Efficient Data Gathering and Improving Network Lifetime in Wireless Sensor Networks

C. Murali M.E.¹, Dr. M. Sabrigiriraj Ph.D.²

¹ PG Student, Department of Computer Science and Engineering, SVS College of Engineering, Coimbatore, Tamilnadu, India
² HOD, Department of Computer Science and Engineering, SVS College of Engineering, Coimbatore, Tamilnadu, India.

Abstract— In large-scale wireless sensor networks the data-gathering mechanism to introducing mobility into the network. We consider the location service in a WSN, where each sensor needs to maintain its location information by 1) frequently updating its location information within its neighboring set of polling points in the network. In the previous work a mobile data collector, for convenience called an M-collector for gathering information from the sensor and send to sink node. M-collector could be a mobile robot or a vehicle equipped with a powerful transceiver and battery, working like a mobile base station and gathering data while moving through the field. An M-collector starts the data-gathering tour periodically from the static data sink, polls each sensor while traversing its transmission range, then directly collects data from the sensor in single-hop communications, and finally transports the data to the static sink. Since data packets are directly gathered without relays and collisions, the lifetime of sensors is expected to be prolonged. In this paper, we mainly focus on the problem of minimizing the length of each data-gathering tour and remove the redundancy for data gathering. We develop a stochastic sequential decision framework to analyze this problem. Under a Markovian mobility model, the location update decision problem is modeled as a Markov Decision Process (MDP). For the work with strict distance/ time and energy constraints, we consider utilizing single and multiple M-collectors and propose a data-gathering algorithm where multiple M-collectors traverse through several shorter subtours concurrently to satisfy the distance/time and energy constraints. To identify the redundancy of data from the sensor using Pearson auto correlated technique. Our proposed system for mobile data gathering scheme can improve the scalability and balance the energy consumption among sensors.

Keywords— M-collector, Markov Decision Process, Pearson auto correlated, spanning tree, polling points, data gathering.

I. INTRODUCTION

Nowadays, wireless sensor networks (WSNs) have emerged as a new information-gathering paradigm in a wide range of applications, such as medical treatment, outer-space exploration, battlefield surveillance, emergency response, etc. Sensor nodes are usually thrown into a large-scale sensing field without a preconfigured infrastructure. Before monitoring the environment, sensor nodes must be able to discover nearby nodes and organize themselves into a network. Most of the energy of a sensor is consumed on two major tasks: sensing the field and uploading data to the data sink. Energy consumption on sensing is relatively stable because it only depends on the sampling rate and does not depend on the network topology or the location of sensors. On the other hand, the data-gathering scheme is the most important factor that determines network lifetime. Although applications of sensor networks may be quite diverse, most of them share a common feature. Their data packets may need to be aggregated at some data sink.

In a homogeneous network where sensors are organized into a flat topology, sensors close to the data collector consume much more energy than sensors at the margin of the network, since they need to relay many packets from sensors far away from the data collector. As a result, after these sensors fail, other sensors cannot reach the data collector and the network becomes disconnected, although most of the nodes can still survive for a long period. Therefore, for a large-scale data-centric sensor network, it is inefficient to use a single static data sink to gather data from all sensors. In some applications, sensors are deployed to monitor separate areas. In each area, sensors are densely deployed and connected, whereas sensors that belong to different areas may be disconnected. Unlike fully connected networks, some sensors cannot forward data to the data sink via wireless links. A mobile data collector is perfectly suitable for such applications. A mobile data collector serves as a mobile “data transporter” that moves through every community and links all separated subnetworks together. The moving path of the mobile data collector acts as virtual links between separated subnetworks.

We consider applications, where sensing data are generally collected at a low rate and is not so delay sensitive that it can be accumulated into fixed-length data packets and uploaded once in a while. To provide a scalable data-gathering scheme for large-scale static sensor networks, we utilize mobile data collectors to gather data from sensors. Specifically, a mobile data collector could be a mobile robot or a vehicle equipped with a powerful transceiver, battery, and large memory. The mobile data collector starts a tour from the data sink, traverses the network, collects sensing data from nearby nodes while moving, and then returns and uploads data to the data sink. Since the data collector is mobile, it can move close to sensor nodes, such that if the moving path is well planned, the network lifetime can be greatly prolonged. Here, network lifetime is defined as the duration from the time sensors start sending data to the data sink to the time when a certain percentage of sensors either run out of battery or cannot send.
data to the data sink due to the failure of relaying nodes. In the following, for convenience, we use M-collector to denote the mobile data collector.

We consider the data-gathering problem in which the M-collector can visit the transmission range of every static sensor, such that sensing data can be collected by a singlehop communication without any relay. While an M-collector is moving, it can poll nearby sensors one by one to gather data. Upon receiving the polling message, a sensor simply uploads the data to the M-collector directly without relay. We define the positions where the M-collector polls sensors as polling points. When an M-collector moves to a polling point, it polls nearby sensors with the same transmission power as sensors, such that sensors that receive the polling messages can upload packets to the M-collector in one hop. After gathering data from sensors around the polling point, the M-collector moves directly to the next polling point in the tour. Thus, the problem of finding the optimal tour can be considered as the problem of determining the locations of polling points and the order to visit them. Before an M-collector starts a data-gathering tour, it needs to determine the positions of all polling points and which sensors it can poll at each polling point. We define the neighbor set of a point in the plane as the set of sensors that can upload data to the M-collector directly without relay, if the M-collector polls sensors at this point. Since the M-collector can only collect data at polling points, each sensor must be in the neighbor set of at least one polling point to upload data without relay. In other words, the union of neighbor sets of all polling points must cover all sensors. Thus, each data-gathering tour of an M-collector consists of a number of polling points and the straight line segments connecting them.

In that existing system presents [1] new data-gathering mechanisms for large scale sensor networks when single or multiple Mcollectors are used. We focus on the problem of minimizing the length of each data-gathering tour and formulate it into a mixed-integer programming (MIP). We present a spanning tree covering algorithm for the single M-collector case. We also consider utilizing multiple M-collectors and propose a data-gathering algorithm where multiple Mcollectors traverse through several shorter subtours concurrently to satisfy the distance/time constraints.

In this previous work the multipath routing increases the probability of reliable data delivery. In multi-path routing, the energy cost overhead for data retransmissions due to link failure or node failure and an alternate path construction is minimized. Sensor nodes must be able to discover nearby nodes and organize themselves into a network. After these sensors fail, other sensors cannot reach the data collector and the network becomes disconnected, although most of the nodes can still survive for a long period. Therefore, for a large-scale data-centric sensor network, it is inefficient to use a single static data sink to gather data from all sensors. To overcome this problem determining optimal sink positions that maximize the network lifetime by reducing energy consumption related to data transmissions from sensor nodes to different sinks. Balanced graph partitioning techniques were used to split the entire WSN into connected sub networks.

In this paper, find the shortest path between the polling points using MDP. To gather the information from the sensor, first identify the sensor information could be redundant or not. Identify the redundant data from the sensors using spatially auto correlated techniques. We use a pearson auto correlation-based search tree structure to propose new processing strategies for correlation-based similarity range. We carry out extensive simulations. The effectiveness of our proposed algorithms is verified by comparing with another data-gathering algorithm.

II. RELATED WORK

Here, we briefly outline some related work on data-gathering mechanisms in WSNs. It has been widely known that data routing can cost significant energy expenditure in sensor networks with a flat topology. To overcome this problem, some works in the literature have introduced a hierarchy to the network [2]–[5], [6], and [7]. In such a network, sensor nodes are organized into clusters and form the lower layer of the network. At the higher layer, cluster heads collect sensing data from sensors and forward data to the outside data sink. In general, such two-layered hybrid networks are more scalable and energy-efficient than homogeneous sensor networks. A cluster head acts not only as a data aggregation point for collecting sensing data from sensors but also as a controller/scheduler to make various routing and scheduling decisions. In a homogeneous network, where all nodes have identical capability and energy at the beginning, some of the nodes are selected to serve as cluster heads [2]–[5]. However, cluster heads will inevitably consume more energy than other sensor nodes. To avoid the problem of cluster heads failing faster than other nodes, sensor nodes can become cluster heads rotationally [2]. In this type of network, since every sensor node may possibly become a cluster head, each of them has to be “powerful” enough to handle incoming and outgoing traffic and cache sensing data, which will increase the overall cost of the entire sensor network. Furthermore, selecting cluster heads dynamically results in high overhead due to the frequent information exchange among sensor nodes. Some efforts have been made to improve the intrinsic disadvantage of homogeneous networks by introducing a small number of resource rich nodes. Unlike homogeneous networks, a heterogeneous sensor network contains a small number of resource-rich nodes together with a large number of resource-limited basic sensor nodes. Basic sensor nodes have limited communication capability and mainly focus on sensing the environment, whereas resource-rich nodes are equipped with more powerful transceivers and batteries. In [6] and [7], resource-rich nodes act as cluster heads, and the network is organized into a two layered hierarchical network. However, it is generally difficult to deploy powerful cluster heads to appropriate positions without learning the network topology.

In [8], a data-gathering scheme was proposed to minimize the maximum average load of a sensor by jointly considering the problems of movement planning and routing. Based on the assumption that sensors are distributed according to a Poisson process, the average load of a sensor can be estimated as a function of the node density.

In [9], the benefits of mobile relay in WSNs were investigated. It was shown that when a mobile node is used as mobile relay, the network lifetime is improved compared with an all-static network. It was also proved that the mobile relay
needs to stay only within a two-hop radius of the sink. In [10], an offline heuristic algorithm was proposed to compute the periodic trajectories of mobile base stations according to the data generation rates of sensors and their locations. The work in [11] discussed the event-collection problem by leveraging the mobility of the sink node and the spatial–temporal correlation of the events, in favor of maximizing the network lifetime with a guaranteed event-collection rate.

This problem was modeled as a sensor selection problem, and the design of a feasible movement route for the mobile sink was analyzed to minimize the velocity requirements for a practical system. In [12], a rendezvous-based data-collection approach was proposed for two scenarios, where the mobile base station has fixed or variable moving tracks. Two approximate algorithms were provided for the respective scenarios to minimize the distance of multihop routing paths for local data aggregation under the constraint that the tour length of the mobile base station is no longer a threshold. In [13], a stochastic compressive data-collection protocol for mobile WSNs, named SMITE, was presented. SMITE consists of three parts: 1) random collector election; 2) stochastic direct transmission from common nodes to collectors when common nodes are in the collectors’ transmission range; and 3) angle transmission from collectors to the mobile sink when collectors gather enough data using a predictive method.

In [14], a protocol to minimize the overall network overhead and energy expenditure associated with the multihop data retrieval process was proposed. Sensors are organized into clusters, and each cluster head performs data filtering upon raw data exploiting potential spatial–temporal data redundancy and forwards the filtered information to appropriate end nodes with sufficient residual energy, located in proximity to the mobile sink’s trajectory.

In [15], the tradeoff between energy saving and data-gathering latency in mobile data gathering was studied by exploring a balance between the relay hop count of local data aggregation and the moving tour length of the mobile collector. Two algorithms were correspondingly proposed, in which a subset of sensors is selected as polling points to buffer locally aggregated data and upload the data to the mobile collector when it arrives. Meanwhile, when sensors are affiliated with these polling points, it is guaranteed that any packet relay is bounded within a given number of hops.

The contributions of this paper [1] can be summarized as follows.

1) We propose new data-gathering mechanisms for large scale sensor networks when single or multiple Mcollectors are used.

2) We focus on the problem of minimizing the length of each data-gathering tour and formulate it into a mixed-integer programming (MIP).

3) We propose a spanning tree covering algorithm for the single M-collector case.

4) We also consider utilizing multiple M-collectors and propose a data-gathering algorithm where multiple Mcollectors traverse through several shorter subtours concurrently to satisfy the distance/time constraints.

We consider the problem of finding the shortest moving tour of an M-collector that visits the transmission range of each sensor. The positions of sensors are either the polling points in the data-gathering tour or within the one hop range of the polling points. For the sake of simplicity, we assume that M-collectors move at a fixed speed and ignore the time for making turns and data transmission, such that we can roughly estimate the time of a data-gathering tour by the tour length. Clearly, by moving through the shortest tour, data can be collected in the shortest time such that the users will have the most up-to-date data. We refer to this as the single-hop data gathering problem (SHDGP).

The basic idea behind our proposed greedy algorithm is to choose a subset of points from the candidate polling point set, each of which corresponds to a neighbor set of sensors. At each stage of the algorithm, a neighbor set of sensors can be covered when its corresponding candidate polling point is chosen as a polling point in the data-gathering tour. The algorithm will terminate after all sensors are covered.

The data-gathering algorithm with multiple M-collectors can be described as follows. First, find the polling point set P by running the spanning tree covering algorithm. Then, find the minimum spanning tree (T, V, E) on polling points. We refer to the minimum spanning tree on polling points as the spanning covering tree.

III. EXISTING SYSTEM

In this paper [1], we mainly focus on the problem of minimizing the length of each data-gathering tour and refer to this as the single-hop data-gathering problem (SHDGP). We first formalize the SHDGP into a mixed-integer program and then present a heuristic tour-planning algorithm for the case where a single M-collector is employed. For the applications with strict distance/time constraints, we consider utilizing multiple M-collectors and propose a data-gathering algorithm where multiple M-collectors traverse through several shorter subtours concurrently to satisfy the distance/time constraints. Our single-hop mobile data gathering scheme can improve the scalability and balance the energy consumption among sensors. It can be used in both connected and disconnected networks. Simulation results demonstrate that the proposed data-gathering algorithm can greatly shorten the moving distance of the collectors compared with the covering line approximation algorithm and is close to the optimal algorithm for small networks.

IV. PROPOSED WORK

While an M-collector is moving, it can poll nearby sensors one by one to gather data. Upon receiving the polling message, a sensor simply uploads the data to the M-collector directly without relay. We define the positions where the M-collector polls sensors as polling points. When an M-collector moves to a polling point, it polls nearby sensors with the same transmission power as sensors, such that sensors that receive the polling messages can upload packets to the M-collector in one hop. After gathering data from sensors around the polling point, the M-collector moves directly to the next polling point in the tour. Thus, each data-gathering tour of an M-collector consists of a number of polling points and the straight line segments connecting them.

Thus, the problem of finding the optimal tour can be considered as the problem of determining the locations of polling points and the order to visit them. Before an
Mcollector starts a data-gathering tour, it needs to determine the positions of all polling points and which sensors it can poll at each polling point. We define the neighbor set of a point in the plane as the set of sensors that can upload data to the Mcollector directly without relay, if the M-collector polls sensors at this point. Since the M-collector can only collect data at polling points, each sensor must be in the neighbor set of at least one polling point to upload data without relay. In other words, the union of neighbor sets of all polling points must cover all sensors.

M-collector gathers the information from the sensor before that find the sensor information’s are similar or not. So our proposed method identifies the similarity between the sensor using Pearson auto correlation coefficient. In practice, we rely on a single correlation measure motivated by the properties of the two sensors. The proposed detection algorithm is based on the Pearson auto correlation coefficient (PCC), one of the most widely used correlation measures. Given two sequences described by two sensors P and Q with N samples, the PCC is defined as

\[
r = \frac{\text{cov}(P,Q)}{\sigma_P \sigma_Q} = \frac{\sum_{i=1}^{N}(P_i - \bar{P})(Q_i - \bar{Q})}{\sqrt{\sum_{i=1}^{N}(P_i - \bar{P})^2 \sum_{i=1}^{N}(Q_i - \bar{Q})^2}}
\]

where \(\text{cov}(P,Q)\) is the sample covariance, \(\sigma_P\) and \(\sigma_Q\) are sample standard deviations, and \(\bar{P}\) and \(\bar{Q}\) are sample means.

If both the sensor have the similar data, to select the best sensor based on the distance between polling points to sensor and which node have maximum energy.

**Energy**

The power consumed for transmitting a packet is given by the

**Consumed energy** \(= TP(i,j) \times t\) (1)

Where \(TP\) is the transmitting power of node \(i\) to \(j\) and \(t\) is transmission time. The power consumed for receiving a packet is given by

The value \(t\) can be calculated as

\[
t = \frac{\text{Ds}}{\text{Dr}}
\]

\(\text{Ds}\) is Data size and \(\text{Dr}\) is Data rate

Hence, the residual energy \((E)\) of each node can be calculated using

\[E = \text{Current energy} - \text{Consumed energy}\]

**Distance**

Each node has a coordinate \((x,y)\). Supposed that the coordinate of \(u\), \(v\) is \((x_1,y_1)\), \((x_2,y_2)\), respectively. The distance between \(u\) and \(v\) is

\[
\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}
\]

To find probability based on the three constraints like similarity, distance and energy. Update the polling points using Markov Decision Process. Find the shortest path between the polling points spanning tree covering algorithm.

More formally, consider a set of points \(\{x_1, \ldots, x_n\}\) with a metric \(d(x_i, x_j)\). We first construct a symmetrized K nearest neighbor graph \(G\) over the points and assign a weight \(W_{ik} = \exp(-d(x_i, x_k)/\sigma)\) to each undirected edge in the graph. The weights are symmetric and \(W_{ii} = 1\) as we include self-loops; \(W_{ij} = 0\) for all non-neighbors. Note that the product of weights along a path in the graph relates to the total length of the path in the same way as the edge weights relate to the distances between the corresponding points. The one-step transition probabilities \(p_{ik}\) from \(i\) to \(k\) are obtained directly from these weights:

\[
p_{ik} = \frac{W_{ik}}{\sum_{j} W_{ij}}
\]

**Spanning tree covering algorithm**

Using this algorithm find the shortest path between the polling points for single M-Collector.

Create an empty set \(P_{curr}\)

Create a set \(U_{curr}\) containing all sensors

Create a set \(L\) containing all candidate polling points

While \(U_{curr} \neq \Phi\)

Find the polling points \(l \in L\), which minimizes \(a = \text{cost}(\text{nb}(l))\)

Add the corresponding polling point of \(\text{nb}(l)\) to \(L\)

Remove sensors in \(\text{nb}(l)\) from \(L\)

End while

Find the approximate shortest tour on polling points in \(P_{curr}\)

**Data-gathering algorithm with multiple M-Collectors**

Find the polling point set \(P\)

Find the spanning covering tree \(T\) on all polling points in \(P\)

For each vertex \(v\) in \(T\), calculate the weight value \(\text{Weight}(v)\)

While \(T \neq \Phi\)

Find the deepest leaf vertex \(u\) in \(T\)

Let the root of the subtree \(t\), \(\text{Root}(t)=u\)

While \(\text{Weight}(\text{Parent}(\text{Root}(t))) \leq \frac{l_{max}}{2}\)

\(\text{Root}(t)=\text{Parent}(\text{Root}(t))\)

End while

Add all child vertices of \(\text{Root}(t)\) and edges connecting them into \(t\) and remove \(t\) from \(T\)

Update weight value of each remaining vertex in \(T\)

End while

A straightforward way to deal with the tour planning problem of multiple M-collectors can be borrowed from the traditional delivery vehicle routing problem, where a number of trucks are sent out from the facility center, and each of them moves through a subtour, delivers packages home by home, and finally returns to the facility center before the center is closed.

**IV. EXPERIMENTAL RESULTS**

![Fig 1. Tour Length Graph](http://www.ijcttjournal.org)

We also compare the tour length of the MDP based spanning tree covering algorithm with the spanning tree covering algorithm in larger networks. We measured the
relative tour length of the MDP based spanning tree covering algorithm with the spanning tree covering algorithm. From the Fig.1 graph number of nodes represents in X-axis and tour length represented in Y-axis. Analysis results our proposed system has better results than existing algorithm.

Fig. 2. Polling point Graph

We also compare the number of polling points required for gathering the information from the sensor using MDP based spanning tree covering algorithm with the spanning tree covering algorithm in larger networks. We measured the relative polling points of the MDP based spanning tree covering algorithm with the spanning tree covering algorithm. From the Fig.2 graph number of nodes represents in X-axis and number of polling points represented in Y-axis. Analysis results our proposed system has better results than existing algorithm.

V. CONCLUSION AND FUTURE WORK

We proposed a mobile data-gathering scheme for large-scale sensor networks. We introduced a mobile data collector, called an M-collector, which works like a mobile base station in the network. An M-collector starts the data gathering tour periodically from the static data sink, traverses the entire sensor network, polls sensors and gathers the data from sensors one by one, and finally returns and uploads data to the data sink. Our mobile data-gathering scheme improves the scalability and solves intrinsic problems of large-scale homogeneous networks. By introducing the M-collector, data gathering becomes more flexible and adaptable to the unexpected changes of the network topology. In addition, data gathering by M-collectors is perfectly suitable for applications, where sensors are only partially connected. For some applications in large scale networks with strict distance/time constraints for each data-gathering tour, we introduced multiple M-collectors by letting each of them move through a shorter subtours than the entire tour. Our proposed system is mainly concentrate on reduce the tour length and efficient information gathering from the sensors. We also extend our work to identify and avoid the redundant information gather from the sensors. It can prolong the network lifetime significantly compared with the scheme that has only a static data collector and the scheme in which the mobile data collector can only move along straight lines.

REFERENCES


