Multiple-Sink Placement Strategies in Wireless Sensor Networks

Debasree Das1, Zeenat Rehena2, Sarbani Roy3, Nandini Mukherjee3

1WBSEDCL, Salt Lake, Kolkata, India
2Dept. of Computer Science and Engineering
Aliah University, Salt Lake, Kolkata, India
3Dept. of Computer Science and Engineering
Jadavpur University, Jadavpur, Kolkata, India
debasree.cse@gmail.com, zeenatrehena@yahoo.co.in, sarbani.roy@cse.jdvu.ac.in, nmukherjee@cse.jdvu.ac.in

Abstract—Multiple sinks are deployed in large-scale wireless sensor networks (WSNs) to minimize transmission delay and energy consumption and also to extend the network life time. Since the data collected by sensor nodes are forwarded to the sink, therefore proper placement of sinks has a great impact on the performance of the WSNs. This paper introduces two sink placement strategies and discusses their advantages and disadvantages in comparison with an existing strategy. The two strategies are compared with the Geographic Sink Placement (GSP) [3] strategy which is used as a benchmark. Both GSP and proposed two strategies are implemented and evaluated in a simulation environment. Performances of these strategies are analyzed and analysis results are presented in this paper. It has been observed that the proposed strategies exhibit better performances with respect to energy usage and lifetime in comparison with GSP.

Keywords— Wireless Sensor Network; Multiple Sink; Sink Placement; Grid Cell

I. INTRODUCTION

In large-scale single sink wireless sensor networks (WSNs), nodes closer to the sink become bottleneck due to heavy traffic load for packet transmission. This happens because these nodes not only collect data within their sensing range but also forward data for nodes which are far away from the sink. This leads to unbalanced power consumption among the sensor nodes and connectivity within the network may be lost. Furthermore, time being the major constraint for applications like disaster management, it is also desired that most sensor nodes be close to the sink. Contemporary research works [1,2] demonstrate that performance, such as data transmission time from source to sink is improved in multiple sink networks in comparison with single sink networks. A suitable multiple sink placement strategy can strongly decrease both aforementioned issues by shortening multi-hop distances between the sensor nodes and sinks. Therefore, in this paper we explore multiple sink placement strategies for large-scale WSNs in order to reduce time for transmitting data from source to sink, to provide better energy-efficiency and as a result to prolong network lifetime.

In this paper two sink placement strategies are proposed and their performances are analyzed and compared with Geographic Sink Placement (GSP) strategy [3]. “Placing the sink in a region where number of nodes is maximum” - this is the main design idea of the proposed sink placement strategies. The network is assumed to be partitioned and therefore the objective of each strategy is to place a sink in each partition. Unlike the previous research works in this area, the strategies proposed in this paper are based on a grid structure that divides the network into square-shaped grid cells. Use of the grid structure helps to find the proper location for the sinks to place. Other issues which are handled by these two algorithms are avoiding energy hole [4] near the sink and prolonging the network lifetime.

The rest of the paper is organized as follows. Section II presents the related work. The sink placement issue and proposed two strategies are explained elaborately in Section III. Section IV presents the simulation environment and analyses the performance of the proposed two sink placement strategies in comparison with GSP sink placement strategy. For the purpose of simulation, we use flooding routing protocol. Finally, the paper concludes in Section V.

II. RELATED WORK

In general the sink placement problem is NP-complete [6], and finding the best position of sink is very hard. The authors in [3] introduce different sink placement strategies and discuss their advantages and disadvantages. Among the other proposed strategies, Geographic Sink Placement (GSP) strategy places the sinks at center of gravity of a sector of a circle. In case of Intelligent Sink Placement (ISP), candidate locations are determined by sampling all possible regions and depending on the number of sinks, all combinations of these candidate locations are enumerated to find an optimal sink placement. This strategy (ISP) is found to be an optimal one. However, ISP is computationally expensive and it is assumed that the location information of the sensor nodes be provided by some localization system. Another algorithm, called Genetic Algorithm-based sink placement (GASP) is also introduced. GASP provides a good heuristic based on Genetic Algorithm for optimal sink placement. Vincze. Z. et. al. [7] gives a mathematical model that determines the locations of the sinks by minimizing the average distance of sensors from the nearest sink. A self-organized sink placement strategy (SOSP) for multiple sinks in a large scale network is proposed in [12, 13]. It is claimed that SOSP has lower communication overhead.
Some research works in this field focus on the use of integer linear programming and iterative clustering techniques. In [9] sink placement and data route problems have been formulated based on linear programming and the optimal locations of multiple sinks and data flow in the WSN are proposed. Another solution is presented in [10] based on iterative clustering algorithms, such as k-mean. The idea here is to define some initial clusters, place the sinks in the center of those clusters, and then reshape them, so as to allow sensors to choose the nearest sink. This procedure is repeated until the clusters are not reshaped anymore. In both papers [9, 10], the basic objective is to improve the network lifetime.

In contrary to all the above algorithms, we propose to start with a partitioned network. A network is partitioned using one of the techniques described in [14]. The objective is to place a sink in every partition. Unlike the previous works, it is assumed that each partition is divided into square-shaped grid cells and it is also proposed that the size of the grid cells be based on the communication range of the sensor nodes.

The proposed sink placement strategies are mainly intended for time-critical WSN applications. Several sink placement strategies [3] have been presented, discussed and evaluated to achieve the goal of maximizing the lifetime of a WSN. Among them RSP places sinks randomly in the network and the results are arbitrary and uncontrolled, even in the same network [3]. Therefore, it is not suitable for time-critical purposes, due to the random placement of sinks.

An ISP (under certain restrictions) strategy provides optimal sink placement but it is computationally very expensive.

Also GASP gives a good solution as other heuristic methods, although it cannot guarantee to reach an optimal solution. Whereas GSP strategy is proposed as a benchmark sink placement strategy in WSNs and it is clearly computationally efficient. This is an obvious advantage over the ISP and GASP strategy. For these reasons we chose GSP to compare our strategies.

### III. Sink Placement Strategies

Let us consider that \( n \) numbers of sensors are deployed in a square area and there is \( p \) number of sink nodes to be placed in the WSN. Before placing the sinks into the WSN, we partitioned the WSN into some sub-partitions and partition of a WSN can be defined as follows:

Let \( S \) is the set of sensor nodes, \( E \) is the set of communication links between nodes \( s_i \) and \( s_j \), \( s_i, s_j \in S \). Therefore, the partition of the network into \( p \) sub-partitions \( Sk_i \), where \( \{Sk_i : S : 1 \leq i \leq p\} \), such that

\[
\bigcup_{i=1}^{p} Sk_i = S \quad \text{and} \quad Sk_i \cap Sk_j = \emptyset \quad \text{for} \quad i \neq j.
\]

\( Sk_i \) is the disjoint set of sensor nodes.

The main objective is to place a sink in each partition for achieving longer lifetime, energy efficiency, as well as faster data delivery to each sink. So a sink placement strategy is needed for achieving the above-mentioned goals for each sub-network, and all these sub-networks collectively will achieve the goals for the large scale WSN. Now if the sink is placed in a location where number of neighboring nodes is very less then these limited number of nodes will be repeatedly used for relaying packets to the sink. As a result these nodes will run out of battery power very soon and thus the lifetime of the WSN will become shorter. So, for achieving longer lifetime, sinks should be placed in appropriate locations where number of neighboring nodes is high. Figure 1 depicts a scenario where sink has less number of neighbors for communication.

To find a region having maximum number of nodes, we propose to divide the network into equal sized square grid and the grid cell which contains maximum number of nodes is the probable location for placing the sink. Figure 2 shows the grid structure of WSN and two possible locations where the sink can be placed.

![Figure 1. Sink Placement in Low Density Area](image1)

![Figure 2. Grid Structure of WSN](image2)

#### A. Grid Size Estimation

For our first algorithm, we are interested to find the area or region having densely deployed sensor nodes. Therefore, determining the size of the grid cell is very important. Let communication range of each sensor node be \( R \). If the sink is placed at the centroid of the nodes in a grid cell, then the position of the sink can be anywhere within the grid cell (including the vertices), and one (or more) vertex of the grid cell will be the farthest point from the sink. So if a node is at the farthest point then it can only communicate with the sink if the distance is less than or equal to \( R \). Now if the sink is placed at one vertex of the grid then the maximum distance of a node from the sink is the distance of the diagonally opposite vertex and the distance should be at most \( R \). Thus, diagonal of a grid
cell should be R and each side should be $R/\sqrt{2}$. This is shown in Figure 3.

So in the following algorithm, we propose a strategy to relocate the sink in the dense area.

**a) Finding Candidate Locations:** For each partition, the algorithm first identifies the grid cells that contain nodes of that partition. Centroids of these grid cells are selected as initial sink position. Then for each grid cell the initial location is refined as follows.

To find the dense area, number of 1-hop neighbor nodes of the initial sink position is counted. Next the centroid of these 1-hop sensor nodes is calculated. This centroid position is called NEXT LOC and is considered to be the probable next location for the sink. Next, number of 1-hop neighbor nodes of NEXT LOC is counted. If number of neighbor nodes of NEXT LOC is higher than that of the sink, then the sink is relocated to the current NEXT LOC position and the centroid of the neighbor nodes of the current NEXT LOC position is calculated and termed as the new NEXT LOC. This process continues until there is no increase in neighbor nodes. If number of neighbor nodes of NEXT LOC is same or less than the number of neighbor nodes of the sink, then the sink remains at the same position, otherwise the sink is placed at the position where neighbor nodes of NEXT LOC is found to be higher. Figure 4 depicts the steps of finding the candidate location of a grid cell in a partition. First the sink is positioned at 1 (centroid of a grid cell) in Figure 4(a) and NEXT LOC is found at 2 which becomes the new sink position. In the next iteration, NEXT LOC of position is found at 3, but the sink is placed at the same position, otherwise the sink is placed at the position where neighbor nodes of NEXT LOC is found.

**b) Deciding Final Location:** Once the candidate locations for each grid cell are determined, final location of the sink should be selected from amongst these candidate locations. It may be noted that the some candidate locations may be on the boundaries of one or more grid cells or may actually be shifted to another grid cell while finding the dense region. The final location of the sink in each partition is decided as follows.

Let $C_i$ to $C_p$ be the candidate locations in a partition.

Let $F_j$ be the farthest node from $i^{th}$ candidate location having $H_j$ hop distance where $i \in 1$ to $p$.

Now we are interested to find such candidate location $C_i$ which has minimum hop count $H_j$ from the farthest node ($F_j$) from $C_i$.

Thus the final location $S$ is given by

$$S = \{C_j \mid \text{dis tan} \text{cel}(C_j, F_j) = \min(H_j) \forall C_j, 1 \leq j \leq p\}$$

The reason for choosing the candidate location with minimum hop distance from farthest node is that this candidate location gives the minimum distance to all other nodes in that partition. This is because while redefining the sink’s position it moves sink towards dense region of the partition.
3) **Centroid of the Nodes in a Partition (CNP)**: The above-mentioned algorithm uses a grid cell structure of the network. However, in case of CNP algorithm no grid cell structure is considered. Here the sink is initially placed at the centroid of all sensor nodes in a partition. Next the number of 1-hop neighbors of every sink is calculated. Then a new location of the sink is found by calculating the centroid of 1-hop neighbors for each partition. Again the number of 1-hop neighbors of the new location is calculated. If the number of 1-hop neighbors of the new location is greater than the number of 1-hop neighbors of the old location, then new location becomes the sink location and again the previous steps are repeated until we finalize the sink location. Sink location is finalized when the number of 1-hop neighbors of the new location is found to be less than the number of 1-hop neighbors of the old location and the old location is taken as the final location of the sink.

**IV. SIMULATION ENVIRONMENT**

Performances of these two algorithms have been evaluated in Matlab environment and compared with GSP sink placement strategy. GSP is also implemented and evaluated in Matlab. One hundred (100) sensor nodes are randomly deployed in a 200m x 200m square area. All nodes have the same capabilities. The communication range is 45m. Each algorithm is run in 10 different topologies. Initially for CLMH and CNP, the WSN is partitioned into 4 sub-partitions by using Modified Recursive Spectral Bi-Section algorithm [15]. As discussed earlier, the WSN is considered to be divided into several square shaped grid cells. In case of CLMH algorithm, the grid cell size is $R/\sqrt{2}$. R is the communication range of a sensor node. The CNP algorithm does not require any grid cell.

Sink placement using CLMH algorithm and CNP algorithm are shown in Figure 5 and Figure 6 respectively for random deployment of sensor nodes. Sink placement using GSP with random node deployment is shown in Figure 7.

Figure 4. Finding candidate location

(a) (b) (c)

Figure 5. Sink Placement using CLMH Algorithm

Figure 6. Sink Placement using CNP Algorithm

Figure 7. Sink Placement using GSP
A. Performance Metrics

The following metrics have been measured in case of each simulation of the strategies.

1) Execution Time: Execution time of each sink placement algorithm
2) Avg. Energy Consumption: Energy consumption for a single event to reach the sink node
3) First Node Die: Number of rounds before the first node die.
4) Last Node Die: Number of rounds until the last node die.

B. Simulation Results

Since the GSP strategy uses uniform node deployment, simulations of these three strategies are carried out with uniform node deployment, as well as with random node deployment. In GSP, the authors considered a circular network area. However, for CLMH and CNP we consider square shaped network area.

1) Uniform Node Deployment: Uniform node deployment is applied for evaluating all three strategies. Results of the above mentioned performance metrics are measured. The execution times for GSP and CNP are almost equal, while CLMH requires larger execution time than the other two strategies. The result is shown in Figure 8. Average energy consumption in the three cases is depicted in Figure 9. It is observed that GSP and CNP consume same amount of energy. As expected, CLMH consumes less energy among the three. Since in case of CLMH, the sink is placed at the location from where the distance of all nodes is minimum in each partition.

Further, among the three strategies CLMH covers maximum rounds until the first node dies as shown in Figure 10. Similarly, Figure 11 shows rounds until last node dies. It is clear from the figure that CLMH results longer lifetime.

2) Random Node Deployment: The four performance metrics mentioned in Section IV A with random sensor node deployment are shown in Figure 12, Figure 13, Figure 14 and Figure 15 respectively. With random deployment, CNP has lower execution time in comparison with GSP and CLMH (Figure 12). CLMH has higher execution time in case of uniform as well as random deployment.

CNP and CLMH consume less energy compared to GSP (Figure 13). Further, these two strategies cover higher number of rounds until first node die than GSP. Among the two, number of rounds in case of CNP is higher than CLMH as depicted in Figure 14. Figure 15 shows rounds until last node die for these three strategies. Here CNP covers maximum rounds and hence achieves the longer lifetime.
C. Result Discussion

Observing the above results, it is noticed that CLMH strategy has higher execution time than other two strategies for both random and uniform node deployments pattern.

According to the algorithm of CLMH, candidate locations are calculated in each grid cell in each partition. This huge calculation increases the execution time and the execution time increases as the number of grid cells increases in large scale network. On the other hand, for random and uniform deployments both GSP and CNP require less expensive calculation to find the sink location than CLMH. GSP always finds the centroid for sink location while in CNP sink location may or may not be the centroid position.

For GSP strategy, the average energy consumption is high in both deployments. This is because GSP would not provide the better location for sink to place and the communication distance to all nodes in the sector is not minimum. Again for GSP in random deployment, average energy consumption is higher than uniform deployment (0.014 J in random and 0.013 J in uniform). But CLMH places the sink in a location from where every node in each partition has smaller distance in comparison with other candidate locations (in terms of hop).

Therefore, CLMH consumes less amount of energy for data transmission. Also CNP needs less amount of energy in random deployment than uniform deployment.

Here, lifetime of the network is measured by using the metric (3) and (4). As expected, since both CNP and CLMH need lesser amount of energy than GSP for data transmission, CNP and CLMH give almost same lifetime and improve over GSP.

It is also observed that GSP and CNP have the similar performances only for uniform node deployment. GSP does not work well in random node deployment.

Thus, we can conclude that CLMH and CNP give better performances over GSP.

V. CONCLUSION

The paper focuses on multiple-sink placement problem in a wireless sensor network. The network is assumed to be partitioned. We propose two sink placement strategies in the partitioned network. These strategies are compared with existing sink placement strategy, GSP. Performance of the network after application of each strategy is observed.

From the results it can be concluded that the CNP strategy shows better performance in all respect when compared with other algorithms for random node deployment. It has low execution time, low energy consumption and also lifetime (in terms of first node and last node die) is longer for this algorithm. The CLMH strategy also demonstrates similar performance like CNP. However, execution time of CLMH is higher in both uniform and random deployment. For uniform node deployment CLMH shows low energy consumption and longer lifetime. Therefore, considering the network performance, CLMH and CNP strategies appear to be good solutions to the sink placement problem and outperform a
benchmark algorithm called GSP [3], which is used to solve the sink placement problem.

REFERENCES


