

In Wireless Sensor Networks Efficient Data Aggregation With Multiple Sinks

1Dr G Sanjay Gandhi, 2Popuri Vyshnavi, 3Rajaneeni Manasa, 4Nuthakki Vinuthna, 5Manukonda Sai Satwika
1Professor, 2Student, 3Student, 4Student, 5Student
Vasireddy Venkatadri Institute of Technology

Abstract - A Wireless Sensor Network (WSN) is a network comprised of sensor nodes and a designated device called a sink to which nodes transmit their sensed data. Existing efficient data aggregation protocols developed for WSNs with a single sink may not show the similar efficiency in WSNs with multiple sinks. Usually WSNs are deployed with a single sink and left unattended for a long time, and the sink may go down or crash due to energy depletion, harsh environment, or for some other reasons. If the sink crashes, then the network becomes useless as the sink will not be able to collect data from nodes. One way of making the network more reliable is to deploy networks with more than one sink. Here, we first define the data aggregation problem in WSNs with multiple sinks, and then propose two data aggregation algorithms. The first one is based on a Minimum Spanning Tree algorithm and the second one is based on a Shortest Path Tree algorithm. Simulation results show the effectiveness of the proposed algorithms.

keywords - Wireless Sensor Networks, Data aggregation, Multiple sinks, Routing, Scheduling.

I. INTRODUCTION

A wireless sensor network (WSN) consists of a number of resource-constrained sensor nodes with RF transceivers and thus can communicate with each other with a wireless connection. These nodes sense the environment for events and subsequently forward the sensed data towards a dedicated device called a sink. A wireless sensor network can consist of different types of sensor nodes and can be used in many different applications, including industrial, environmental, military, and health-care [1].

Once deployed in the working field, WSNs may be left inactive for a long time. During this time some nodes and links can go down. Furthermore, the sink and the links between the sink and nodes can experience failures. For example, in deployments [2]–[5], the sinks were not accessible over time due to different reasons. In [2], 4 of the 7 nodes had communication failure with the sink over time, furthermore, the authors observed a sink outage due to power failure. In [3], due to the crash of the database running on the sink resulted in the complete loss of data. Likewise, in [4], two weeks of data were lost due to a sink outage. In [5], the sink went down due to harsh weather conditions.

Although node crashes can be tolerated, the loss of a single sink cannot be handled unless there exist multiple sinks in the network. As mentioned above, the loss of the single sink results in the loss of the network. One way to tolerate single sink crashes is to have WSNs deployed with several sinks. A number of protocols developed for WSNs with multiple sinks exist in the literature. If some of them have been developed to make the network energy efficient, some have been developed to make the network more reliable. However, to our knowledge, the data aggregation protocols that are developed for the WSNs with multiple sinks aggregate data to only one of the sinks. Data aggregation to all sinks makes the network more reliable. For instance, if one of the sinks goes down, end-users can continue to receive the data from other sinks. Another example can be a WSN application where the adversary tries to attack one of the sinks with the purpose of stealing real data and sending fake data to the end-user. If there exist multiple sinks, then inconsistency of data can easily be identified as the data received from the sinks will be different.

Our goal is to develop a data aggregation algorithm that aggregates data packets from all nodes to all sinks in the network. To this end, in this paper, we make the following contributions: (i) We formalize the data aggregation problem in WSNs with multiple sinks. (ii) We provide two algorithms that solve the data aggregation problem with minimal number of message transmissions. (iii) We simulate our algorithms and show the effectiveness.

The paper is organized as follows: In Section II we review the related work. Section III represents the network model and defines the problem of data aggregation in WSNs with multiple sinks. Section IV explains two algorithms that solve the defined problem and Section V demonstrates the simulation setup and results. Finally, Section VI concludes the paper.

II. RELATED WORKS

A number of data aggregation protocols [8], [9] have been proposed for WSNs with a single sink. Also there exist several other protocols that have been designed for WSNs with multiple sinks [10]–[16]. We can divide these protocols into two categories: those which consider only the routing level and those which consider the MAC layer as well.

The idea behind the data collection protocol proposed in [13] is to decrease the number of redundant transmissions by using the neighbors' information. The algorithm proposed in [14] constructs two node-disjoint paths from every node to two separate sinks. If any of the paths fails, the other path is used to deliver data to a sink. A routing protocol that is based on hexagon-based architecture is proposed in [15], where nodes in the network is grouped into hexagons according to their locations. The protocol showed a good data delivery ratio and a good data delivery latency, which are very essential requirements for WSNs used in emergency applications. Another routing protocol proposed in [11] uses distributed ant colony algorithm. In this protocol, different trees rooted at different sinks are used to forward data. All the above works consider only the routing level.

The protocols presented in [10], [12], [16] consider also the MAC layer. The authors of [12] proposed a protocol that forms shortest path trees rooted at each sink in a multiple-sink WSN, and then based on these trees, they proposed a scheduling protocol to support TDMA (Time Division Multiple Access) communication which solves the problem of collision. When combined, these two protocols allow nodes to send their data to their nearest sink without collision. The authors of [10], propose two different algorithms for forming forests and scheduling data aggregation in multiple- sink WSNs. However, in both algorithms different portions of sensor nodes send their data to a single sink. The idea of [16] is that the nodes send their data to the sink that has the highest weight, and weights are calculated according to the distance between nodes and sinks. In other words, each sink receives the data from the closest nodes, which is similar to the idea of [10]. Although these protocols are designed for WSNs with multiple sinks, in the protocols different nodes send data to a different sink, not to all sinks, in other words, the protocols perform a many-to-one communication.

Recently, in [6], the authors focused on solving the problem of data aggregation in a WSN with two sinks and proposed an algorithm. Their proposed algorithm performs data aggregation from all nodes to two sinks, i.e., in a network where $|S|= 2$. The idea of their algorithm is to develop a backbone that connects two sinks and then aggregate data packets at the nodes on this backbone. This idea of building a backbone can also be used in WSNs with multiple sinks. In [7], the authors showed that, in a network with more than two sinks, the minimum number of nodes that transmit more than once can be obtained by building a minimum Steiner tree [17] that connects all sinks. As minimum Steiner tree is known to be NP-complete [18], it is not applicable as a feasible solution for the problem we are addressing.

III. NETWORK MODEL AND PROBLEM STATEMENT

In this section, we present the network model we assume in the paper and state the problem we address.

A. Network Model

We consider a network to be a graph $G = (V, E)$, where $V = V \cup S$ is the union of V , the set of nodes, and S , the set of sinks, with $|S| \geq 2$. E is the set of links between the nodes. Two nodes n and m can hear one another if $(n, m) \in E$, i.e., n and m are in the communication range of each other. For the sake of simplicity, we assume that all nodes have the same transmission range. A path $p = n_1, n_2, \dots, n_k, S_j$ is a set of nodes that deliver a packet of the node n_1 to the sink S_j . The nodes and sinks in the network are randomly placed and they are not mobile and they are static. We also assume that a node is able to aggregate collected data packets from some of its neighbours. We say that data packets are aggregated at node n , if n generates a single data packet from all received data packets. For example, n can calculate the average of all received values from its neighbours and aggregate that in one packet. Further, we assume that every node in the network senses the environment and generates one data packet in every round, where a round is defined as the process of aggregating all data packets at each sink in the network.

As mentioned in Section II, a number of data aggregation protocols have been proposed for WSNs with a single sink as well as for the WSNs with multiple sinks. However, in those protocols, sensor nodes send data to only one of the sinks. As that sink may go down, to increase the reliability, the same data should be sent at least to several sinks. Therefore, we need a protocol that will address the problem of data aggregation where many nodes send to many sinks. The next section states this problem.

B. Problem Statement

The problem we want to solve is the following: Every node generates a data packet that should be delivered to each sink. Data packets are aggregated at nodes on the path. To send to all sinks in the network, a node may need to send the same packet $|S|$ times in one round. That means that there are at least $|V| \times |S|$ packet transmissions in one round. The question is, how much can we reduce the number of packet transmissions in the network per round?

A data aggregation schedule in a network with multiple sinks, where all sinks get aggregated data per round, is a sequence of set of senders N_1, N_2, \dots, N_i satisfying the following conditions:

$$1) \bigcup_{i=1}^n N_i = V, 2) \forall n \in N_i, \forall s_k \in S, \exists m \in N_j \forall m = s_k, i < j:$$

$(n, m, \dots, s_k \text{ is a path}) \wedge (\bigcup s_k = S)$. The first condition states that every node (other than sinks) transmits at least once. The second condition states that every node's data packet is aggregated at each sink.

Data aggregation towards the super virtual sink (b) Aggregated data packet is sent from the super virtual sinks.

Figure 1: Data aggregation process to all sinks. Pentagon is the super virtual sink; rectangles are virtual sinks; circles are other nodes; arrows show the direction of data flow.

To make a data aggregation schedule collision-free, we need to add a third condition as follows:

$$3) \forall n, m \in N_i : n = m \Rightarrow \neg 2HopN(m, n).$$

Here, $2HopN(m, n)$ means that m and n are two hop neighbours. So, the third condition states that if two nodes m and n transmit at the same time, then m and n are not two hop neighbours. Formally, $G = (V, E)$, state our where problem V as follows: Given $V = V \cup S$, with $|V|$ nodes, $|S| \geq 2$ sinks, and the number of packets sent, M_n , and forwarded, F_n , by a node n in one round. The goal is to find a data aggregation schedule that minimizes the number of packet transmissions sent by one node, $M_n + F_n$ in one round, and thereby minimize the total number of packet minimize our transmissions $\sum |V|$ solutions $n=1$ to in the network in one round, that is to $M_n + F_n$. The following section presents this problem.

IV. PROPOSED SOLUTION

In this section, we present our algorithms, where we use the following terminology:

- Backbone - a tree that connects all sinks;
- Virtual sink - a node in the backbone;
- Super virtual sink - a virtual sink that is chosen to collect all aggregated data packets from other virtual sinks; As mentioned in Section III, the minimum number of packet transmissions can be obtained by building a minimum Steiner tree. As it is NP-

complete, we use two different algorithms to solve the above problem: i) an algorithm that uses a Shortest Path Tree (SPT) algorithm (e.g. Dijkstra's algorithm [19]) and ii) an algorithm that uses Minimum Spanning Tree (MST) algorithm (e.g. Prim's algorithm [19]). In both algorithms we use the edge count as weight. Although these algorithms may not give optimal results, we believe that each of these algorithms minimizes the number of nodes on the backbone and thus minimizes the number of packet transmissions. Pseudocodes for these algorithms are given in Algorithms 1 and 2.

In both algorithms, there exist two phases: i) a building a backbone and trees phase and ii) a scheduling phase. The idea of the algorithms is explained through the example illustrated in Figure 1. First, algorithms build a backbone which connects all sinks in the network using a SPT or MST algorithm (the tree with thick lines), then build trees rooted at each virtual sink (rectangles) on this backbone using a SPT algorithm.

After building a backbone and trees, we start the second phase, where we apply a scheduling algorithm to assign *two* time slots to each virtual sink and *one-time* slot to each of other nodes. The scheduling algorithm presented in [6] can be adopted for this purpose. Nodes are assigned only one slot as they need to send only once towards its virtual sink. Virtual sinks use one of their two slots for sending a data packet to the super virtual sink, and the other slot for sending a data packet to a sink. Figures 1a and 1b illustrate this. The data aggregation process starts from the leaf nodes. The leaf nodes send their data packets to their parent, the parent aggregates data packets and send it to its parent and so on until the packets reach a virtual sink. A virtual sink

(a) Data aggregation towards the super virtual sink.

aggregates all packets and then sends that to the super virtual sink. The super virtual sink sends the final aggregated packet to all sinks through the backbone. This way these algorithms make virtual sinks send twice and other nodes only once.

Algorithm 1 SPT Approach

- 1: Choose a node as super virtual sink (*svs*) randomly from the central area/any area.
- 2: Build a SPT using Dijkstra's algorithm rooted at *svs*.
- 3: Build a backbone by choosing paths that connect *svs* to the sinks.
- 4: Build a SPT rooted at each node on the backbone, i.e., at each virtual sink (*vs*).

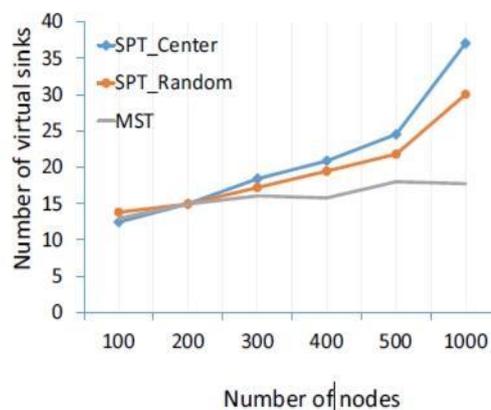
Algorithm 2 MST Approach

- 1: Choose one node as super virtual sink (*svs*).
- 2: Build a MST starting from *svs* using Prim's algorithm.
- 3: Build a backbone by choosing paths that connect *svs* to all sinks.
- 4: Build SPTs rooted at each *vs* on the backbone to other nodes.

V. SIMULATION SETUP AND RESULTS

We have developed a simulation tool in Java to evaluate the performance of our two algorithms. We have evaluated them in networks of sizes 100, 200, 300, 400, 500, and 1000 nodes. The nodes were randomly deployed in an area of size 100 m x 100 m. We placed 4 sinks randomly at the corners of the area. The communication range was set to 25 meters. We repeated each experiment 100 times and computed the average. We simulated MST approach and SPT approach (Algorithms 1 and 2). To see an effect of the location of the super virtual sink, we simulated SPT approach in two ways: i) the super virtual sink is selected randomly and ii) the super virtual sink is selected from the nodes around the centre of the area. Figure 3 shows built backbones by our algorithms. As only virtual sinks (VS) transmit twice per round, the total number of packet transmissions per round is the sum of two times the number of virtual sinks and the number of other nodes, i.e., the number of packet transmission is equal to $|V| + |V S|$. So, the metric we have focused on was the number of virtual sinks on the backbone.

Figure 2 shows that both algorithms show good results. For example, the number of packet transmissions per round in a network with 500 nodes is between 515 and 525. The figure also shows that the number of virtual sinks formed by MST approach is smaller compared to that of SPT approach, and increases slightly as the network size increases. This is because as the area is constant and the number of nodes in the network increases, the number of hops between the super virtual sink and the sinks is the same. In other words, as the network becomes denser, the hop number between the super virtual sinks and a sink does not increase.



REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks: a survey. Computer Networks, 38, 4, pp. 393-422, 2002.

- [2] P. Paritosh, M. Kirk, R. Alistair, H. Ong, and J. Hart, Glacial environment monitoring using sensor networks, In Real-World Wireless Sensor Networks, 2005.
- [3] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, Wireless sensor networks, Analysis of Wireless Sensor Networks for Habitat Monitoring, pp. 399-423, Kluwer Academic Publishers, 2004.
- [4] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, A macroscope in the redwoods, In Proceedings of the 3rd international conference on Embedded networked sensor systems, SenSys 05, pp. 51-63, ACM, 2005.
- [5] R. Szewczyk, J. Polastre, A. M. Mainwaring, and D.E. Culler, Lessons from a sensor network expedition, In EWSN, pp. 307-322, 2004.
- [6] S. Saginbekov and A. Jhumka, Many-to-Many Data Aggregation Scheduling in Wireless Sensor Networks with Two Sinks, Computer Networks, vol. 123, pp. 184-199, 2017.
- [7] S. Saginbekov, A. Jhumka, Y. Mademikhanov, A Lower Bound on the Number of Nodes with Multiple Slots in Wireless Sensor Networks with Multiple Sinks, Sensornets 2017, pp. 202-206, 2017.
- [8] K. Akkaya, M. Demirbas, and R. S. Aygun, The impact of data aggregation on the performance of wireless sensor networks, Wirel. Commun. Mob. Comput., 8(2):171-193, February 2008.
- [9] B. Krishnamachari, D. Estrin, and S. B. Wicker, The impact of data aggregation in wireless sensor networks, In Proceedings of the 22nd International Conference on Distributed Computing Systems, pp. 575-578, 2002.
- [10] Y. Bo and J. Li. Minimum-time aggregation scheduling in multi-sink sensor networks. In SECON, pp. 422-430. IEEE, 2011.
- [11] H. Zhou, D. Qing, X. Zhang, H. Yuan, C. Xu, A Multiple-Dimensional Tree Routing Protocol for Multisink Wireless Sensor Networks Based on Ant Colony Optimization. International Journal of Distributed Sensor Networks, 2012.
- [12] K. Ryouhei and M. Toshiaki, Distributed data aggregation in multisink sensor networks using a graph coloring algorithm, AINA, pp. 934-940, 2008.
- [13] L. Mottola and G. Pietro Picco. Muster: Adaptive energy-aware multisink routing in wireless sensor networks. IEEE Trans. Mob. Comput., 10(12):1694-1709, 2011.
- [14] P. Thulasiraman, S. Ramasubramanian, M. Krunz. Disjoint multipath routing to two distinct drains in a multi-drain sensor network, In INFOCOM, pp. 643-651. IEEE, 2007.
- [15] A. Tuysuz Erman and P. Havinga. Data dissemination of emergency messages in mobile multi-sink wireless sensor networks. In Med-Hoc-Net 2010, pp. 1-8, 2010.
- [16] A. Das and D. Dutta. Data acquisition in multiple-sink sensor networks. SIGMOBILE Mob. Comput. Commun. Rev., 9(3):82-85, July 2005.
- [17] E. N. Gilbert and H. O. Pollak. Steiner minimal tree. SIAM Journal on Applied Mathematics, 16:1-29, 1968.
- [18] R. M. Karp. Reducibility among combinatorial problems. In Proceedings of a symposium on the Complexity of Computer Computations, pp. 85-103, 1972.
- [19] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, Introduction to Algorithms (2nd ed.). McGraw-Hill Higher Education, 2001.