the propagation loss can be modeled as inversely proportional to \(d^4\). In this simulation, the free space channel models were used. The Electronic energy, \(E_{elec}\) is the energy consumed in the electronics circuit to transmit or receive the signal which depends on factors such as the digital coding, modulation and filtering of the signal before it is sent to the transmit amplifier. The amplifier energy \(E_{amp} \times d^2\) depends on the distance to the receiver and the acceptable bit-error rate. It is assumed that \(d^2\) energy loss is due to channel transmission. The simulation described in this proposed work is based on the communication energy parameters shown in Table 1.1.

Table 4.1 Simulation Setup

<table>
<thead>
<tr>
<th>Network Area</th>
<th>100m X 100 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensing Range</td>
<td>10m -100m</td>
</tr>
<tr>
<td>Initial Energy of each Node</td>
<td>2KJ</td>
</tr>
<tr>
<td>Decision Phase</td>
<td>2 seconds</td>
</tr>
<tr>
<td>Slot time</td>
<td>0.5 ms</td>
</tr>
<tr>
<td>Round time</td>
<td>20 minutes</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>20 - 100</td>
</tr>
</tbody>
</table>
Transmitter:

\[ E_t(k,d) = E_{elec} \times k + \delta_{amp} \times k \times d^2 \quad ; \text{if } d < d_c \tag{4.4} \]

\[ E_{elec} \times k + \delta_{amp} \times k \times d^4 \quad ; \text{otherwise} \tag{4.5} \]

Receiver

\[ E_r(k,d) = E_{elec} \times k \tag{4.6} \]

where \( d_c \) the cross is over distance, \( k \) is the packet bit size and \( d \) is the distance between transmitter and receiver antennas.

### 4.4 SIMULATION RESULTS

The proposed work involves the employment of CSKA and LSKA in the networks of various sizes using the energy model is represented in equations (4.3) to (4.6). Most of the existing researches, the extension of lifetime of sensor networks is mainly depends on either CSKA or LSKA algorithms alone. But both algorithms have their own advantages and disadvantages. Hence to exploit the advantages of both algorithms this proposed uses CSKA and LSKA in the same network. For this a threshold should be found when to switch from one algorithm to next in order to maximize the network life time. The simulation setup for proposed work is shown in Table 4.1.

#### 4.4.1 Deployment and Subset Formation

Figure 4.6 shows the sample for a random deployment of 100 nodes in the two dimensional area of 100x100m which monitors the entire coverage region with redundant nodes. These redundant nodes can be used to form subsets such that every subset can cover the entire surveillance area. The
assumptions are made such that each sensor has the sensing range of 3m and communication range as twice the sensing range to ensure connectivity. By using PCL and Prune Greedy Algorithm six disjoint subsets are formed from the 100 deployed nodes. The Table 4.2 shows the disjoint subsets. From the subsets formed, it is found that each sensor participates in only one subset and hence the name disjoints subsets.

![Figure 4.6 Sample Network for Random Deployment of 100 Nodes](image)

### Table 4.2 Disjoint Subsets

<table>
<thead>
<tr>
<th>Subset</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subset 1</td>
<td>14 19 20 42 44 55</td>
</tr>
<tr>
<td>Subset 2</td>
<td>13 27 1 17 62</td>
</tr>
<tr>
<td>Subset 3</td>
<td>32 45 51 58 59 63 66</td>
</tr>
<tr>
<td>Subset 4</td>
<td>3 68 70 77 78 8</td>
</tr>
<tr>
<td>Subset 5</td>
<td>69 23 24 29 57 74</td>
</tr>
<tr>
<td>Subset 6</td>
<td>37 38 67 11 46 35 36 65</td>
</tr>
</tbody>
</table>
4.4.2 Number of Nodes and Subsets

From the Figure 4.7, it is found that the number of subsets increases as the number of nodes increases. Also when number of node is 60 and $k = 1$, 10 disjoint subsets and $k = 2$ only 3 disjoint subsets is formed. The plot infers that as $k$ increases the number of subsets decreases since more sensors are needed per subset to achieve the required coverage level.

![Number of Subsets for Various Levels of k](image)

**Figure 4.7 Number of Nodes Based on Subsets**

4.4.3 Network Lifetime Comparison with CSK Algorithm

Figure 4.8 infers that the network lifetime is increased by using the CSKA algorithm. The network lifetime is given in terms of rounds till which all the subsets are died. A subset is assumed to be useless when one of the sensor’s energy is below the threshold value. Also it shows that if the
coverage level increases from \( k = 1 \) to \( k = 2 \), the network lifetime decreases. This is because the number of nodes in the subset increases if the required coverage level \( k \) increases. The results show that without scheduling, the lifetime of the network is \( 0.2 \times 10^4 \) time slot and with scheduling algorithm when coverage level is \( k = 2 \) the lifetime of the networks is \( 4 \times 10^4 \) time slot, when the number nodes deployed in the surveillance area is 100. By using scheduling algorithm the lifetime of the network is 20 times improved than without scheduling scheme.

![Network lifetime comparison with CSA-k Coverage Algorithm](image)

**Figure 4.8 Deployed Sensors Vs Network Lifetime (CSKA)**

### 4.4.4 Energy Consumption

Figure 4.9 shows that CSKA algorithm reduces energy consumption due to the less number of redundant neighbours and hence by using scheduling algorithm 58.6% energy is saved compared to without scheduling scheme.
Figure 4.9 Energy Consumption of Network Based on Number of Nodes

4.4.5 Sensing Range and Subsets

Figure 4.10 illustrates that, as the sensing range increases the number of subsets formed is also increases. This happens because only less number of sensors is needed to provide the same coverage. That is with larger sensing ranges, the number of sensors in each subset decreases such that more subsets can be constructed. When the sensing range is 50m and the number of node is 100, then the number of disjoint subset is 6 and when the sensing range is 100m, the number of disjoint subset is 16 because of less number of sensors in each subset.
4.4.6 Sensing Range and Number of Sensors per Subsets

Figure 4.11 shows that as the sensing range increases number of sensors covered by each subset get reduced because only less number of sensors is needed to provide the same coverage. When the number of sensor deployed in the application area is 100 and sensing range is 50m only 10 numbers of sensors are covered by each subset where as when the sensing range is 100m for same number of sensors are deployed in the application area 2 sensors are covered by each subset. From this we know that as the sensing range increases number of sensors in the subset gets reduced.
4.4.7 Residual Energy Extraction and Node Utilization by Varying Area Coverage Levels

Using disjoint subset algorithms only 30% of deployed sensors are utilized in subset formation and remaining nodes are not used. Hence to utilize the remaining 70% of sensor nodes the proposed method opted to go for the compromise in coverage area. Thus by compromising in coverage area more number of nodes are utilized, residual energy is reduced and network lifetime is improved.

4.4.7.1 Number of Utilized Sensors

However from Figure 4.12, it is observed that when centralized scheduling algorithm is applied, number of utilized nodes are very much less
than the available nodes and hence to use the all or most of the deployed sensors, as a part of the work an idea to go for compromise in area coverage is developed here.

![Graph showing number of utilized sensors vs number of deployed nodes.](image)

**Figure 4.12 Number of Utilized Sensors**

### 4.4.7.2 Lifetime Comparison with Varying Area Coverage Levels

First the subset set is formed by using 95% of area coverage. When all the subsets are dead, the percentage of area coverage level is changed from 95% to 85% percentages. This ensures the usage of the nodes that are not used in the earlier rounds. This process is repeated till the acceptable level of coverage that can be used. Here it is assumed that the monitoring service requires at least 50 percentage of coverage. The Figure 4.13 shows the improvement in network lifetime by employing the CSKA to variable area coverage levels. If 95% of area alone is considered, only less number of nodes
is used in subset formation and in monitoring service. Even though there is some nodes available with enough energy, the network is announced dead. To avoid this residual energy is extracted from the nodes that are still alive.

![Network Lifetime With Varying Coverage Levels](image)

**Figure 4.13 Lifetime Comparison with Varying Area Coverage Levels**

### 4.4.7 Network Lifetime Comparison for $k=1$

As explained already the aim of the proposed work is to find the threshold to switch from CSKA to LSKA and vice versa to improve the network lifetime of the WSN. This can be done by employing CSKA and LSKA on the networks of various sizes deployed in the same area. The Figure 4.14 shows the network lifetime comparison for $k=1$ with and without applying scheduling algorithms. The lifetime in localized scheme is lesser compared to the centralized scheme due to the frequent exchange of hello packets. The graph illustrates that for the given conditions the localized
scheduling algorithm provides better results when the network size is less than 60 nodes and above that centralized takes over.

Figure 4.14 Network Lifetime Comparisons for $k = 1$

4.4.9 Network Lifetime Comparison for $k=1, k=2$

The Figure 4.15 shows the relationship between network lifetimes, desired coverage level $k$, network size threshold value to switch from centralized to localized and vice versa. From the figure it is found that when $k$ increases the network lifetime decreases. Also it comes into light that as $k$ increases the network size threshold increases since more number of nodes are used to cover the monitored area with increased coverage level ($k=2$). For $k=1$, the threshold is found to be 61 nodes whereas for $k=2$ it increases to 83 nodes. Thus the threshold is proportional to network size.
Figure 4.15 Network Lifetime Comparisons for $k=1, k=2$

4.4 SUMMARY

This proposed algorithm used coverage-preserving centralized node-scheduling scheme to reduce energy consumption and therefore increase system lifetime, by turning off some redundant nodes. Further to increase the network lifetime and to extract the residual energy from the remaining nodes by changing the monitoring area levels. The improvements obtained from the proposed scheme are network lifetime is increased by 30000- 50000 rounds, the energy consumption is reduced by 41.2% to 58.6 % compared to without scheduling scheme. But it also infers that only some 30% of deployed sensors are utilized in subset formation and remaining nodes are not used.

Hence to utilize those nodes here it is opted to go for the compromise in area .Thus from the results, it is concluded that by
compromising in area more number of nodes are utilized, residual energy is reduced and network lifetime is improved. From the results we conclude that the localized algorithm provides better performance than centralized when the network size is smaller. That is when the network has 25 nodes it provides 21,000 rounds greater life, but when the network size increases the exchange of hello packets consumes a considerable amount of energy in LSKA. Hence CSKA provides better result for larger networks. With the results of CSKA and LSKA are plotted in a same graph, a crossing point is found. This is the threshold to switch from LSKA to CSKA. This threshold value is used to switch from one algorithm to another algorithm, depends on the required coverage level \( k \). Thus the network lifetime is improved compared to all other existing methods and this proposed method is suitable for both regularly and randomly deployed nodes.