

## Efficient Data Gathering Mechanism using Mobile Collector in Wireless Sensor Networks

Mr. Shanmugapriyan.R<sup>1</sup>, Mr. Sivasakthivel.A<sup>2</sup>

1 – Student, Department of ECE, Arunai Engineering College, Thiruvannamalai.

2 – Assistant Professor, Department of ECE, Arunai Engineering college, Thiruvannamalai Tamil Nadu, India.

---

**Abstract:** In this paper, we proposed a new data-gathering mechanism for large-scale wireless sensor networks by introducing mobility into the network. A mobile data collector, for convenience called an M-collector in this paper, 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. Here the complexity in computation is reduced by introducing new mechanism and it leads to more efficiency in network lifetime and transmission range. Simulation results demonstrate that the proposed data-gathering algorithm can greatly shorten the moving distance of the collectors and power of sensors.

---

### I. Introduction

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. Data-gathering scheme is the 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 sub networks together. The moving path of the mobile data collector acts as virtual links between separated sub networks. 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 wellplanned, 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.

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

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

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

4) We carry out extensive simulations.

### II. Related Works

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 [7]–[8]. 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.[7] 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. 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 resource-rich nodes act as cluster heads, and the network is organized into a two-layered hierarchical network. The work in [9] 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 [13], 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. Meanwhile, when sensors are affiliated with these polling points, it is guaranteed that any packet relay is bounded within a given number of hops. However, it is generally difficult to deploy powerful cluster heads to appropriate positions without learning the network topology.

### III. Data Gathering Algorithm Of M-Collector

Here 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 up to date data.

#### 3. The Distributed Spanning Tree Algorithm

The first algorithm is a modification of the distributed spanning tree formation algorithm for general networks. We modify this general algorithm below so that clusters which are sub-trees are also formed with energy considerations of the WSN. We assume that the sensor nodes are distributed randomly and densely over the area to be monitored and the sensor field can be mapped into a two-dimensional space. Furthermore, all the sensor nodes have identical and fixed transmission ranges and hardware configurations and each sensor node can monitor its power level  $EP$ .

##### 3.1 Description of the Algorithm

The algorithm we propose is described informally as follows. The sink periodically starts the algorithm by sending a *PARENT* message to its neighbors. Any node  $i$  that has not received a *PARENT* message before sets the sender as its parent, sends *ACK* ( $i$ ) message to its parent and sends a *PARENT* ( $i$ ) message to all of its neighbors. We provide a *depth of sub tree* parameter  $d$  as the modification to the above classical algorithm to form a spanning tree. Every node that is designated a parent performs  $n \text{ hops} = (n \text{ hops} + 1) \text{ MOD } d$  to append to its outgoing message. The recipient of the message with  $n \text{ hops} = 0$  are the *SUBROOTS*, and  $n \text{ hops} \leq d$  are *INTERMEDIATE* nodes or *leaf* depending on their level within a sub-tree.

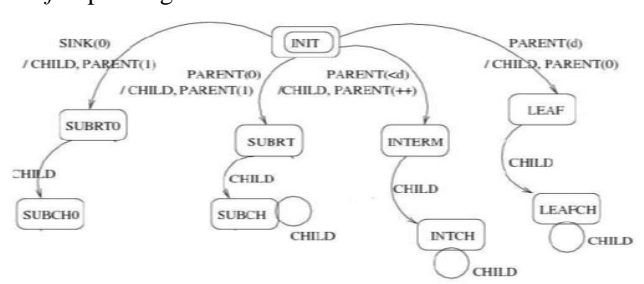


Fig 1. Finite state machine diagram of DSTA

The state diagram of Fig. 1 depicts the operation of the Distributed SpanningTree Algorithm (DSTA). The algorithm is initiated by the sink at regular intervals. Any ordinary node that has not been labeled before, receiving a PARENT message from an upper node, labels itself according to the number of hops the message has traveled which is shown by the parameter of the PARENT message any further change of states between sub root, intermediate and leaf nodes are not shown for simplicity. The following is a list of messages used in DSTA:

- {PARENT: Sent by a parent to the neighbors soliciting for children.
- {CHILD: Sent by the child to parent acknowledging to be a successor.
- {TIMEOUT: Internal message informing a timeout has occurred. This message prevents a sub root waiting indefinitely for acknowledgements from potential children.

The message contains the following fields:

- {Sender: SINK, SUBROOT, SUBROOT0, INTERMED, LEAF;
- {Type: PARENT, CHILD;
- {n hops: integer showing the number of hops the message has travelled. If the number of hops in the message is equal to zero, the node labels itself as the SUBROOT. Else if the number of hops is smaller than the allowed depth *d* of the sub-tree, the node is an intermediate (INTERM) node.
- Once the number of hops equals the depth, the node is classified as a LEAF. Each labeled node Acknowledge its parent by the CHILD message.

The following is the list of sensor node states:

- { SUBRT : A node is labeled as a sub root as the message it has received from its parent has n hops = 0. { SUBCH : A Sub root node has at least one confirmed child in the local tree. { INTERM : A node is an intermediate node, that is, it is not a sub root or a leaf node. { INTCH : An intermediate node with at least one child { LEAF : A node that is the leaf of a local spanning tree.
- { LEAFCH : A leaf node with at least one child.
- {SUBRT0: A sub root node that has received a SINK message {SUBCH0: A sub root 0 node that has at least one child.

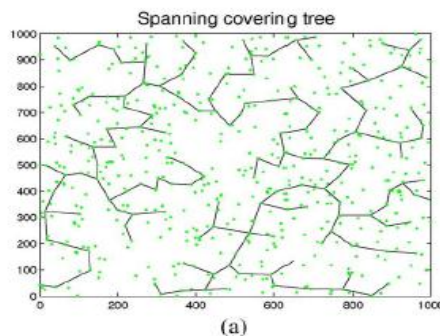
**Remark 1. Energy Consideration:** A sensor node rejects being labeled as sub-root if its energy level is below a threshold, for example, two thirds of *EP*. This is required as a sub root will have more message transfers than an ordinary node. A branch of the spanning tree formed constitutes a cluster where a sub root node is the cluster head. Sub roots may have other attributed roles in application specific settings. For our purpose, each sub root has the capability to manipulate or filter any incoming message to it during converges cast.

### 3.2 Analysis of DSTA

In this section, we analyze the number of steps (count of messages) to form the spanning trees using DSTA and comment on its performance. Based on the state machine of Fig.1, the labeling of a sensor node as SUBROOT, INTERMED or LEAF requires two messages called PARENT and CHILD. The first message is sent by the parent soliciting children and the second message is the acknowledgement of the child to its parent.

**Theorem 1:** Time complexity of DSTA is  $O(D)$  where *D* is the diameter of the network from the sink to the furthest leaf and its message complexity is  $O(n)$ .

**Proof:** The time required for the algorithm is clearly the diameter *D* of the network. Once a node is labeled and has a designated parent, it will only send a message to its neighbors once. If  $\phi$  is the maximum degree of the network graph, total number of messages is  $\phi * n$  and for small  $\phi$ , message complexity is  $O(n)$ .



(a)  
Fig (a) Spanning tree formation

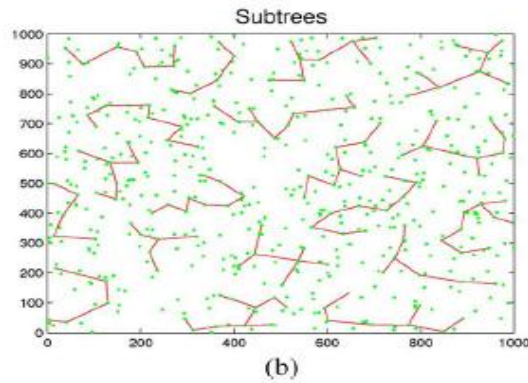


Fig (b) Formation of sub trees

The four major steps of the data-gathering algorithm with multiple M-collectors: 1) build the spanning covering tree; 2) decompose the spanning covering tree into a set of sub trees; 3) find an approximate shortest sub tour on the points of each sub tree; and 4) sensing data collected from sensors are forwarded to the nearest M-collector to the data sink.

#### IV. Wsn With M-Collector

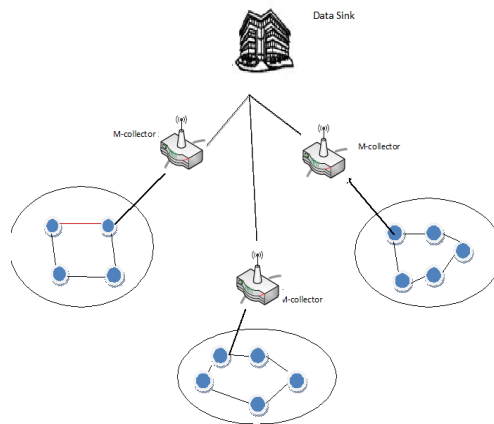


Fig2. Architecture of WSN with MC

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. For example, let  $P = \{p_1, p_2, \dots, p_t\}$  denote a set of polling points and  $DS$  be the data sink. Then, the moving tour of the M-collector can be represented by  $DS \rightarrow p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_t \rightarrow DS$ . 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 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. In some existing work, the transmission range of an omnidirectional antenna was simply assumed to be a disk-shaped area around the transceiver. Based on this assumption, given a point in the plane, the neighbor set of this point consists of all sensors within the disk-shaped area around this point. However, due to the uncertainties of a wireless environment, such as signal fading, reflection from walls and obstacles, and interference, it is hard to estimate the boundary of the transmission range without real measurement [14], [15]. Therefore, in practice, it is almost impossible to obtain the neighbor set of an unknown point, unless the M-collector has moved to this point and tested wireless links between it and its one-hop neighbors, or a sensor has

been placed at this point and acquired all its one hop neighbors during the neighbor discovering phase. Thus, it is only possible to test a finite number of points and their corresponding neighbor sets in the plane, and we must select polling points from this finite set of points, which we refer to as the *candidate polling point set*. If the connection pattern of sensors can be obtained, or in other words, we know the one hop neighbors of every sensor, the position of each sensor can be a candidate polling point, since the neighbor set of this point is already known. However, the connection pattern may not always be available before sending out M-collectors, unless the network is completely connected so that the connection pattern can be reported to the data sink via wireless transmissions. To obtain the candidate polling points without the information on the connection pattern, after sensors are deployed, one or more M-collectors need to explore the entire sensing field. While exploring, each M-collector can broadcast “Hello” messages periodically with the same transmission power as sensors. Each sensor that can decode the “Hello” message correctly replies with an “ACK” message to notify the M-collector where it is. Upon receiving the “ACK” message from the sensor, the M-collector marks its current location as a candidate polling point and adds the ID of the sensor into the neighbor set of this candidate polling point. Thus, all wireless links between sensors and the M-collector at the candidate polling points is bi-directionally tested. In addition, each sensor can also discover its one-hop neighbors by broadcasting the “Hello” messages during the neighbor discovering phase. After the sensor reports the IDs of its one-hop neighbors to the M-collector by including the information into the “ACK” message, the position of the sensor can also become a candidate polling point.

### V. Performance Evaluation

We have conducted extensive simulations to validate the proposed algorithms. In the simulations, we assume that a bunch of sensor nodes is uniformly deployed in the sensing field. For data gathering, we evaluate the tour length of a single M-collector in both small and large networks, compare the relative network lifetime of the proposed spanning tree covering algorithm

#### A) Network Efficiency:

Our energy model is like the energy model in [23]. In this model energy consumption for transmitting  $K$  bit is

$$E_{TX}(K, d) = E_{elec} \times K + \epsilon_{amp} \times K \times d^2$$

equal to:

The energy for receiving  $K$  bit is equal to:

$$E_{RX}(K) = E_{elec} \times K$$

In these equations,  $d$  is a constant value which relates to the distance between two nodes and the parameters below are the constant values which are defined previously and they are equal to:

$$\epsilon_{amp} = 100 \text{ pJ/bit/m}^2 \quad E_{elec} = 50 \text{ nJ/bit}$$

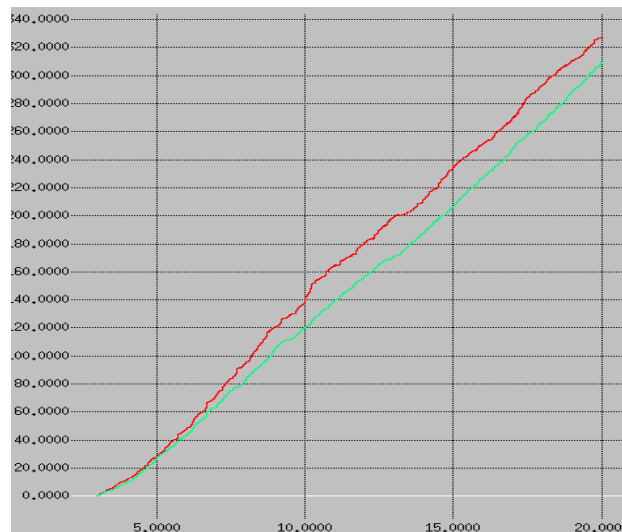


Fig. (a) Throughput in kbps

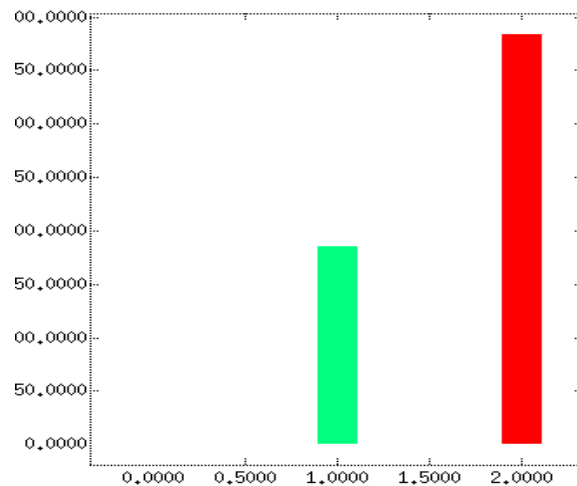


Fig. (b) Transmission range

## VI. Conclusion

In this paper, 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. The simulation results demonstrate that our proposed data-gathering scheme can greatly reduce the moving length. The following issue can be solved in future work when using multiple M-Collectors: the computation is more complex which leads to multiple execution of algorithm. When using multiple M-Collectors in smaller area the cost will be high and efficiency will be low.

## References

- [1] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," in Proc. IEEE Workshop Sens. Netw. Protocols Appl., 2003.
- [2] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in Proc. IEEE RTSS, Dec. 2004, pp. 296–305.
- [3] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, Exploiting mobility for energy efficient data collection in wireless sensor networks. Norwell, MA: Kluwer, 2005.
- [4] D. Jea, A. A. Somasundara, and M. B. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," in Proc. IEEE/ACM Int. Conf. DCOSS, Jun. 2005, pp. 244–257.
- [5] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in Proc. 22nd IEEE Intern. Parallel Distrib. Symp., Miami, FL, Apr. 2008, pp. 1–9.
- [6] I. Slama, B. Jouaber, and D. Zeglache, "Multiple mobile sinks deployment for energy efficiency in large scale wireless sensor networks," e-Bus. Telecommun., Commun. Comput. Inf. Sci., vol. 48, no. 5, pp. 412–427, 2009.
- [7] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in Proc. HICSS, Maui, HI, Jan. 2000, pp. 1–10.
- [8] X. Liu, J. Cao, S. Lai, C. Yang, H. Wu, and Y. Xu, "Energy efficient clustering for WSN-based structural health monitoring," in Proc. IEEE INFOCOM, Apr. 2011, pp. 2768–2776.
- [9] X. Xu, J. Luo, and Q. Zhang, "Delay tolerant event collection in sensor networks with mobile sink," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.
- [10] L. Guo, R. Beyah, and Y. Li, "SMITE: A stochastic compressive data collection protocol for mobile wireless sensor networks," in Proc. IEEE INFOCOM, Apr. 2011, pp. 1611–1619.
- [11] X. Liu, J. Cao, S. Lai, C. Yang, H. Wu, and Y. Xu, "Energy efficient clustering for WSN-based structural health monitoring," in Proc. IEEE INFOCOM, Apr. 2011, pp. 2768–2776.
- [12] M. Zhao and Y. Yang, "Bounded relay hop mobile data gathering in wireless sensor networks," IEEE Trans. Comput., vol. 61, no. 2, pp. 265–277, Feb. 2012.
- [13] Z. Zhang, M. Ma, and Y. Yang, "Energy efficient multi-hop polling in clusters of two-layered heterogeneous sensor networks," IEEE Trans. Comput., vol. 57, no. 2, pp. 231–245, Feb. 2008.
- [14] G. Zhou, T. He, J. Stankovic, and T. Abdelzaher, "RID: Radio interference detection in wireless sensor networks," in Proc. IEEE INFOCOM, 2005, pp. 891–901.
- [15] G. Zhou, T. He, and J. Stankovic, "Impact of radio irregularity on wireless sensor networks," in Proc. 2nd Int. Conf. MobiSys, Jun. 2004, pp. 125–138.