TO SHIFT OR NOT TO SHIFT: MAXIMIZING THE EFFICIENCY OF A WIRELESS SENSOR NETWORK

Chabli Boler, Sumanth Yenduri, Wei Ding* Louise Perkins, John Harris
University of Southern Mississippi
730 East Beach Blvd
Long Beach, MS 39560

*New York Institute of Technology
Dept. of Computer Science
1855 Broadway
New York, NY 10023

1. ABSTRACT

In this paper, we address the efficient utilization of a wireless sensor network (WSN) to improve the overall data throughput and to cut down the cost of running it. Sensor nodes closest to the sink node expend more energy as they send more routing messages, which decreases their life and introduces dead spots. Having dead spots will lead to the eventual isolation of the sink nodes from the network. We attempt to solve this problem by distributing multiple sink nodes. This will enable us to utilize the network at its maximal capacity as the sensor nodes start to die off. By adding multiple sink nodes, we will switch between them and choose the one that will yield optimal network coverage after the initial sink node is no longer viable to service or cannot service the network optimally. However, at any point of time, there will be only one active sink node. We will explore in three ways. First, we use a single sink node (with no switching) until it is isolated from the network. Second, we use a combination of highest percentage of network coverage combined with the highest areas of node density on the network to derive the optimal use of the network. We call this the Cluster Algorithm for Sink Selection (CASS). Finally, we try to keep the network optimal by switching to the sink node that is farthest from the current sink (assuming to gain the maximum network reach). We will analyze and discuss our findings with respect to the data through put and life of the network.

2. INTRODUCTION

In this paper, we will explore ways to improve data throughput using multiple sinks in a wsn. The goal is to get closer to the maximal lifetime of the sensor node by allowing the network to function as it would in a single node network. Once a certain percentage of the network dies (inaccessible), we switch to a new sink. This will let us get closer to the maximal lifetime of the network by using under utilized sensor nodes. At the same time, it will make the role of previously frequently used nodes less and thus, extends their lifetime.

A wsn is used to gather environmental data. WSNs are mostly deployed in areas, which have no infrastructure at all. A typical way of deploying the sensor nodes is for them to be dropped from an airplane [1]. Once these networks are deployed, they are for the most part self-sustaining. A WSN in comprised of a main node, which is called the *sink node*, and many low cost sensor nodes that will route data back to the sink node. The sink node of the network has many functions. It usually has an infinite power source. It has a connection to some sort of network other that the sensor network. This network can be a privately built secure network for an organization or it can be the Internet. This allows the organization to monitor the sink in order to send various tasks, view any data gathered, and also check the status of the network. The sink contacts the sensor nodes via a wireless medium such as radio, infrared, or optical media [2]. The sensor nodes in the network preform basic functions since they are on a finite power source. Because of finite power supply, the radio range is limited. Once a sensor gathers data, it sends this data back to the sink to be processed. Since the sensor has limited radio range, if it is not in the vicinity of the sink, it has to send its data to another sensor node within its range to relay the information to the sink. The process continues until the data reaches the sink. The sink is initially responsible for setting up the network in such a way that the nodes know the path to send data back to it. The sink does this by flooding the network with a setup message before the first message is passed. This makes a WSN self-sustaining. If there are any new additions or deletions to the WSN, it will send out a message to all nodes closest (to where the addition or deletion is made) and the network will adapt. The network will compensate by rerouting paths when sensor nodes die so other nodes dont lose contact with the sink. As wsn's are adaptable and dynamic, they are ideal for data gathering from remote/hostile locations. They are numerous real world applications of wsns [3, 4]. In the next section, we discuss related research. Next, we will provide an overview of the overall network setup and then, describe the algorithms. Later, we will discuss and compare the results. Finally, we describe our plans for future work and present our conclusions.

3. RELEVANT RESEARCH

Adding multiple sinks to a sensor network is not a new idea. This concept has been demonstrated via various research studies [5,6,7]. In Chen et al. [5], the authors deploy multiple sinks in a WSN to give a network a means of sending data to any sink at any time with all sinks being active on the network. Their method is based on Breadth First Search to use the sensor nodes in a consistent manner. Using this method the authors can distribute the data flow of the network. If you distribute the flow then you don't use the same nodes as often in the network and hence extend their life. In Das and Dutta [6], the authors use multiple sink nodes to find the shortest path for each node to the closet sink in the network. Using this, they conserve energy to send more data. This method also provides an estimated energy savings analysis. Like our own method, they consider a random network placement model.

In Vincze et al. [7], the authors study energy savings in a WSN by moving multiple sinks. They suggest deploying the sink nodes in a coordinated manner to analyze them, and then decide the optimal location to place them. The sinks are placed such that the sensor nodes can reach them in a limited amount of hops. This allows the sinks to last longer. As they decide the initial placement of the sinks, they also allow for later movement of the sinks to re-optimize the network flow. All the above methods use the same principle of shortest path to the closest sink. They all focus on energy savings. The problem with these methods is that they must use some sort of localization in order to figure out the approximate shortest path is.

Our goal is to maximize the lifetime of the sensor node while getting valuable data from the network. We cut down any additional expense to reposition a sink node when a dead spot occurs, which is the case in single sink networks. Other methodologies that have multiple active sinks have extra cost associated in maintaining them. Sinks are nodes with infinite power source. Our method maintains only one active sink node at all times, rendering the network to be less expensive.

4. SET UP

Since our goal is to get the maximal usage and efficiency from the network, we use multiple sinks to extend the life of the network as a whole by trying to reach the maximal life span of each sensor node. However, there will be only one sink active at any time in the wsn. The maximal lifetime that we wish to reach will be a finite time between $0 < S_r < S_m$ where S_r is the runtime of a sensor node and S_m is the maximum time a sensor node can live with its basic operations. The closer we can get to S_m while getting valuable data from the network, the longer the network will live. While it is probably not feasible to get to S_m, as that would mean the node is not alive, the closer we get to this, the better. We will try to achieve this by allowing the sink to function as if it were on a single sink network, and then shift it when a certain percentage of the network has been lost due to interference or energy depletion. Our network will be set up to be completely random. The random positions will apply both for sensor nodes and the sink nodes as well. This will help us generalize the setup so that it can be adapted for all scenarios. By using multiple sinks it will increase the message throughput of the entire network. One of the demerits with a single sink is that you can get completely cut off from a portion of the network. With this, perfectly good nodes cannot send data to the sink. Instead of losing data, we simply switch sinks so that the lost portion of the network can have a chance to send this data. The sink shift will happen once the network hits a desired loss rate. Once that rate is reached, the current sink will then contact the sinks that still have nodes around them and use our customized heuristic for the purpose of selecting the next sink that will increase our throughput. This heuristic is called CASS (Cluster Algorithm for Sink Selection). CASS was created to maximize the throughput by selecting the sink that has the largest network reach. CASS will be described later. The rest of the nodes that were connected to the previous sink will do one of two things: if they can be reached, they will be rerouted to the new sink, else, with their remaining energy, they will enter a state of listening and wait to be contacted for use by another sink, till they approach their maximum lifetime. For control purposes we will compare CASS against a heuristic that uses the farthest sink. We will also compare the throughput and overall lifetime of both methods to a single sink non-shifting WSN.

5. SIMULATION

Network Setup: The simulation was run on the Microsoft Windows operating system. The simulation was coded in C++ using Microsoft Visual Studio 2010. The network setup for the simulation will have *n* sinks with *k* sensors. The transmission media will be simulated using a message passing class that was created to process all nodes at particular time interval, simultaneously. We use the CASS heuristic and the Farthest sink heuristic to determine the next sink to shift. In order to get consistent results, we ran numbers for randomness, only once. With a large list of random numbers ranging from 0-99 (sensor node ID), we were able to make the results of a network setup reproducible by removing the randomness that is associated with nodes sending data to the sink. Using this method, we compare two heuristics for the same network setup with the same depletion pattern as a single sink network up until the point where the sink shift occurs in the multi sink network.

Sink Node Setup: The sink node will be the main focal point of the network. It is responsible for gathering the data. It does so by establishing a network in which the sensor nodes can create routes and links to each other in order to get the data back to the sink. Initially, the sink floods the network with a setup message. This message will be passed to every sensor node in the network. After receiving the message, each node contacts the sink and specifies the path the message took to reach it. The sink node can now establish a "Tree" in which it is the root node. It also knows how to contact every other node in the network that is in range of at least one other node who can reach the sink (Fig 1).

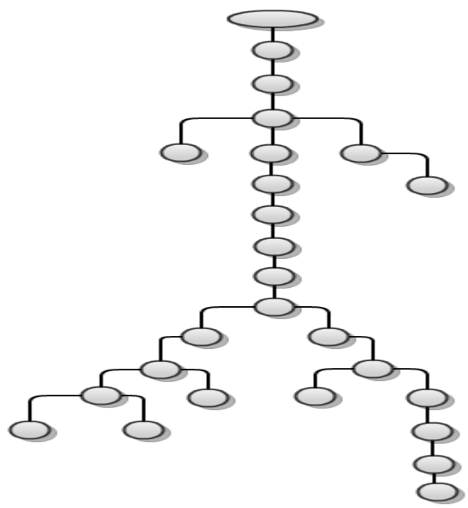

Figure 1: Sink Node Network View.

And then, the sink gathers the data to do any appropriate calculations or for sending the data to any other network. Before a sensor node dies, it notifies the sink so the network tree is updated to remove any branches that may no longer be valid. If another active node is chosen from the dying sensor's neighbors, then the new sensor sends its id as the replacement, so the sink can reinstate the previously lost branches and update the tree accordingly. If a certain percentage (preset threshold) of the network is lost and it does not correct itself in a certain amount of network cycles, then the sink node tests to see if it meets the shift sink network life ratio (reachable nodes/ starting nodes). If it does not, it keeps on going until this ratio is met at which point, the sink implements CASS. Once CASS runs, the sink either shifts or stays the same depending on the maximum network reach criteria.

Sensor Node Setup: We have based the sensor nodes off of the IRIS mote built by MEMSIC. It is a low power 2.4 GHZ module that has a 500 meter line of sight range without any amplification [8]. The sensor nodes will start by being placed randomly on a 100 x 100 grid. This will simulate them being thrown from a plane. Once these nodes hit the ground they will start sending hello messages to their neighbors. This will allow them to know their neighbors. Some time later, the sink will send them a setup message at which point they will see where the message came from and the path it took to get back there. After a path to the sink has been established then the sensor nodes will be able to send any data they collect during their lifetime. The sensor nodes will have a 5% chance of sending data, each network cycle (or whatever the time period specified). The sensor nodes will also function as message passers since the range of the radio for each sensor is limited. In our simulations, we use 15 units. The message passing of the sensor nodes will be based upon the same tree that is setup by the sink. Each sensor node only knows about its parent and its children. It also knows suitable backups, in case of failure.

Data Transfer (Routing): Since the simulation has to pass messages from one node to another, we will simulate a wireless medium. We will do this by processing each sink and sensor node. After all nodes have had one run cycle, the data transfer system will visit each node and retrieve the message and send it to the appropriate destination. We represent one iteration of the network, each time this process is completed. The process repeats until there is no more communication on the network.

6. CASS (CLUSTER ALGORITHM FOR SINK SELECTION)

CASS was created in order to choose the sink node that will give you the maximum amount of exposure to the network. CASS works based upon clusters of sensor nodes. A cluster by our definition contains at least three sensor nodes. The clusters will be determined using the tree view (Fig 1) which is generated during the sink setup phase. CASS will traverse down the tree and make note of all parents that have at least two children (Fig 2). Next, the algorithm will combine any clusters whose parent is a child of another cluster.

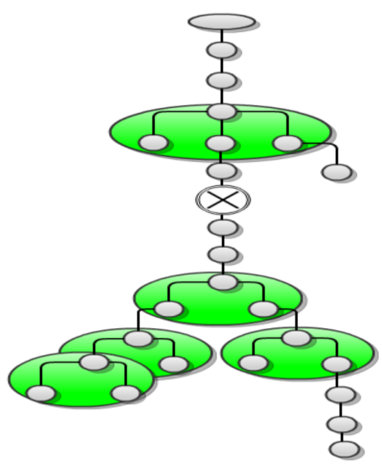

Figure 2: Stage 1 CASS

This will merge smaller clusters into larger ones (Fig 3). Then the algorithm will look to see if any cluster has sufficient access to another cluster. We will define sufficient access by saying they have at least two pathways or links to reach another cluster (Fig 4).

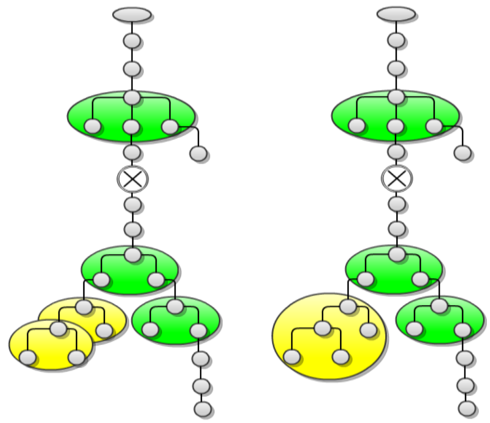

Figure 3: Stage 2 CASS

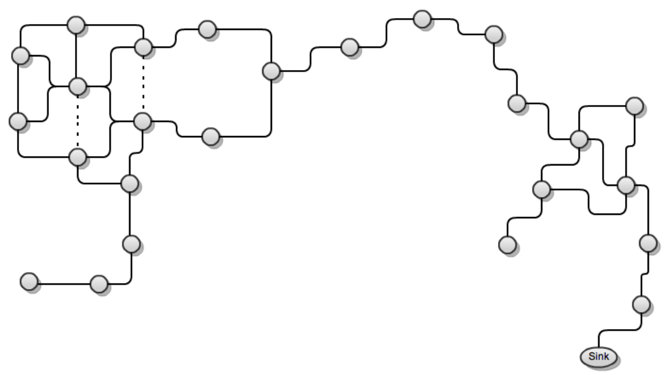

Figure 4: Top View of CASS Example Network (Dashed lines Denote pathways).

We define a pathway by saying that if one cluster has a sensor that contains a child, neighbor, or parent that is a part of the comparing branch in the tree then that is one qualifying pathway. If this is true and you do have two pathways then those two clusters will be combined into one cluster (Fig 5).

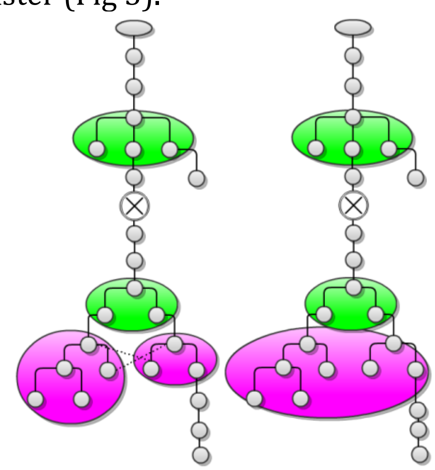

Figure 5: Stage 3 CASS (Dashed Lines Denote Pathways)

Once the clusters have been established, we then calculate the reach of the cluster. This is done by taking the size of a cluster and assigning it a weight. This weight will allow the sink selection to be more directed at bigger clusters. The reason we select bigger clusters is because they will have sufficient re-routing ability and will allow for less sink shifts in the longer run. The algorithm then calculates the number of children a sink can reach. This value is assigned a smaller weight. The theory here is that the more children you have, the more parts of the network you can potentially access without shifting to a new sink. Finally the algorithm calculates the nodes that do not belong to any cluster, but are reachable by the sink. There are accounted as single weight point in the reach total. Once this calculation is finished the sink that has the highest reach total will be the next active sink of the wsn.

Network Reach:
I = current sink
CR(I) = # Nodes contained in clusters reachable by I
CN(I) = # Child nodes of I
SN(I) = # of single nodes (not in a cluster) reachable by I
Formula:
*Network Reach = CR(I) * 1.33 + CN(I) * 1.2 + SN(I) (arbitrary weights)*

7. ANALYSIS OF RESULTS

The simulation was run in three different ways. The first time, we ran the simulation was to establish a base. This base is set for the lifetime and all the messages were generated using a single sink network (with no switching). The second time, the simulation was run using multiple sinks and the CASS algorithm for switching the sinks. The last time, the simulation was run using multiple sinks and the Farthest sink method for switching sinks. The initial power level of the sensor nodes in all simulations was set at 100 joules of power. A total of 9 random grids were used for our experiment. Single sink was run 9 times. Each of the two heuristics (CASS and Farthest sink) was run 90 times (9 grids * 10 shift rates * 2 heuristics = 180 simulations). The number of simulations for the 3 methods totalled to 189. The results for a sample grid are shown below:

Table 1: Sensor Node, 100 Joules of Energy Results. CASS (Grid 1).

Network Setup	Data Messages Received	Network Lifetime (Days)
Single Sink WSN	396407	14.3903
45% Shift Rate	542883	14.414
50% Shift Rate	542888	14.414
55% Shift Rate	542981	14.4142
60% Shift Rate	545415	14.4142

65% Shift Rate	545415	14.4142
70% Shift Rate	545415	14.4142
75% Shift Rate	539990	14.4143
80% Shift Rate	532940	14.4143
85% Shift Rate	532940	14.4143
90% Shift Rate	532738	14.4143

As you can see from Table 1, the samples taken at different shift points were all better that the single sink WSN. Our tests show that using multiple sinks in a network out performed the single sink network setup. Out of the two sink selection algorithms for the multi sink networks, the CASS algorithm sent more data messages to be processed by the sink nodes. Table 2 shows the results for farthest sink heuristic for the same network grid. CASS performs better in terms of data throughput.

Table 2: Sensor Node, 100 Joules of Energy Results, Farthest Sink Selection. (Grid 1)

Network Setup	Data Messages Received	Network Lifetime (Days)
Single Sink WSN	396407	14.3903
45% Shift Rate	532329	14.3977
50% Shift Rate	532329	14.3977
55% Shift Rate	533196	14.3977
60% Shift Rate	533601	14.3962
65% Shift Rate	533601	14.3962
70% Shift Rate	533587	14.3962
75% Shift Rate	533334	14.3955
80% Shift Rate	533530	14.3955
85% Shift Rate	532756	14.3971
90% Shift Rate	532861	14.3968

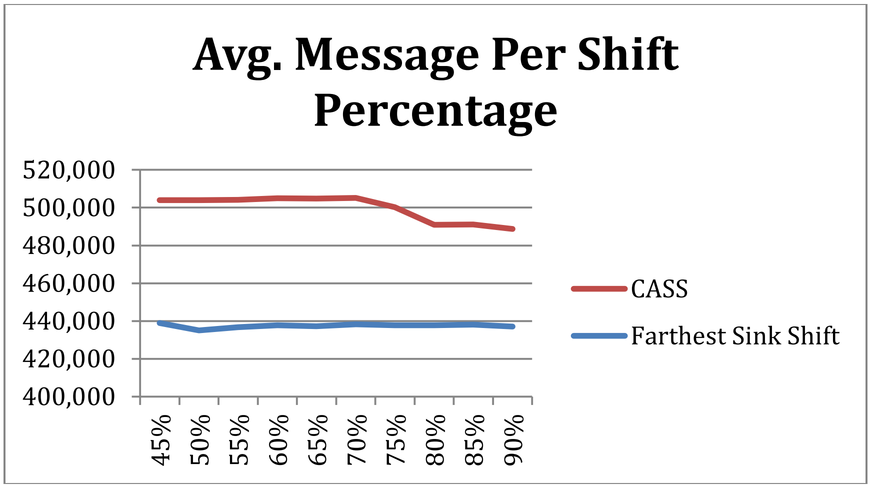

Figure 6. Avg Messages Generated Per Shift Percentage at Maximum Data Throughput

Fig. 7 shows the total average messages generated for all the simulation runs. Both the heuristics performed better than the single sink method. However, CASS method performed marginally better than the Farthest sink.

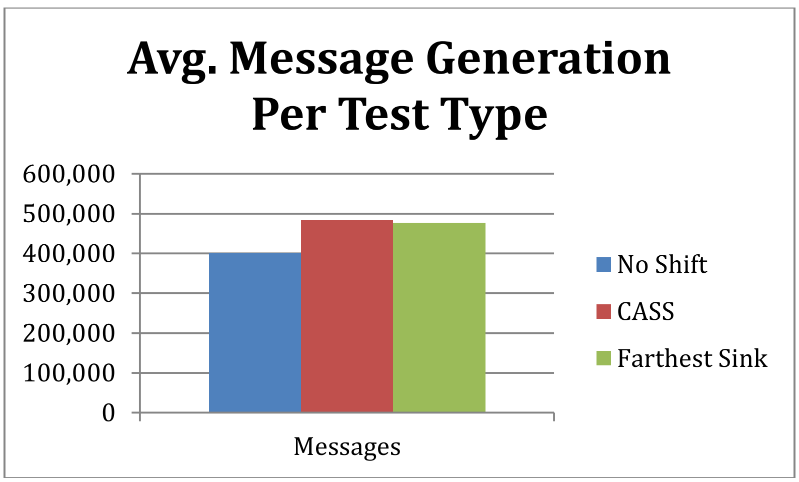

Figure 7. Average Messages Generated for All Tests

8. FUTURE WORK AND CONCLUSIONS

In the future, we would like to explore adding power cycling to the sensor nodes in order to extend the life of the sensor nodes in the network. While this is an efficient means of extending the lifetime of the network it should in theory yield the same amount of data from the network. Another issue will be to find an optimal shift range that will yield consistent results. It will need an analysis of a lot of various random network structures and shift percentages in order to get a general percentage that will be useful to any WSN. This percentage might change with the number of sinks that are in the WSN so it would also be wise to study the optimal sink to sensor node ratios as well. We plan on comparing the CASS algorithm versus the other multiple sink usage algorithms such as Chen et al, Das and Dutta, Vincze et al. [5,6,7]. This will allow us to study the affects of each method and then determine if shifting our focus is warranted in the field of WSNs. Finally, we intend to study via a cost analysis, to figure if the additional network output is justified. While having more sinks is desirable, we would like to know the cost per extra message generated.

From our experimental results, we conclude that multiple sink network using the CASS heuristic or Farthest sink has higher data throughput than single sink with no switching. By using CASS, we can be sure the sinks will be selected in a manner that will always give us the most out of the network in terms of data throughput. Though, we did not see an increase in the life of network, we observed a higher data throughput using the CASS heuristic. Further, since our method has only one active sink node at all times, the cost of the network is much lower.

BIBLIOGRAPHY

[1] Archana Bharathidasan, Vijay Anand, Sai Ponduru. Sensor Networks: An Overview. University of California, Davis.2002

[2] Ian F.Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci. A Survey on Sensor Netowrks. IEEE Communications Magazine. 102-114. Aug. 2002

[3] Joseph Polastre, Robert Szewczk, Alan Mainwaring, David Culler and John Anderson. Analysis of Wireless Sensor Networks for Habitat Monitoring. University of California, Berkley, Intel Research Lab Berkley, College of the Atlantic.2004

[4] Ning Xu. A Survey of Sensor Network Applications. University of Southern California.

[5] Sixia Chen, Mathew Coolbeth, Hieu Dinh, Yoo-Ah Kim, Bing Wang. Data Collection with Multiple Sinks in Wireless Sensor Networks. University of Conneticut.2009

[6] Abhimanyu Das, Debojyoti Dutta. Data Acquisition in Muliple-Sink Sensor Networks. University of Southern California. 2005

[7] Zoltan Vincze, Rolland Vida, Attila Vidacs. Deploying Multiple Sinks in Multi-Hop Wireless Sensor Networks. Budapest University of Technology and Economics.2007

[8] Memsic. IRIS Mote Datasheet. http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=135%3Airis

[9] Luca Mottola and Gian Pietro Picco. MUSTER: Adaptive Energy-Aware Multi-Sink Routing in Wireless Sensor Networks. Mobile Computing, IEEE Transactions.Volume: PP Issue:99. 1–1.2010

[10] Chen Xu, Lei Cao, Wei Shao. Multiple Sink Dynamic Destination Geographic Routing in Wireless Sensor Networks . Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2010 International Conference. Issue Date: 10-12 Oct. 2010. 121-124. 2010

[11] Wint Yi Poe, Jens B. Schmitt. Self-organized Sink Placement in Large-scale Wireless Sensor Networks. Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium. Issue Date: 21-23 Sept. 2009.1-3.2009