

# Upper Bound of Sensor Network Lifetime – A Flow Optimization Approach

Tuli Nivas, Gergely V. Záruba

*The University of Texas at Arlington, Department of Computer Science and Engineering*

*Arlington, Texas 76019*

*txn3232@omega.uta.edu, zaruba@uta.edu*

## Abstract

*The growth in technology, digital electronics and wireless communications has made sensor networks a reality. The low cost, low power but multifunctional sensor nodes communicate with other nodes and need to be energy efficient in order to save battery power. This paper deals with the problem of sensor network lifetime optimization. By setting up data flows among sensor nodes as optimization variables, the framework will set up an optimum network with maximum lifetime. Modified simulated annealing has been used to deal with the flow reorganization problem. We will show lifetime results versus number of active sensors and population using a C++ implementation of our framework and the modified simulated annealing approach. This framework is not a routing protocol and since it does not use any special routing strategies to maximize the lifetime it can be used as a benchmark for WSN routing protocols. It can be used to see how simple routing methods compare with the optimal strategy.*

## 1. Introduction

WSN consist of independent, collaborating nodes that can sense, process, and exchange data as well as act upon the data content. They are miniature in size and combine one or more sensors, memory, processor and wireless radio links for data transmission and reception. Sensor networks are usually deployed randomly in inaccessible or unknown terrains to collect data about what is happening in these environments. Since nodes possess an on board processor, instead of sending all the data they collect to the central processing unit, they can use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. Due to these features sensor networks have found a widespread use in a variety of applications including remote climate and wildlife migration monitoring, medical, intelligence, security data gathering, and battlefield awareness.

The compact form of sensor nodes makes them limited in power, computational capacities and memory. Also since the nodes are likely battery powered they need to conserve energy as replacing batteries in possibly harsh terrains is infeasible. Thus, one of the major challenges when working with sensor networks is to maximize their post deployment lifetime.

This paper concentrates on deriving a framework to calculate an upper bound on the lifetime of a sensor network. Getting a fair idea of the longevity of a network before it is deployed is rather advantageous, because the proper

functioning of the network is dependent on it. Network lifetime is not only affected by the amount of energy consumed at each node, but also on the distance between the nodes, the path chosen to transmit the data and the placement of sensor nodes in the network. The density of the network also plays an important role in the total lifetime of the network and so does the position of the sink, path loss characteristics and the starting initial energy at each node. Hence minimizing energy consumption and maximizing the lifetime is a big task. The knowledge on the upper bound of the lifetime helps in understanding the factors that keep the network from reaching its full potential and can help in the design of better protocols.

The main objective of this paper is to find an upper bound on the network lifetime without relying on complex routing strategies or knowledge of the location of other nodes. It is not an energy efficient routing protocol for WSN. This means that the framework calculates the lifetime for a scenario where all nodes to do what is best for the network. And because of that it can be used as a benchmark to compare the performance of other routing. So existing and future routing, and aggregation approaches may be compared to how close they come to an optimal behavior.

In this paper we consider a sensor network where all nodes are capable of exchanging information/data with other nodes and also conserve energy in an optimal manner. Nodes sense and/or relay information to an information sink in the network. We investigate how long it will take in such a network before the first node is completely depleted of its initial energy (the lifetime of the network).

The rest of the paper is organized as follows: Section 1.1 outlines the most significant related work to our research. Section 2 describes the basic system model that we have considered for a sensor network; it explains the network flow constraints and the way we calculate the amount of energy dispensed with each action at the individual nodes. Section 3 describes how a simulated annealing process can be modified to deal with the flow reorganization to find the flows that result in a maximum lifetime. Section 4 shows lifetime results that we have obtained by creating a C++ optimization model for our framework. Section 5 gives an example of how the framework can be used as a benchmark. Finally, we conclude the paper in Section 6 and provide an outline of our current research direction.

## 1.1. Related Work

Several researchers have dealt with the topic of trying to maximize the post deployment lifetime of sensor networks. Here we are reviewing the most significant papers that relate to this work.

Bhardwaj et al. [3] have worked on modeling the lifetime of sensor networks depending on node placements and data-generation at individual sensor nodes. They have also derived the upper bound on the sensor network lifetime by assigning roles to the nodes in the network [1]. However, they did not consider the complexity of their flow model when providing solutions. There is also work done by Zhang et al. [11] to find an upper bound of a sensor network taking into consideration the K-coverage problem. In contrast the network lifetime derived in this paper considers no energy saving techniques, nor does it consider special cases of networks where a particular number of nodes are needed to achieve the network functionality. It is a more general version of the lifetimes derived in the above stated papers, and hence can be applied to find the upper bounds of almost any algorithm or strategy that is used in a sensor network. Giridhar et al. [13] have considered 2 network configurations and the parameters that affect their functional lifetime. Paschalidis et al. [14] have worked on asymptotical transmission policies while considering various performance parameters like transmission rates, throughput etc.

Topology control is also an important technique to reduce energy consumption and radio inference in a network **Error! Reference source not found.** Energy conserving routing algorithms have been proposed to tackle the problem in ad hoc networks [8]. Both [4] and [5] suggest that routes in an ad hoc network should be selected based on available energy; this in turn can increase lifetime. A number of energy aware MAC and Cluster based protocols (see [2], [7], [6], or [10]) have also been proposed in order to maximize the lifetime of networks. In contrast, our work could be used to compare the lifetime achieved by these protocols to a maximum achievable lifetime and thus be able to tell how close they come to an optimum.

## 2. Lifetime Flow Model

Sensor networks or data gathering networks aim to sense a change in the environment collect the information and then transmit it to the main processing center or sink, in the most energy efficient manner. In the network nodes nodes can be classified as follows [1]:

- *Sensor Node*: in the network is responsible to collect data. It can also act as a relay node in order to transmit data.
- *Relay Node*: is responsible for transmitting data along a chosen path in the network.
- *Aggregator Node*: The aggregator node combines two or more data streams into a single stream of data.

Taking into consideration the fact that the whole process of data sensing and relaying in a network can be viewed as a network flow problem, the following observations can be made. We adopt and modify the flow constraints and transmission reception calculations from [1] as follows:

### 1. Flow and Power Calculations

- The data flow: The flow  $f_{ij}$  between any two nodes  $i$  and  $j$  can be termed as the amount of data that is being transmitted from node  $i$  to node  $j$  during the lifetime of the network. Thus,  $f_{ij}$  is the proportion of the total number of bits transmitted from node  $i$  to node  $j$  divided by the product of the lifetime and the rate of sensing.
- Transmission Power: The power required for node  $i$  to transmit data to node  $j$  is dependent on the distance between the nodes  $d(i,j)$ , the rate of transmission  $r$  [bits/sec] and the path loss index  $n$  along with some positive constants  $\alpha_1$  and  $\alpha_2$ :  

$$P_{tx}(i, j) = (\alpha_1 + \alpha_2 d(i, j)^n) r$$
- Reception Power: Similarly the battery power required for receiving is dependent on the rate of transmission  $r$  and some positive constant  $\alpha_3$ :  $P_{rx} = \alpha_3 r$
- Sensing Power: Likewise, the sensing power is required to sense enough information for a single bit. It is dependent on the rate  $r$  and a positive constant  $\alpha_4$ :  

$$P_{sense} = \alpha_4 r.$$

### 2. Flow Constraints

- Considering the transmission of data in a sensor network as flow between nodes there are some constraints that need to be abided. Let us use the following denotations:  
 $\{N\}$  = set of all nodes in the network, including the sink  
 $R$  = sink  
 $\{S\}$  = set of all nodes that will be sensing in the network  
 $\{NS\} = \{N\} - \{S\} - R$   
 $k$  = number of active sensors at any given point of time,  $k \leq |\{S\}|$ .

The constraints are as follows:

- The amount of flow between nodes cannot be negative  

$$f_{ij} \geq 0 \text{ for } \forall i, j \in \{N\}$$
- The amount of data that flows out of a (non-active sensor) node needs to be equal to the amount of data coming into the node:  

$$\sum_{\forall j(j \neq i)} f_{ji} - \sum_{\forall j(j \neq i)} f_{ij} = 0, \forall i \in \{NS\}$$
- Active sensor nodes cannot consume data and cannot generate more than a unit flow.  

$$1 \geq \sum_{\forall j(j \neq i)} f_{ij} - \sum_{\forall j(j \neq i)} f_{ji} \geq 0, \text{ for } \forall i \in \{S\}$$
- The total amount of flow generated by all sensors is  $k$  (recall  $k$ 's definition):

$$\sum_{\forall i \in \{S\}} \left( \sum_{\forall j(j \neq i)} f_{ij} - \sum_{\forall j(j \neq i)} f_{ji} \right) = k$$

- The energy dispensed at any node is dependent on the amount of power used for transmitting and receiving data. For a sensor node we also have to consider the

amount of power required to sense data. We denote the initial energy of each node by  $e_0$ . Then:

$$e_0 \geq t \left( \sum_{\forall j(j \neq i)} f_{ij} \cdot p_{rx}(i, j) + \sum_{\forall j(j \neq i)} f_{ji} \cdot p_{rx} \right) + t \cdot p_{sense} \sum_{\forall j(j \neq i)} f_{ji} - f_{ij} \quad \text{for } \forall i \in \{N\} - R \quad (1)$$

### 3. Optimizing Flows

The previous section provides the flow constraints and the energy constraint function (in Equation 1). Since, what we are interested in is maximizing the lifetime  $t$ , we would like to find values for all flows  $\{f_{ij}\}$  so that  $t$  is maximized while all constraints are satisfied. Thus, our objective function is:

$$\max t$$

Now, that we have formulated the problem, we need to investigate its complexity. At first sight the problem seems to be a simple linear programming problem [1]. However, we need to note the convoluted dependencies of the optimization variables on each other, as they need to satisfy strict flow constraints. In addition, the objective function becomes non-linear, when we express it in its right form of:

$$t = \max_i \left( \frac{e_0}{\sum_{\forall j(j \neq i)} f_{ij} \cdot p_{rx}(i, j) + \sum_{\forall j(j \neq i)} f_{ji} \cdot p_{rx} + p_{sense} \sum_{\forall j(j \neq i)} f_{ji} - f_{ij}} \right)$$

Clearly, our optimization problem cannot be solved with regular linear programming methods. Thus, we need to devise a heuristic to provide a solution. In this section we outline a simulated annealing based heuristic, where flows will be modified during each iteration such that they do not violate constraints.

In simulated annealing, an artificial initial temperature  $T_0$  is chosen, and reduced throughout the optimization process by a cooling factor. At each temperature a number of iterations are done, where a new random (depending on the temperature) set of optimization variables is chosen, and the objective function is evaluated. (Note that in general all constraints would have to be evaluated as well, however we will make sure that flow variables are appropriately changed.) If the new value of the function is larger (in our case) than the previous value, then the new solution (the new set of optimization variables) is accepted; otherwise the new set of variables may still be accepted based on a random function with the temperature and the ‘‘badness’’ of the new value as parameters.

What we are interested here is the method with which we can randomly modify flows so that all constraints remain intact. When we look at the objective function we notice, that the lifetime of the network is determined by the node that runs out of energy first. We will refer to this node - in each iteration - as the current *bottleneck node*. Thus, the network lifetime will change only if we modify flows involving the bottleneck node. The selection of the initial feasible solution for the annealing process is the trivial solution: all active sensors’ flows to the sink are set to one, while all other flows are set to zero.

#### 3.1. Semi-random Flow Adjustments

During the optimization process there can be two kinds of situations: i) the bottleneck node is a sensor node); ii) the bottleneck node a relay node. Thus, to increase the life of the bottleneck node, we either have to redirect some of its outgoing flow to a physically closer node (case 1); or both incoming and outgoing flow has to be redirected (case 2). Keep in mind that the part of flow being redistributed during these cases is chosen randomly. Also the node selection is done in a random fashion.

**Case 1.** Here the bottleneck node is generating data that we want to redirect. The first step is to randomly pick a node from all the nodes that the bottleneck is transmitting to, i.e., we choose an *outgoing node* at random. Then we check to see if this outgoing node is the sink or just another node:

- 1a. If the outgoing node is the sink, then to save the energy of the bottleneck node, another node needs to be found that is closer to the bottleneck node than the sink is. Thus, we search and choose such a node from this set (intermediate node); then we redirect a part of the outgoing flow from the bottleneck node to this intermediate node. In addition, we modify the flow between the intermediate node and the sink with the same amount to maintain the flow constraints. This situation is depicted in Figure 1.

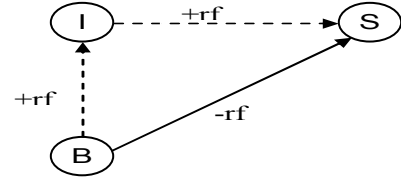


Figure 1: Flow redirection from the sink.

- 1b. If the outgoing node is not the sink node, then we determine whether the sink is closer to the bottleneck node as compared to the outgoing node. If indeed it is closer then the outgoing flow is redirected to the sink directly (there can not be a more energy efficient way); refer to Figure 2. However, if the sink is not closer then we find all nodes that are closer than the outgoing node and select the intermediate node among them. A fraction of the original flow from the bottleneck is then redirected towards the intermediate node; the same amount of flow is added from the intermediate node to the original outgoing node. See Figure 3.

**Case 2.** In this case we would like to relieve the bottleneck node by re-routing some of the flow that goes through it. The first step is to select a node (incoming node) from all the nodes that are providing flow to the bottleneck node. In addition we select a node among all the nodes that are receiving flow from the bottleneck node (outgoing node). Then we pick a third random node (intermediate node) from the remaining nodes in the network. We now pick a fraction of the flow from the incoming node (making sure that this fraction does not exceed the flow to the outgoing node/s). This flow is then redirected from the incoming node through

the intermediate node to the outgoing node as depicted in Figure 4.

Alternatively, we can redirect a fraction of flow from the incoming node to the bottleneck node to the intermediate node which in turn splits this flow up and distributes it randomly among the bottleneck nodes' outgoing nodes. This is depicted in Figure 5.

By redistributing flows in the above fashion the network lifetime may be increased in every iteration throughout the annealing process. Note, that with the above method all possible feasible flows may be generated, the proof of which is omitted here due to space restrictions.

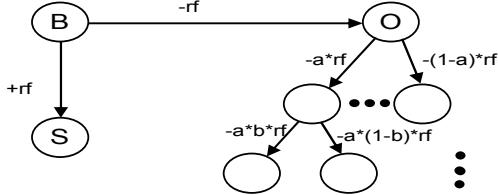


Figure 2: Flow reorganization (S is closer to B)

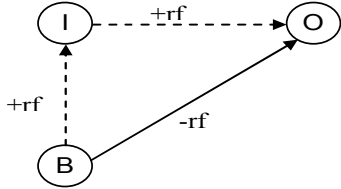


Figure 3: Flow reorganization (with intermediate node)

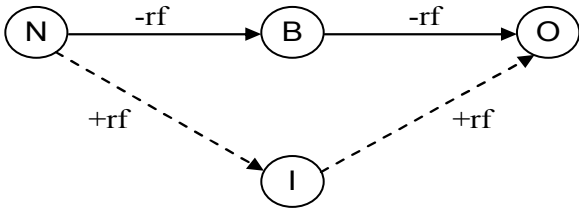


Figure 4: Flow reorganization (Case 2).

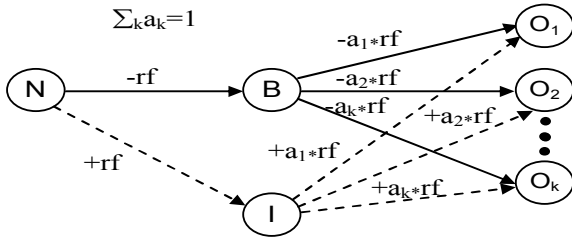


Figure 5: Flow reorganization (Case 2 alternate).

#### 4. Numerical Result

For all evaluation purposes a square grid of a side length of 300 meters has been chosen to deploy nodes uniformly at random. The position of the base station has been fixed at coordinate (0, 0) in the grid. For our sample experiment we have both the number of active sensors (between 1 and 10) and the population (i.e., the density) as factors, and thus we will show our result in a 3-D graph. By active sensors we imply that of all the nodes in the network we are choosing in a range of 1 -10 nodes that are sensing along with relaying.

We have randomly selected the active sensor nodes from the entire node population. And to see how the network scales we gradually increase the number of nodes in the network. For each density and active sensor value we have run enough random experiments, averaging the obtained lifetime, to claim a 95% confidence that our relative error is less than 5%.

We have used the power model of [1] and [12] ; the power related constants are as follows:  $\alpha_{11}=45\text{nJ/bit}$ ,  $\alpha_{12}=135\text{nJ/bit}$ ,  $\alpha_2=10\text{ pJ/(bit*m}^2)$  and thus the path loss index we used is  $n=2$ . Furthermore, we set  $\alpha_3=50\text{nJ/bit}$ , and  $\alpha_4=1\text{nJ/bit}$  (typical values for  $\alpha_4$  are between couple of pJ/bit and couple of 10nJ/bit). Active sensors are transmitting with a rate of  $r=100\text{bits/sec}$ . For the simulated annealing process the initial temperature has been selected as 1000000 and is being reduced by a factor of 2. At each temperature 30 iterations are carried out.

To normalize our lifetime results, we fixed the initial energy of all nodes. The initial lifetime is derived by placing a sensor and a sink 1 meter apart. All other experiments' lifetimes are then normalized to this value, thus showing lifetimes in fractions of that of the benchmark network's lifetime.

Figure 6 shows the lifetime results obtained through our optimization solution for the system model. The figure reveals that the denser a sensor network is the more the lifetime we can expect. This is due to the fact that nodes are closer to each other and thus have to spend less energy to transmit. Another observation we can make is that the lifetime does not significantly depend on the number of active sensors as long as their number is small compared to the total number of sensors. This can be attributed to the many diverse ways that flows can be routed from sensors to the sink.

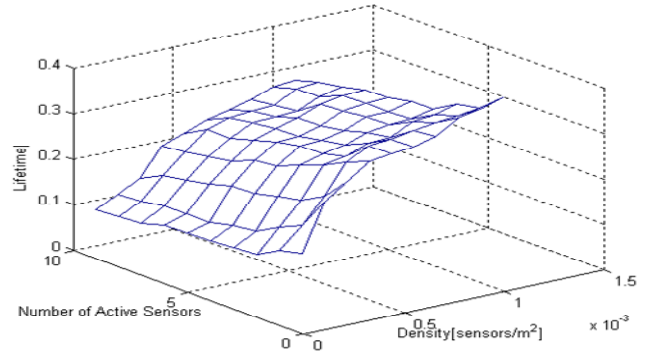


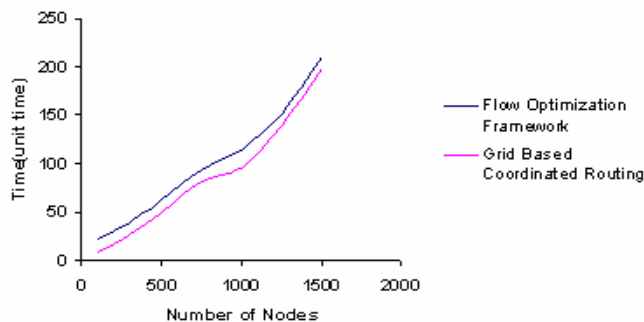
Figure 6: Lifetime vs. density and active sensors

#### 5. Flow Optimization Framework as a benchmark

In order to use this flow optimization framework as a benchmark we will review a simple routing strategy and then see how well it compares to the optimal solution. We will consider grid based coordinated routing [15] where certain nodes in the network are selected as coordinators. Fixed energy powered nodes are placed randomly in the network. The sensor field is divided into square grids and for each grid a coordinator is selected, which stays active till it runs out of

energy. The source node starts flooding the network with query messages and sends them out to each coordinator. The coordinators form the path to transmit data from the source to the sink. When a coordinator runs out of energy, a new coordinator is chosen. This process continues until the network is partitioned and there is no connectivity between the source and the sink. Hence the partition time is considered to be the network lifetime.

In order to compare this routing strategy with our framework we have made a slight change. We have fixed the grid size, energy and power levels as used in [15]. Also now instead of considering the network lifetime as the time when the first node runs out of energy, we will consider the lifetime to be the loss of connectivity between the source and the sink.



**Figure 7:** Optimization Framework Vs Grid Based Coordinated Routing

As can be seen from Figure 7 grid based coordinated routing can do much better. The results show that the routing strategy can do at least 1.5 times better as compared to what it is doing right now. This is how the flow optimization framework can be used as a benchmark to compare the performance of routing protocols.

## 6. Conclusions and Future Work

This paper addressed the lifetime of sensor networks. More precisely, we have set up a framework where optimization variables are data flows that go in and out of sensor nodes. By optimizing these flows, the maximum lifetime of the sensor network can be calculated. We have implemented our optimization approach in C++ using modified simulated annealing and run experiments varying the density and the number of active sensors. We have also shown how this framework can be used as a benchmark to compare the performance of existing routing strategies.

Furthermore, in a real sensor network, nodes will sense the occurrence of an event (e.g., an entity moving through the sensor network). Here, we have assumed that active sensors are uniformly distributed. We are currently modifying our framework to account for a more concentrated active sensor distribution. We are also trying to introduce mobility in the network and see how that would affect the lifetime. That way the framework would be more useful since then it could be used to compare performance of real world applications. Also it would then be feasible to not only consider density distribution and lifetime, but other performance parameters that are integral to evaluating a sensor network.

## References:

- [1] M. Bhardwaj, and A.P. Chandrakasan, "Bounding the Lifetime of Sensor Networks via Optimal Role Assignments," Proceedings of the IEEE INFOCOM, vol. 3, pp. 1587 – 1596, June 2002.
- [2] I.F. Akyildiz, W. Su, Y. Sanakarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," IEEE Computer Networks Magazine, vol. 38(4), pp. 393-422, March 2002.
- [3] M. Bhardwaj, "Lifetime Modeling in a Sensor Network," Proc. of IEEE INFOCOM, vol. 1, pp.202 - 203 , 2002.
- [4] Y. Ganjali, and A. Keshavarzian, "Load Balancing in Ad Hoc Networks: Single-path Routing vs. Multi-path Routing," Proceedings of the IEEE INFOCOM, vol. 2, pp. 1120 – 1125, March 2004.
- [5] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networking," Proceedings of the IEEE INFOCOM, vol 1, pp. 22 – 31, March 2000.
- [6] V. Mhatre, and C. Rosenberg, "Design Guidelines for Wireless Sensor Networks: Communication, Clustering and Aggregation," Ad Hoc Networks Journal, Elsevier Science, pp. 45-63, 2004.
- [7] S. Singh and C. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks," ACM Computer Communication Review, vol. 28 , Issue 3, pp. 5- 26, July 1998.
- [8] S. Singh, M. Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks," Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pp. 181 – 190, 1998.
- [9] P. Krishna, N. Vaidya, M. Chatterjee, and D. Pradhan, "A cluster-based approach for routing in dynamic networks," ACM SIGCOMM Computer Comm. Review, vol. 27 , Issue 2, pp. 49 – 64, April 1997.
- [10] Paolo Santi, "Topology control in wireless ad hoc and sensor networks," ACM Computing Surveys (CSUR), vol. 37, Issue 2, pp. 164 – 194, June 2005.
- [11] Honghai Zhang and Jennifer Hou, "On Deriving the Upper Bound of a-Lifetime for large Sensor Networks," Proceedings of the ACM MobiHoc, vol. 1(2), pp. 272 – 300, November 2005.
- [12] W. Heinzelman, "Application Specific Protocol Architectures for Wireless Networks", PhD Thesis, MIT, 2000
- [13] Paschalidis,I.C.Lai,W. Starobinski D., "Asymptotically Optimal Transmission Policies for Large-Scale Low-Power Wireless Sensor Networks", IEEE/ACM Transactions on Networking vol. 15, pp.105-118, February 2007
- [14] Arvind Giridhar,Kumar P.R. "Maximizing the functional lifetime of sensor networks", IPSN 2005, pp.5-12,April 2005
- [15] Robert Akl, Uttara Sawant,"Grid Based Coordinated Routing in Wireless Sensor Networks",Proceedings of IEEE CCNC 2007: Consumer Communications and Networking Conference, January 2007