Multiple Sink Placement in Partitioned Wireless Sensor Networks

ZEENAT REHENA [1], DEBASREE DAS[2], SARBANI ROY[3] AND NANDINI MUKHERJEE[3]
[1]Department of Computer Science and Engineering, Aliah University, Kolkata-700156, India,
Email: zeenatrehena@yahoo.co.in
[2]WBSEDCL, West Bengal, India, Email: debasree.cse@gmail.com
[3]Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India,
Email: {sarbani.roy, nmukherjee}@cse.jdvu.ac.in

Multiple sink deployment in large-scale wireless sensor networks (WSNs) can reduce transmission distance between source nodes to sink. This reduces energy consumption for data transmission and extends the network life time. However, proper placement of sinks has a great impact on the performance of the WSNs. This paper introduces six sink placement strategies and discusses their advantages and disadvantages. The network is assumed to be partitioned into several sub-networks and therefore the objective of each strategy is to place a sink in each partition. These strategies are based on a grid structure that divides the network into square-shaped grid cells. Results of application of the six strategies in a network with random deployment of sensor nodes are analyzed. Performances of the network after application of these strategies are also analyzed and analysis results are presented in this paper.

Keywords: Multiple Sink, Sink Placement, Grid Cell, Lifetime, Wireless Sensor Networks

1. INTRODUCTION

In Wireless Sensor Networks (WSNs) data is transmitted from source nodes to sink node in a multi-hop manner. However, in large-scale networks, a single sink is placed at a position which may be far away from many nodes and therefore, data transmission through multiple intermediate nodes may be expensive in terms of energy. Propagation delay from the distant nodes to the sink node may also be large.

In addition to the above two major problems, some other problems are encountered in a single sink WSN that significantly deteriorate its performance. Uneven energy depletion is one of them. Sensor nodes near the sink deplete their energy faster than those are far apart from the sink due to their heavy overhead for relaying messages. Nodes closer to the sink not only collect data within their sensing range, but also forward data from the nodes which are away from the sink and thus become bottleneck due to heavy traffic load for packet transmission. Non-uniform energy consumption degrades network performance and shortens network lifetime. Moreover, if all sensor nodes around the sink run out of energy, then sink becomes isolated from the network, as a result the entire network fails.

In case of some applications of WSN, it may be required to randomly deploy several thousands of sensor nodes within a given region and responses from these nodes must be sent to the base station quickly and reliably while maintaining optimum energy consumption within the network. Such applications include forest fire detection, disaster relief, emergency rescue operation etc.

Disaster management applications usually require sensor nodes to be deployed over a large monitoring region which may be spread over several kilometers. Therefore, scalability is a major issue in such a large-scale network. Deployment of a single sink in such a large area cannot make the network scalable.

One solution to the above problems is to use multiple sink nodes which can be strategically placed within the network. Contemporary research works [Chen et al. 2005; Flathagen et al. 2011] demonstrate that performance, such as data transmission time from source to sink is improved in multiple sink networks in comparison with single sink networks. Also uses of multiple sinks ensure fast and reliable delivery of information and optimum energy consumption. The distance between source nodes to sink nodes can be reduced by deploying multiple sinks instead of a single one. In [Li et al. 2008], the authors observe that the average number of hops a data packet requires to travel decreases with the increase in number of sinks. Therefore, the average energy consumption for data communication also decreases. Moreover, in order to obtain a scalable network, the sensor nodes may be divided into clusters or sub-partitions. The sink nodes can be deployed in such a manner that an individual sink node becomes responsible for each cluster or sub-partition. It is also demonstrated in [Oyman and Ersoy 2004] and [Flathagen et al. 2011] that an increase in the number of sinks can affect the network lifetime of WSN.

Thus, there are significant benefits of using multiple sink networks. These benefits are summarized below:

(1) Multiple sinks in a network alleviates unbalanced energy consumption among sensor nodes since the data transmission load is shared among all the sinks.
(2) Multiple sink networks can remarkably reduce the mean distance as well as the hop distance between sensor nodes and sinks. Thus basically it provides energy saving and lower latency.
(3) Finally, multiple sink deployment avoids the single sink bottleneck problem. In case of disconnection of one particular sink, sensor nodes can still transmit their data towards other sinks in the network and hence the network functions continuously.

However, in order to avail the above benefits, proper placement of the sinks within the network is an essential requirement. Therefore, in this paper we explore sink placement strategies in large-scale WSNs in order to reduce distance for transmitting data from source to sink, to provide better energy-efficiency and as a result to prolong network lifetime. Here we propose six sink placement strategies and compare their relative performances. The basic design principle adopted for the six strategies is - "Placing the sink in a region where number of nodes is maximum". A grid cell structure of the network is used for all the strategies presented here. The algorithm proposed in this paper finds locations for placing the sinks that maximizes the 1-hop neighbor nodes of sink and minimizes the average Euclidean distance of all nodes from the sink.

The rest of the paper is organized as follows. Some related works are presented in Section 2. The sink placement problem is described in Section 3. The proposed six sink placement strategies are explained elaborately in Section 4. Section 5 presents the results of application of these strategies in a simulation environment. For performance analysis of the multiple-sink network with various sink placement strategies, we have used flooding routing protocol. The results of this analysis are presented in Section 6. These strategies are also compared with the CP strategy (sink placement at each corner of a square region) described in our earlier work [Rehena et al. 2011]. Finally, the paper concludes in Section 7 with a direction of future work.

2. RELATED WORK

In general the sink placement problem is NP-complete [Bogdanov et al. 2004], and finding the best position of sink is very hard. Vincze. Z. et al. in [Vincze et al. 2007] give a mathematical model that determines the locations of the sinks by minimizing the average distance of sensors from the nearest sink. The idea used here is similar to the k-mean clustering algorithm [Forgy 1965]. They present two algorithms in this paper. First an iterative algorithm called global is presented which finds the sink locations based on the global knowledge of the network. Given an initial sink setup, the sinks use their global knowledge to decide which sensors are closest to them and divide the network into clusters. Next the centroids of these clusters are determined and new locations of sinks are found using the mathematical model and clusters are recalculated. The process repeats until there is any change in the clusters. The paper also proposes another iterative algorithm, called 1hop that carries out the sink deployment based only on the location information of the neighboring nodes and locations of the distant nodes are approximated. For getting the location information of the sensor nodes the algorithm relies on message transmission from every sensor node to the sink. The same mathematical models is used here to calculate new sink locations based on the location information and the process iterates as long as the sinks stop moving to a new position.

In [Poe and Schmitt 2009b], the authors introduce different sink placement strategies and discuss their advantages and disadvantages. All these sink placement strategies are mainly intended for time-critical WSN applications, except only one known as Random Sink Placement (RSP). RSP is not suitable for time-critical purposes due to the random placement of sinks. RSP can be considered as a lower bound. Among the other proposed strategies, Geographic Sink Placement (GSP) strategy places the sinks at the 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.

Some research works in this field focus on the use of integer linear programming and iterative clustering techniques. In [Kim et al. 2005] 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 [Oyman and Ersoy 2004] 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 [Kim et al. 2005; Oyman and Ersoy 2004], the basic objective is to improve the network lifetime.

Mixed-integer linear programming (MILP) formulation is done in [Guney et al. 2010] and an integrated model of sink location and routing problem is described. The sensor field is divided into many grids and the nodes are arranged as the grid points. The solution minimizes the total energy consumption and balances the data flow in the network.

The sink placement problem is defined as base station positioning problem of sensor networks in [Bogdanov et al. 2004]. The authors considered the effect of base station positioning on energy consumption of the network. They considered the network as a regular grid structure. They used a greedy algorithm and a local search algorithm for finding base station position. The greedy algorithm selects position of each base station to improve performance as much as possible while keeping the location of the previous base stations fixed. The local search algorithm starts with a random configuration of base stations and tries to improve performance as much as possible by moving any base station. When no improvement is possible it reaches at local maximum and the algorithm again restarts with a new random configuration. After repeating the process with a number of random configurations, the highest local maximum is selected as final base station positions.

In [Poe and Schmitt 2009b; 2009a] the authors propose a self-organized sink placement strategy (SOSP) for multiple sinks in a large scale network which has lower communication overhead. A circular field shaped network is considered and the field is divided into equal sized sectors based on the number of sinks. Then the center of gravity of each sector of that circle is calculated and initially a sink is placed in each sector. In the next step, candidate locations are created by determining distances of 1-hop neighbors from each sink and using trilateration with TOA. Sinks are moved to each candidate location and the maximum worst-case delay is calculated. Finally the best candidate location is chosen where the worst-case delay is minimum.

In [Chen and Li 2013], the authors have explored the sink placement problem and propose strategies to find the optimal position of the sink. They investigate the energy-oriented and lifetime-oriented sink placement strategies for the single-hop and multiple-hop WSNs, respectively. The energy-oriented strategy considers minimizing the total energy consumption in the network, while the lifetime-oriented strategy focuses on the lifetime of the nodes which consume energy fastest. To simulate single-hop and multiple-hop networks they also adopt a routing-cost based ant routing algorithm. The authors also demonstrate that lifetime-oriented strategy performs better than energy-oriented strategy in terms of lifetime of the network.

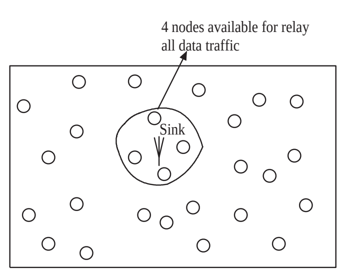
In [Zadeh et al. 2011], authors use dynamic sink placement strategies. They propose a new dynamic approach to find the optimal position for the sink with the goal of reducing the total energy consumption. The algorithm is based on exhaustive search to find the initial optimal sink position. It also relocates the sink in an appropriate location when needed. In contrary to all the above algorithms, the main contribution of this research work is to propose sink placement algorithms for a partitioned network. A network is partitioned or clustered using one of the techniques described in [Rehena et al. 2012]. The objective is to place a sink in every partition with an objective of increasing the lifetime of the network. In this work, it is assumed that each partition in the network 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. On the basis of the sensor node density in grid cells, a concept of MaxGrid is introduced and used by the proposed techniques to find a new location for sink placement. Besides network lifetime, maximum coverage of 1-hop neighbor nodes, and minimum hop distance to farthest neighbor nodes are also considered in the proposed approaches. Since, energy and computation resources are very limited in WSN, the basic idea behind these strategies is to keep the computation as simple as possible.

3. SINK PLACEMENT PROBLEM

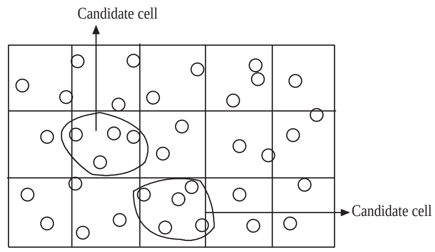
The main objective is to place a sink in each partition for achieving longer lifetime, energy efficiency as well as faster data delivery to each sink. So a sink placement strategy is needed for achieving the above-mentioned goals for each sub-network, and all these sub-networks collectively will achieve 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 a probable location for placing the sink. Figure 2 shows the grid structure of WSN and this network has two possible locations (with maximum number of nodes) where a sink can be placed.

Therefore, we introduce the concept of MaxGrid and it is defined as a grid cell which contains maximum number of sensor nodes of a partition.



. Figure. 1: Sink Placement in Low Density Area

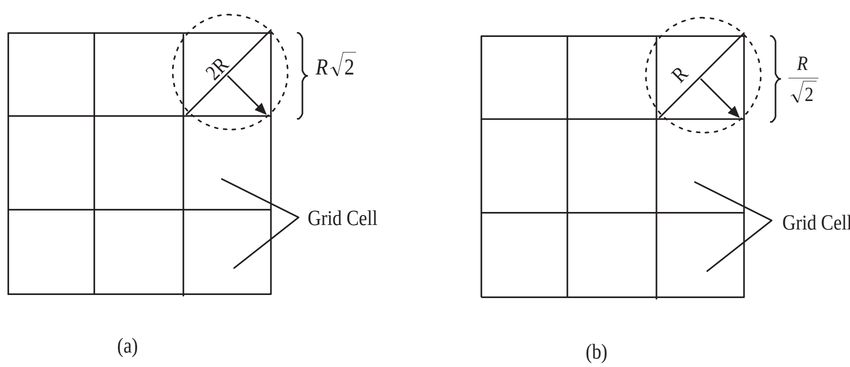


. Figure. 2: Grid Structure of WSN

3.1 Grid Size Estimation

For some of our algorithms 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. Here two sizes for the grid cell is considered. Let communication range of each sensor node be R. If sink is placed at the center of the grid cell then each vertex of the grid cell is the farthest point from the center. The nodes which are on the farthest point should be able to communicate with centrally placed sink. This is possible only if their distances from sink is less than or equal to R. So the diagonal of the square-shaped cell should be at most $2R$. Therefore, the side of each grid cell is $R\sqrt{2}$. The Figure 3(a) depicts the grid cells.

However, if the sink is placed at the centroid of the nodes in a grid cell, then the position can be anywhere within the grid cell (including the vertices) and any 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. So if the sink is placed at one vertex of the grid then the maximum distance of a node from the sink is the diagonally opposite vertex and the distance should be at most R. So diagonal of a grid cell should be R and each side should be $R/\sqrt{2}$. This is shown in Figure 3(b).



. Figure. 3: Different Grid Cell structure

4. PROPOSED STRATEGIES

It is assumed that a set of sinks are to be deployed in a large scale WSN and according to the number of available sinks the entire network is partitioned. Our goal is to place a sink in each partition so that all sensor nodes in a partition can communicate to the corresponding sink. Here it is also assumed that the global knowledge about locations of the sensor nodes is available. Based on the above assumptions, six algorithms are presented in this paper. These are: CMG, CNMG, CNMG-1, CLMN, CLMH and CNP. The six algorithms are discussed below.

4.1 Center of the MaxGrid (CMG) Algorithm

This algorithm initially converts the entire WSN into square grid structure. The diagonal of each grid cell is $2R$ where R is the communication range of a sensor node. Now for each partition, the algorithm finds the grid cell which contains maximum number of nodes i.e. MaxGrid. The center of that grid cell in each partition is chosen as the final location of the sink in that partition.

This algorithm is straightforward and simple. Sometimes it may happen that a MaxGrid cell is present at partition boundary and the sink is placed at the center of that MaxGrid cell. Therefore it may increase the hop distance from sensor nodes to sink. In such scenario this strategy may not give good energy efficiency and may result in to decrease in lifetime or slow data delivery.

While finding time complexity, we make an assumption. We assume that the partition sizes are almost same and therefore, number of grid cells in each partition is almost equal. Hence, the time complexity of the algorithm is linear time complexity of order of $O(lm)$, where l is the number of partitions in the network, and m is the average number of grid cells in each partition.

4.2 Centroid of Nodes of MaxGrid (CNMG) Algorithm

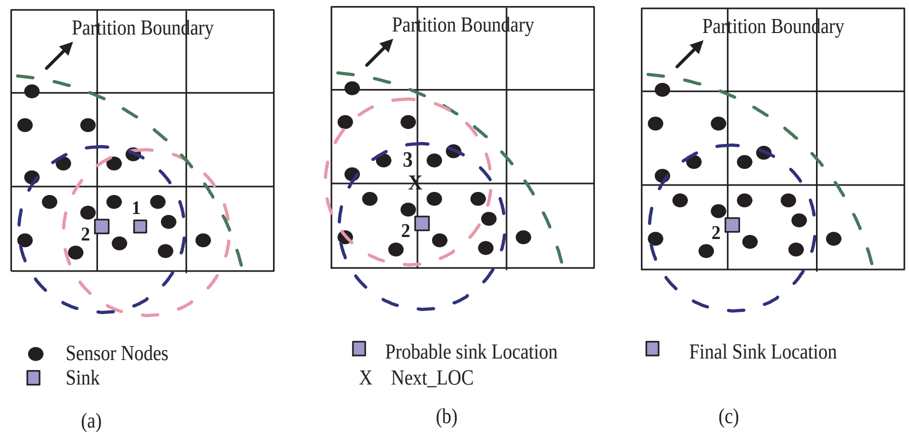
As in the case of CMG algorithm, this algorithm also divides the WSN into square-shaped grid cells. But unlike CMG algorithm, here the length of the diagonal is taken as R. Now for each partition, the algorithm finds the MaxGrid cell. However, instead of choosing center of the MaxGrid as the sink location, the centroid of the all sensor nodes within the grid cell is computed. The centroid then becomes the location of the sink in that partition. Thus, in case of CMG, sometimes the sink may be placed on the boundary of the partition. But by choosing the centroid of the sensor nodes in MaxGrid, this algorithm can solve the problem arising in such situation.

This algorithm is also straightforward and simple to implement. But in certain cases, finding a good sink location becomes difficult. Sometimes it may happen that the area close to the boundaries of two or more grid cells is covered with maximum number of sensor nodes. Therefore, in order to cover maximum number of sensor nodes, the sink must actually be positioned in the dense region which is spread over two or more grid cells. Problem is to find such a location the grid cells must be combined. In the next algorithm we propose a strategy to relocate the sink in the dense area. The algorithm uses NEXT_LOC which is defined as the Centroid of the 1-hop neighbor nodes of a sink. This algorithm also has the similar complexity like the previous one i.e. $O(lm)$.

4.3 CNMG with 1-hop Neighbor (CNMG-1) Algorithm

This is an iterative process and the sink is placed in a new location every time by finding its 1-hop neighbours starting with the sink location found by CNMG.

During every iteration, from the current position of the sink, number of 1-hop neighbor nodes 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 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 the algorithm. First the sink is positioned at 1 (centroid of the MaxGrid) 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 remains in its old position (2) because there is no increase in 1-hop neighbor node count (Figure 4(b)). Thus, the final position of the sink is 2 which is shown in Figure 4(c).



. Figure. 4: Steps of CNMG-1 Algorithm

We next define a Candidate Location as the final location is calculated by using CNMG-1 algorithm in each grid cell of each partition.
The worst case time complexity of this algorithm is $O(n_{P_i})$,
where P is the set of partitions, i.e. $P = \{P_1, P_2, \cdots, P_l\}$ and n_{P_i} is the number of nodes in partition P_i. Here $P_i \in P$, $1 \leq i \leq l$.

4.4 Candidate Location with Maximum Neighbor (CLMN) Algorithm

In this algorithm the definition of Candidate Location is used. The sink is placed by selecting Candidate Location which has maximum number of 1-hop neighbors. For each partition the algorithm first identifies all the grid cells which contain any node of that partition (the entire grid cell may not fall within the partition). Instead of choosing MaxGrid cell, here each grid cell is chosen and the CNMG-1 algorithm is used to find a Candidate Location in each grid cell. In the next step, for each Candidate Location, number of 1-hop neighbors is counted. The Candidate Location with maximum 1-hop neighbors is selected as the final location of the sink and the sink is placed at that location.

4.5 Candidate Location with Minimizing Hop (CLMH) Algorithm

This algorithm is same as CLMN algorithm. The only difference is that the sink location is finalized by selecting candidate position which has minimum hop distance from the farthest node.

The algorithm is described as follows. For each partition the algorithm first identifies the grid cells that contain any node of that partition (as in the previous case). Instead of finding MaxGrid cell, here a candidate location is found for each grid cell using CNMG-1 algorithm. In the next step, for each candidate location the algorithm finds hop distance of the farthest node within the partition. The candidate location which has minimum hop distance to farthest node is selected as final location.
CLMN and CLMH both algorithms also have worst case time complexity of $O(n_{P_i})$.

4.6 Centroid of the Nodes in a Partition (CNP) Algorithm

All the above mentioned algorithms use the 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 for new sink 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. That is 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. This algorithm is similar to CNMG-1 algorithm described earlier, however, unlike the CMG, CNMG, CNMG-1, CLMN and CLMH algorithms, here we do not use grid cell structure.

This algorithm also has the same worst case time complexity of the previous algorithms like CNMG-1, CLMH, and CLMN i.e. $O(n_{P_i})$.

The above mentioned algorithms are different in many aspects. At this point, we compare the algorithms which are shown in Table I.

Table I. Analysis of characteristics for the Algorithms

Scheme	**Grid size (Diagonal)**	**Final Sink position**	**Type**	**Time Complexity**
CMG	$2R$	Center of the MaxGrid cell	Non-iterative	$O(lm)$
CNMG	R	Centroid of MaxGrid cell	Non-iterative	$O(lm)$
CNMG-1	R	Next_Loc is used to find the sink location	Iterative	$O(n_{P_i})$
CLMN	R	Candidate location which has maximum number of neighbor nodes is used as the sink location.	Iterative	$O(n_{P_i})$
CLMH	R	Candidate location which has minimum hop distance from farthest node is used as sink location.	Iterative	$O(n_{P_i})$
CNP	No grid cell	Sink location is found by calculating centroid of 1-hop neighbors. This is similar to CNMG-1.	Iterative	$O(n_{P_i})$

5. SIMULATION ENVIRONMENT

Performances of these algorithms have been evaluated in Matlab environment. One hundred (100) sensor nodes are randomly deployed in a $200m \times 200m$ square area. All nodes have the same capabilities. The communication range is $45m$. Each algorithm is run in 10 different topologies. Initially the WSN is partitioned into 4 sub-partitions by using Modified Recursive Spectral Bi-Section algorithm [Kabelikova 2006]. As discussed earlier, the WSN is considered to be divided into several square shaped grid cells. In case of CMG algorithm, the grid cell size is $2R$, whereas in case of other algorithms (except CNP) each grid cell is of size R. R is the communication range of a sensor node. The CNP algorithm does not use any grid cell.

5.1 Performance Metrics

The following metrics have been measured for each simulation.

—**Execution Time:** Execution time needed for each sink placement algorithm.

—**Farthest Hop Distance:** Hop count of the farthest node from the sink in each partition.

—**Average Hop Count:** The total hop length for all nodes to reach sink in each partition is divided by number of nodes in that partition.

—**Co-efficient of Variation:** The coefficient of variation (CV) is defined as the ratio of the standard deviation to the mean. It is used to measure the dispersion of hop count of every node from the sink.

5.2 Simulation Results

Simulation results are shown in Figures 5(a) to Figure 5(f). Figure 5(a) shows the final sink position using the CMG algorithm. It is clearly shown that sinks are placed in the center of the grid cell where maximum numbers of nodes are found. The final sink positions obtained by using the CNMG algorithm are shown in Figure 5(b). From the figure it is clear that sinks are placed at the centroid of nodes of the MaxGrid. Figure 5(c) depicts sink positions using CNMG-1. As expected, the algorithm places the sink at the border of two grid cells.

Sink placement using CLMN algorithm, CLMH algorithm and CNP algorithm are shown in Figure 5(d), Figure 5(e) and Figure 5(f)respectively.

5.3 Result Discussion

Figure 6 shows the execution times of each of the sink placement algorithm. CMG, CNMG and CNP require same execution time. In case of these three algorithms most time is spent for calculation of the center of the grid cell or centroid of the nodes. As depicted in Figure 6, CNMG-1, CLMN and CLMH require higher execution time. Among these three, execution time of CLMH is highest. This is because maximum overhead is incurred in CNMG-1 and CLMN algorithms for finding 1-hop neighbors and selecting final sink positions. On the other hand, in case of CLMH, overhead is incurred for finding the farthest hop length from all possible candidate positions.

The second metric is hop count from the farthest node to sink which is compared in Figure 7 for all the six algorithms. CLMH and CNP have same hop count from the farthest node to sink and give better result in comparison with other algorithms. CNMG and CNMG-1 have highest hop count. On the other hand, performances of CMG and CLMN lie between the two sets that are mentioned above.

Another performance metric is average hop count which is shown in Figure 8. CLMN, CLMH and CNP have same average hop count to sink. In these three strategies the location of the sink is found iteratively, and finally the sink is placed at the best location where the hop count is less. Among the remaining strategies, CNMG shows worst performance and needs maximum hop count to reach sink. In CNMG, sink is placed at the centroid of nodes of the MaxGrid and the MaxGrid may be far from any node in the partition. Thus the distance between the sink and any node in the partition may increase. CMG and CNMG-1 demonstrate moderate performance.

The final performance metric is known as co-efficient of variation of hop counts. This metric is obtained by taking the ratio of the standard deviation and the mean and of hop counts of every node from the sink in each partition. The co-efficient of variation for each partition is calculated and depicted in Figure 9. It is clear from the figure that CLMH and CNP algorithms have low ratio of the standard deviation to the mean than other algorithms. But the CNP algorithm has more or less same co-efficient of variation in each partition.

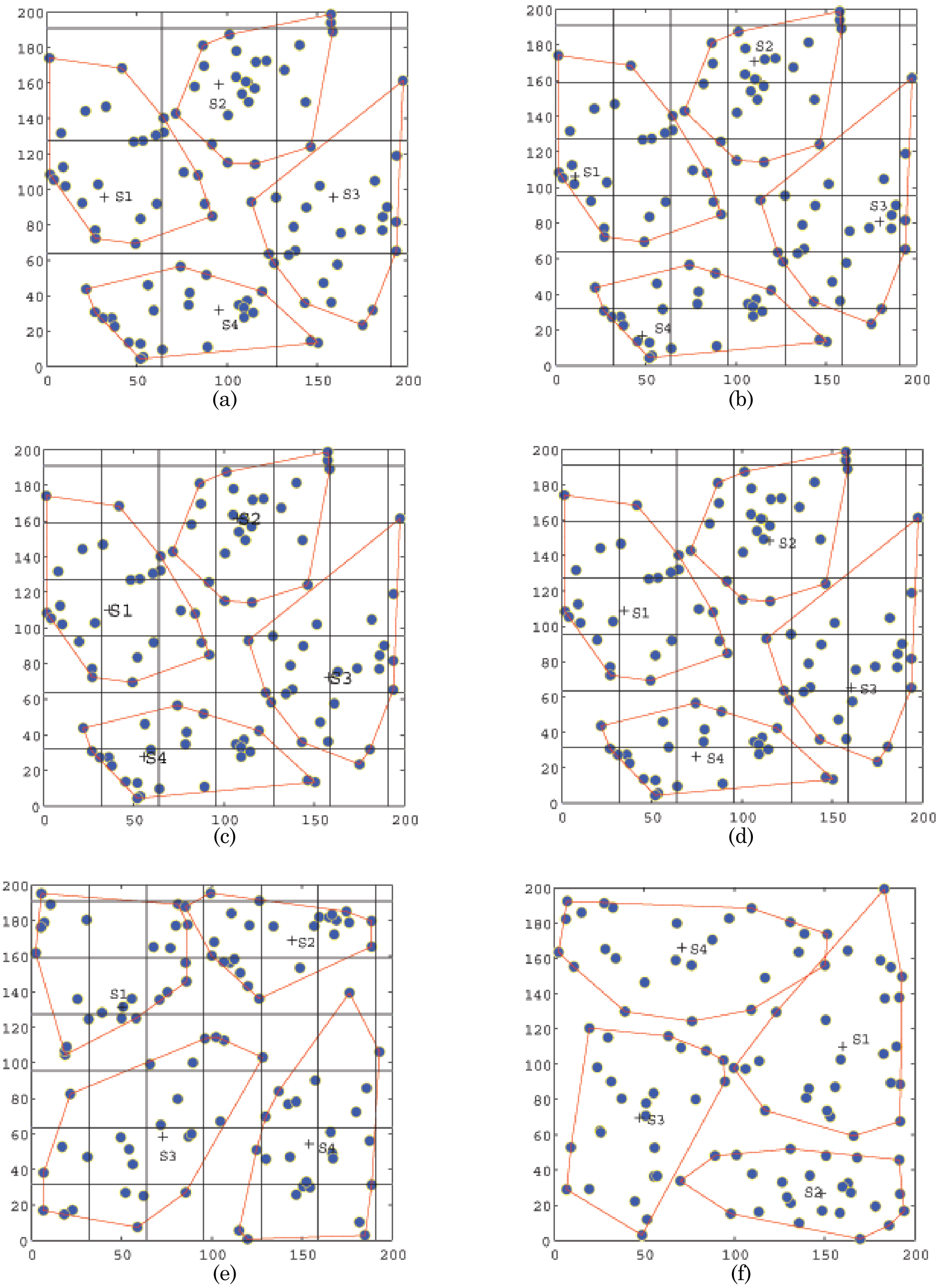
6. DATA TRANSMISSION

Performances of data transmission using flooding is discussed in this section.

6.1 Performance Analysis using Flooding

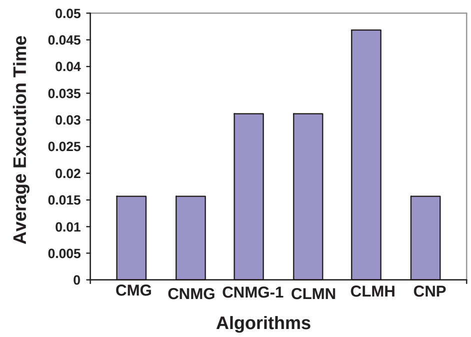
For performance analysis of our multiple sink networks with various sink placement strategies as described above, we have used flooding routing protocol. After placing all the sinks at their final positions, a restricted flooding protocol has been implemented where the data packets are routed only within the respective partitions in the partitioned network. In each case, the network lifetime and total energy consumption of the network have been measured. Thus, following descriptions of each metric are considered:

(i) Energy consumption for a single event to reach the sink node, (ii) Lifetime of the Network: described using two metrics, namely "time elapsed before the first node die" and "time until the last node of the network die".

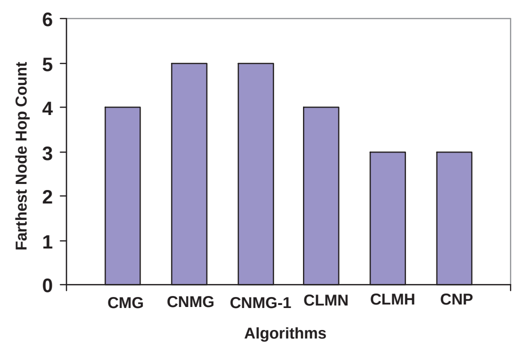


. Figure. 5: Simulation results show different sink positions using different algorithms: (a)CMG (b)CNMG (c) CNMG-1 (d)CLMN (e)CLMH (f) CNP.

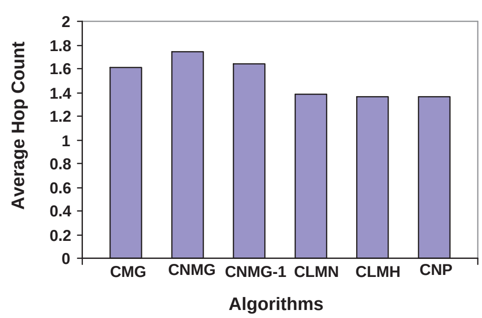
Figure 10 shows the average energy consumption for a single event to reach the sink node for each of the sink placement algorithms. In each case it is assumed that 10 events have occurred at different nodes (nodes are selected randomly). Energy consumption for the entire path (from source to sink) is measured and the average of energy consumption for the 10 events is shown in the figure. From the figure it is clear that CLMN and CLMH give low energy consumption



. Figure. 6: Execution time for each sink placement strategy

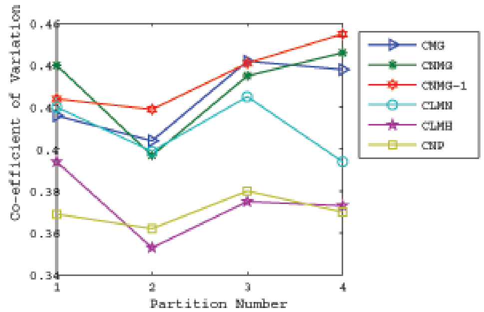


. Figure. 7: Hop count from farthest node to sink



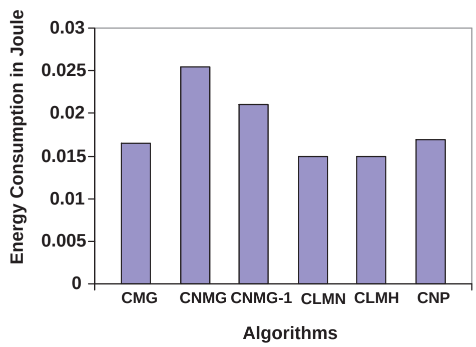
. Figure. 8: Average Hop count

for an event to deliver data to the sink among all other algorithms. Again CMG and CNP need almost same energy and among all these algorithms CNMG consumes highest energy for data transmission.

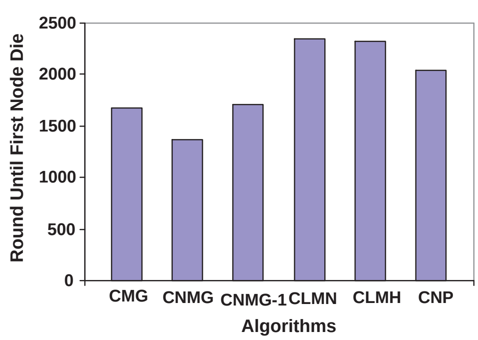


. Figure. 9: Co-efficient of Variation

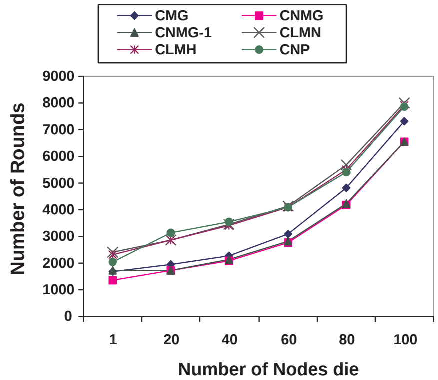
Lifetime of the network is measured in terms of first node die which is shown in Figure 11 and Figure 12 shows the number of rounds after which the last node of the network dies. From these two figures it is clear that CLMN, CLMH and CNP have longer lifetime in comparison with others.



. Figure. 10: Energy Consumption in each algorithm



. Figure. 11. First Node Die in each algorithm



. Figure. 12. Number of nodes dies vs. Round

Now Table II below summarized the performance analysis of all the six algorithms.
In the next section, the algorithms proposed in this paper are compared with another strategy which has been proposed in [Rehena et al. 2011]. In [Rehena et al. 2011], a square area is considered for deployment of the network and sinks are placed at each corner of the square region and network is partitioned into four parts. Each sink is responsible for a single partition only where it is placed. We refer to this sink placement strategy as Corner Position or CP strategy.

Table II. Analysis of characteristics for the Algorithms

Scheme	**Energy Consumption**	**Lifetime**	**Farthest Node Hop distance**
CMG	Moderate	Moderate	High
CNMG	Very High	Low	High
CNMG-1	High	Low	High
CLMN	Low	Good	High
CLMH	Low	Good	Low
CNP	Moderate	Moderate	Low

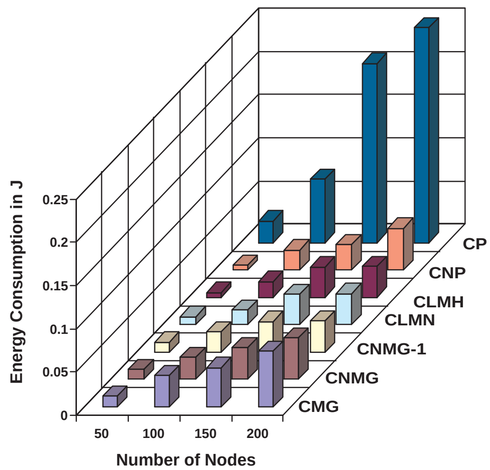
6.2 Comparison with CP

In this case also restricted flooding routing protocol is used for comparing the performance of our proposed algorithms with CP. Comparisons are done based on the basis of the following metrics.

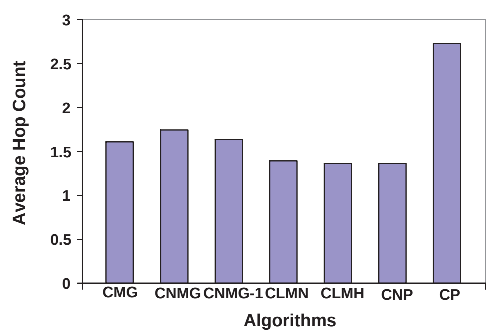
—Energy consumption for delivering an event to the corresponding sink, and

—Average hop count from all sensor nodes to respective sinks and hop count from farthest node in each partition.

For simulation a $200m \times 200m$ square area is considered and the network size is varied from 50 nodes to 200 nodes. The experiments are carried out five times and the result is averaged from these experimental outcomes.
Figure 13 shows energy consumption for all the sink placement strategies described in this paper along with the CP strategy. As expected, CP consumes highest energy for data transmission because of longer communication distance between sink and event location.
Figure 14 and Figure 15 depict the average hop count from sensor nodes to respective sinks and hop count from farthest node in each partition respectively. Similar to Figure 13, here also CP requires highest average hop count. Hop count from farthest node to sink is also high for CP compared to other algorithms except CNMG and CNMG-1.



. Figure. 13. Total Energy Consumption of the varying network size for each algorithm

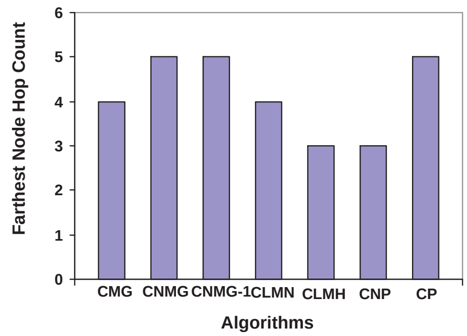


. Figure. 14. Average Hop count

7. CONCLUSION

The paper focuses on multiple-sink placement problem in a wireless sensor network. The network is assumed to be partitioned. We propose six sink placement strategies in the partitioned network. These strategies are compared and their advantages and disadvantages are discussed. Further the 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 the other algorithms. It has low execution time, low co-efficient of variation of hop counts and also hop distance of the farthest node is small for this algorithm. The CLMH strategy also demonstrates similar performance like CNP. However, execution time of CLMH is higher. Also while considering the network performance, CLHM and CNP strategies appear to be good solutions to the sink placement problem. All the strategies are compared with a simple strategy that places the sinks at the corners of the network and all of them proved to perform better than this one. We primarily focus on disaster management applications where fast data delivery and minimum energy consumption are the major two challenges. Thus, after placing the sinks in the network suitably, we aim to propose an appropriate routing mechanism



. Figure. 15. Hop count from farthest node to sink

for our disaster monitoring application. In the current work both sinks and nodes are considered to be static. In future, we aim at focusing on placement of multiple mobile sinks in a network region containing mobile sensor nodes.

REFERENCES

Bogdanov, A., Maneva, E., and Riesenfeld, S. 2004. Power-aware base station positioning for sensor networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. Vol. 1.

Chen, F. and Li, R. 2013. Sink node placement strategies for wireless sensor networks. *Wirel. Pers. Commun. 68,* 2, 303–319.

Chen, Y., Chan, E., and Han, S. 2005. Energy efficient multipath routing in large scale sensor networks with multiple sink nodes. In *Proceedings of the 6th International Conference on Advanced Parallel Processing Technologies*. APPT'05. Springer-Verlag, 390–399.

Flathagen, J., Kure, Ø., and Engelstad, E. P. 2011. Constrained-based multiple sink placement for wireless sensor networks. In *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on.* 783–788.

Forgy, E. 1965. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics 21,* 3, 768–769.

Guney, E., Aras, N., Altinel, I. K., and Ersoy, C. 2010. Efficient integer programming formulations for optimum sink location and routing in heterogeneous wireless sensor networks. *Comput. Netw. 54,* 11 (Aug.), 1805–1822.

Kabelikova, P. 2006. Graph partitioning using spectral methods. In *VSB - Technical University of Ostrava*.

Kim, H., Seok, Y., Choi, N., Choi, Y., and Kwon, T. 2005. Optimal multi-sink positioning and energy-efficient routing in wireless sensor networks. In *Proceedings of the 2005 International Conference on Information Networking: Convergence in Broadband and Mobile Networking*. ICOIN'05. Springer-Verlag, Berlin, Heidelberg, 264–274.

Li, J., Ji, S., Jin, H., and Ren, Q. 2008. Routing in multi-sink sensor networks based on gravitational field. In *Embedded Software and Systems, 2008. ICESS '08. International Conference on*. 368–375.

Oyman, E. and Ersoy, C. 2004. Multiple sink network design problem in large scale wireless sensor networks. In *Communications, 2004 IEEE International Conference on*. Vol. 6. 3663–3667 Vol.6.

Poe, W. and Schmitt, J. 2009a. Self-organized sink placement in large-scale wireless sensor networks. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on*. 1–3.

Poe, W. Y. and Schmitt, J. B. 2009b. Sink placement without location information in large-scale wireless sensor networks. In *Asian Internet Engineering Conference*. AINTEC '09. ACM, New York, NY, USA, 69–76.

Rehena, Z., Das, D., Roy, S., and Mukherjee, N. 2012. A comparative study of partitioning algorithms for wireless sensor networks. In *Advances in Computer Science and Information Technology. Networks and Communications*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 84. 445–454.

Rehena, Z., Roy, S., and Mukherjee, N. 2011. Topology partitioning in wireless sensor networks using multiple sinks. In *Computer and Information Technology (ICCIT), 2011 14th International Conference on*. 251–256.

Vincze, Z., Vida, R., and Vidacs, A. 2007. Deploying multiple sinks in multi-hop wireless sensor networks. In *Pervasive Services, IEEE International Conference on*. 55–63.

Zadeh, P., Schlegel, C., and MacGregor, M. 2011. Dynamic sink placement in wireless sensor networks. In *Advances in Networks and Communications*, N. Meghanathan, B. Kaushik, and D. Nagamalai, Eds. Communications in Computer and Information Science, vol. 132. 656–666.

Zeenat Rehena received her M.E degree in Computer Science and Engineering from Jadavpur University, Kolkata, India in 2009 and received B.Tech degree in Computer Science and Engineering from West Bengal University of Technology, Kolkata, India in 2005. She was a Research Fellow in School of Mobile Computing and Communication, Jadavpur University, Kolkata, India from 2009 to 2012. Since 2012 she has been a faculty of the Department of Computer Science and Engineering, Aliah University, India. Her research interests are in the area of Wireless Sensor Network.

Debasree Das received her M.E degree in Computer Science and Engineering from Jadavpur University, Kolkata, India in 2012 and received her B.Tech degree in Computer Science and Engineering from West Bengal University of Technology, Kolkata, India in 2005. Now she is working as an Assistant Engineer (IT&C) at WBSEDCL, India since 2012.

Dr. Sarbani Roy received the PhD degree in Engineering from Jadavpur University, Kolkata, India in July, 2008, and the M.Tech degree in Computer Science and Engineering, the M.Sc degree in Computer and Information Science and B.Sc Hons. degree in Computer Science from University of Calcutta, Kolkata, India in 2002, 2000 and 1998 respectively. Since 2006 she has been a faculty of the Department of Computer Science and Engineering, Jadavpur University. Her research interests are in the area of Wireless Sensor Networks, Cloud Computing and Distributed Computing.

Professor Nandini Mukherjee has joined as a faculty member in Department of Computer Science and Engineering, Jadavpur University, India in 1992. In 1996 she has been awarded Commonwealth Scholarship tenable at UK. She has completed her PhD in Computer Science from University of Manchester, UK in 1999. She has become a professor in Jadavpur University in 2006. Currently she has acted as the director of School of Mobile Computing and Communication, an interdisciplinary school of research in Jadavpur University from 2008 to 2014. Her research interests are in the area of High Performance Parallel Computing, Grid Computing and Mobile Computing. She is a senior member of IEEE.