Efficient Sink Scheduling for Data Travelling in WSNs with Delay Constraint

Sithara Mathews¹, T.R. Srinivasan²

¹Student, Computer Science and Engineering, Vivekanadha Institute of Engg & Tech for Women, Namakkal, India
²Head Of Department, Computer Science and Engineering, Vivekanadha Institute of Engg & Tech for Women, Namakkal, India

Abstract— The objective of system is to use sink mobility to prolong the network lifetime in wireless sensor networks where the information delay caused by moving the sink should be bounded. In this paper, a unified framework for analyzing the joint sink mobility, routing, delay, and so on is built. The induced sub problems are studied and then present efficient solutions for them. Then, we generalize these solutions and propose a novel approach—HSOT algorithm for the origin problem. Proposed system shows the benefits of involving a mobile sink and the impact of network parameters (e.g., the number of sensors, the delay bound, etc.) on the network lifetime. Furthermore, a study on the effects of different trajectories of the sink and to provide important insights for designing mobility schemes in real-world mobile WSNs is performed.

Keywords— Wireless Sensor Network; Deployment; Sink mobility; Scheduling.

I. INTRODUCTION

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts a radio-transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding. Due to the unique many-to-one (converge-cast) traffic patterns, the traffic of the whole network will be converge to a specific set of sensor nodes (e.g., neighboring nodes of the sink) and results in the hotspot problem. However, as long as the sink and sensor nodes are static, this issue cannot be fully tackled. Therefore, there is a recent trend to exploit mobility of the sink as a promising approach to the hotspot problem.

II. RELATED STUDIES

Efficient sensor deployment in a predetermined target region with minimum number of deployed sensors can be effectively achieved using Bipopulation-based Evolutionary Algorithm for Solving Full Area Coverage Problems (BEFAC) [1]. In [2] the Divide-and-Conquer deployment based on Triangles for Wireless Sensor Networks has been proposed. This algorithm makes use of static sensors and the coverage holes are evaluated collaboratively. There are coverage holes distributed over the monitored area, and the coverage hole can be polygonal. Because a polygon can be divided triangularly, exactly how to deploy sensors on a triangle is a priority. Three static sensors are located individually on the three nodes of triangles following random deployment.

Ant Colony-Based Scheduling Algorithm (ACB-SA) in [3] solve the EEC(Efficient-Energy Coverage) problem. This algorithm is a simplified version of the conventional ant colony optimization algorithm, optimized for solving the EEC problem. In [4], the energy-efficient coverage of WSNs is addressed. Instead of the deployment of sensors, the scheduling of sensor activations is performed through the Ant Colony Optimization (ACO) algorithm. The full-coverage state is obtained perfectly in free space. In [5], the camera network placement problem is solved by a binary Particle Swarm Optimization (PSO) method. With a model of the camera’s field of view, a higher success rate of camera placement is obtained.

In [6], coverage enhancement by considering an overlap-sensing ratio of neighbouring sensors with a directional sensing range in free space has been proposed. By adjusting the sensing direction of the nodes, the coverage area is increased with the reduction of computational complexity. In addition, a modified strategy is presented to shut off the redundant sensors so that network lifetime is prolonged. LRV (Least Recently Visited) algorithm in [7] efficiently and simultaneously solves the problems of coverage, exploration, and sensor network deployment. The basic premise behind the algorithm is that a robot carries network nodes as a payload, and in the process of moving around, emplaces the nodes into the environment based on certain local criteria. In turn, the nodes emit navigation directions for the robot as it goes by. Nodes recommend directions least recently visited by the robot, hence, the name LRV.

Obstacle-Resistant Robot Deployment (ORRD) algorithm proposed in [8], involves the design of a node placement policy, a serpentine movement policy, obstacle-handling rules, and boundary rules. By applying ORRD, the robot rapidly deploys a near-minimal number of sensor nodes to achieve full sensing coverage, even though there exist
unpredicted obstacles with regular or irregular shapes. Random deployment methods are based on the assumption that a randomly deployed excess number of sensors eventually acquire the full-coverage state. However, the number of deployed sensors is far from an optimal number and results in a higher hardware cost. In contrast, deterministic deployment approaches aim at assigning fewer sensors in the target region under a cost restriction. The redundancy of placement of all of the sensors in the target region should be diminished to reduce the number of sensors successfully. However, when non-penetratable obstacles are included in the target region, it is difficult to reduce the number of deployed sensors while retaining the full-coverage state.

III. PROBLEM DESCRIPTION

Full-area coverage in the predetermined target region is a challenging issue in wireless sensor network. Appropriate sensor deployment is required to perform surveillance and monitoring tasks successfully because a sensor has resource constraints, such as a physical sensing range, a battery power limit and a limited memory; as a result, a sensor can cover only a small part of the region of interest. Coverage can be considered to be a measure of quality of the services or tasks. Most of the existing approaches provide full-area coverage perfectly only in free space. Instead of fully random deployment and deterministic deployment, meta-heuristic search algorithms have been applied to various sensor deployment strategies. Sensor deployment is formulated as a search problem to minimize the number of deployed sensors as well as to maximize the coverage of the sensors similar to the mathematical approach.

The important characteristics required for a wireless sensor network are:

- Full-area coverage
- Minimize the number of deployed sensors
- Extended network lifetime
- Reduced delay in the network
- Minimize the cost of implementation

IV. EFFICIENT SINK SCHEDULING FOR DATA TRAVELLING IN WSNS

This section describes about different modules involved in establishing efficient sink scheduling in wireless sensor network. We propose a novel dynamic load handling algorithm for performing efficient sink scheduling by considering delay and traffic in the network.

This work makes use of two important algorithms:

- BEFAC
- HSDT

HSDT stands for Hop Scheduling and Data Travelling Algorithm. It is a novel approach for efficient sink scheduling introduced in this paper.

A. Construction of Wireless Sensor Network Model

The construction of nodes and the network model is done in this module. In traditional WSNS, a sensor node is severely constrained in terms of computation capability and energy reserves. Wireless sensor networks (WSN) will play a significant role at the “edge” of the future “Internet of Things”. WSN applications normally contain hundreds of sensor nodes whose ultimate goal is simply to report sensed data to the sink periodically, without need for any external control. As BEFAC algorithm provides an effective solution for full-area coverage problems with non-penetrable obstacles, this work makes use of BEFAC for deploying sensors.

A full-coverage is obtained in a predetermined target region with minimum number of deployed sensors using Bipopulation-based Evolutionary Algorithm for Solving Full Area Coverage Problems (BEFAC).

B. Estimation of Network Factors in Sink Mobility

Sinks are capable machines with rich resources. Sensors that are generating data are called sources. They transmit their data to one or more sinks for analysis and processing. Here, we consider data gathering from sensors, where sensor data are not aggregated on the way to the sink. That is, each sensor measurement arrives at the sink without any changes. Data transmission could take place either in a push mode or in a pull mode. In the push mode, sources actively send data to sinks; in pull mode, they transmit only upon sinks request. The main source-to-sink communication pattern is multi-hop message rely, as sinks are out of the transmission ranges of most of sources. The communication paths from reporting sources to a sink form a reverse multicast tree rooted at the sink. Uneven energy depletion causes energy holes and leads to degraded network performance. If sensors around a sink all run out of energy, the sink will be isolated from the network. If all sinks are isolated, then entire network fails. Since manual replacement/recharge of sensor batteries is often infeasible due to operational factors, it is desirable to minimize and balance energy usage among sensors.

C. Establishing a Novel Dynamic Load Handling Algorithm with Delay Constraints and Data Routing Techniques

In this, the full area coverage sensor deployment problem is analyzed. An entire region is divided into two mutually exclusive sub regions called Feasible Region(Rf) and Infeasible Region(Ri). Feasible region can be covered by sensors within the sensing range of the sensors. Additionally, sensors can be installed in feasible region. In contrast, sensors cannot be installed in infeasible region. In addition, infeasible region blocks the sight of the sensors, and a region behind infeasible region is not detectable. The feasible region can be divided into three sub regions. First, an uncovered region
(Ru) is defined as the region that is not covered by any sensors in the sensor network. A covered region (Rc) is defined as the region that is covered by at least one sensor. The detection or surveillance task can be performed in the region of interest by the specific sensor installed nearby. An overlapped region (Ro) is defined as the region that is covered by two or more sensors. Infeasible region is directly related to an obstacle region. If there are some obstacles in some regions that block the signals of the sensors, then the coverage of the obstacle region is not meaningful. The covered region is also a subset of the feasible region.

D. Implementing the Algorithm in the Energy efficient Communication Protocol

Bipopulation structure composed of a full-coverage population and a partial coverage population. A set of cells in the feasible region is closed under the proposed recombination operators. The recombination operators do not waste the computational cost of generating chromosomes with infeasible genes. The recombination operators are composed of four distinct operators namely insertion, deletion, fusion and substitution operators as proposed in [1].

1) Insertion Operator: The insertion operator generates an offspring chromosome by adding a gene into a parent chromosome. Physically, a sensor is deployed additionally for covering the uncovered region in the sensor deployment solution that is represented by each chromosome. The insertion operator randomly selects an uncovered cell in Ru and adds the sensor near the cell. The coverage rate becomes higher, and the number of sensors is also increased by one. The insertion operator is repeatedly performed in the initial generation, to generate the full-coverage solutions from scratch. Except for the initial generation, the insertion operator attempts to make partial-coverage solutions into full-coverage solutions by adding only one sensor during the other generations.

2) Deletion Operator: The deletion operator generates an offspring chromosome by randomly removing a gene of a parent chromosome in a full-coverage state. Because the deployed sensors are always in the feasible region, the deletion of a sensor is a simple way to provide a deployment solution with a smaller number of sensors. However, the deletion operator is likely to break the full-coverage state by removing a sensor from the full-coverage solutions. Only the totally redundant sensors can be removed while keeping the full coverage state. Thus, the usage of the deletion operator is limited to removing less meaningful solutions with an excess number of sensors from the partial-coverage population because the partial-coverage solutions with more sensors than the full-coverage populations are far from the full-coverage state with a minimum number of sensors.

3) Fusion Operator: The fusion operator generates an offspring chromosome by merging two genes into a new gene of a parent chromosome. The fusion operator randomly selects a gene and finds the gene that is the nearest neighbor of the selected gene. Attempting to create a new gene by merging the two genes within the square region defined by the position of the two genes, the fusion operator checks the validity of the fusion operator. When the position of the new gene is in the infeasible region Ri, a failure flag is returned, and the result of the fusion operator is not reflected in the chromosome. In contrast, when the center of the new gene is in the feasible region Rf, a success flag is returned, and the newly merged gene is reflected in the chromosome. Physically, two sensors are removed, and a new sensor is added in the intermediate cell of the two removed sensors.

4) Substitution Operator: The substitution operators generate an offspring chromosome by changing a gene of a parent chromosome a small amount. There are two types of substitution operators that are used in the BEFAC. One type is a uniform substitution operator, and the other type is a biased substitution operator. The former type is used to provide the genetic diversity of the full-coverage population, and the latter type is used to increase the coverage rate by moving toward the uncovered region of the partial-coverage population.

E. Performance Evaluation

The performance of the algorithms is measured by the average convergence speed with respect to the number of sensors while retaining the full-coverage state and by the number of fitness evaluations. As the number of deployed sensors is decreased, it is difficult to have a reduction in the number of sensors while keeping the full-coverage state. The redundancy among the deployed sensors is also reduced, which requires a high computational cost to approach the optimal number of sensors. As a result, the performance of the algorithm can be represented by the number of fitness evaluations when the number of deployed sensors is reduced.

V. PROPOSED HSDT ALGORITHM

-Run the Hop Optimization algorithm to get optimal Solution
It involves grouping of sensor nodes as group1 group2 etc. Each group uses a particular path for data transmission. For example for that particular transmission group1 will be using a linear path. This will reduce the complexity of building routing path for each data transfer and thus reducing the delay in the network.
- Randomly select a path that contains all the sink sites
While selecting the path it should be considered that all sink node in the region is included. Missing some node will reduce data precision in the aggregated result. Make the sink node to move through selected trajectory.
-Calculate the Source timings stayed in node and Travelling time from Source node to next node until destination node.
- Choose the Longest network lifetime as best Data travelling Path  
- Calculate linear trajectory, Boundary trajectory and arbitrary trajectory values.  

By initial comparison of time taken for different trajectories for data travel we can conclude which path will take minimum time for data to reach from source to destination.

VI. CONCLUSION AND FUTURE WORKS

In the proposed system, by building a unified framework for analyzing this joint sink mobility, routing, delay, and so on we solve the induced sub problems and present efficient solutions for them. Then, we generalize these solutions and propose a polynomial-time optimal algorithm for the origin problem.

As for the future work, we plan on extending current work to accommodate networks with multiple sinks. Furthermore, using the centralized optimal algorithm developed in this paper as performance benchmark, we want to design distributed online algorithms for fast execution in large-scale networks and test them in real world experiments.

REFERENCES