

# Maximum Network Lifetime in Fault Tolerant Sensor Networks

Petar Djukic and Shahrokh Valaee

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering  
University of Toronto, 10 King's College Road, Toronto, ON, M5S 3G4, Canada  
e-mail: {djukic, valaee}@comm.utoronto.ca

**Abstract**—This paper introduces a novel technique to maximize the lifetime of fault tolerant sensor networks. The proposed architecture uses multipath diversity in the network layer and erasure codes. We use a *distributed sink* where information arrives at the sink via multiple proxy nodes, called “prongs” in this paper. The sender node uses erasure coding and splits each packet into multiple fragments and transmits the fragments over multiple parallel paths. The erasure coding allows the sink to reconstruct the original packet even if some of the fragments are lost. Occasionally, the sink broadcasts a query to awaken the sensors and to allow them to collect information about probability of packet loss and energy consumption in the network. The awakened sensors then use the collected information to distribute their data among different prongs so as to maximize the network lifetime, while keeping reliability in the network above a certain level.

**Index Terms**—Sensor networks, system design, fault tolerance, mathematical programming/optimization

## I. INTRODUCTION

Sensor networks are wireless ad hoc networks used for monitoring and information gathering. They are used to observe natural phenomena such as seismological and weather conditions, collect data in battlefields, and monitor traffic in urban areas. Sensor networks consist of many small, self-organized nodes that form an ad hoc network that reports to a common sink at the edge of the network.

In a typical sensor network, the sink sends a query which is disseminated throughout the sensor network with flooding. The query requests a subset of nodes to send their collected information. The nodes, which have the requested information, send it to the sink by forming a tree with the root at the receiver. However, collection of information with a single sink may not be appropriate for sensor networks. For a single sink, the top level of the tree contains relatively few nodes, compared to the total number of nodes. So, most of the energy is consumed by the few nodes close to the sink. This makes the nodes closest to the sink prone to energy drainage.

In this paper, we propose a sensor network architecture that maximizes the network lifetime, while increasing the fault tolerance in the network. Our design assumes that the sink is distributed throughout the sensor network. The sink uses a number of receivers—called “prongs”—that connect to it with reliable and high bandwidth links. We assume that if a packet arrives at a prong, it will be delivered intact to the sink. This creates a hierarchical architecture where each sensor can be

connected to a prong directly or with multiple hops through other sensor nodes. This architecture distributes the load on the last hop among a larger set of nodes than with a centralized sink design. Therefore, it removes the possibility for a single point of failure.

In order to increase the reliability of transmission, each node in the network (a sensor) sends packets over multiple disjoint paths<sup>1</sup>. One way to increase reliability is to send copies of the same packet over the multiple paths. However, this would be very inefficient. Instead, the reliability can be increased efficiently with Forward Error Correction (FEC). The source splits each packet into many fragments and generates parity fragments with an erasure code [1]. The fragments are then distributed over the paths and simultaneously sent to the sink. The sink can reconstruct the packet if it receives a portion of the fragments which is of the same size as the original packet. Using erasure codes with multiple paths to increase reliability was shown to be effective previously in [2], [3].

The distributed sink and multiple paths also allow us to use load-balancing, which increases the network lifetime. Load-balancing is a natural consequence of using multiple paths. However, in this paper we show that it is possible to distribute the fragments on the paths so as to maximize the network lifetime, while the reliability is kept above a certain threshold. We define the network lifetime as the time after which at least one node will lose all of its energy [4]. This definition of network lifetime is important, since losing a node might have a serious impact on the network connectivity. Usually, this means that some of the paths need to be rerouted and that other nodes need to pick up the traffic from the node that is out of service. The other definition of the network lifetime is the time for the last node to consume all its energy. In this case, network lifetime can be maximized by minimizing the total energy consumption of each sensor, while keeping the reliability above a certain level [3].

## II. NETWORK DESIGN FOR FAULT TOLERANT SENSOR NETWORKS

In this section, we show how the proposed scheme can be implemented in practice. There are two questions raised by our design. First, how does the network layer find multiple

<sup>1</sup>Here, we use multipaths in the network layer, as opposed to a scheme that may use multipaths in the physical layer.

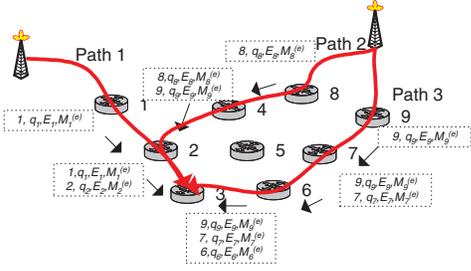


Fig. 1. Our sensor network design. The sink sends queries by flooding the network. The query carries information about the conditions on every hop in the network. This improves load-balancing and increases reliability.

disjoint paths between the sensors and the sink? And second, how do we optimize the use of these paths? We answer the first question in the rest of this section. The second question is the subject of the subsequent sections.

We position our design within the context of existing sensor networks. We assume that our network operates with a system similar to TinyDB [5]. In our design, the sink sends queries from all prongs at the same time. Figure 1 illustrates an example for the query/route discovery operation. The sink floods the network by sending the query through each of the prongs. The query records the path it takes and the conditions on each hop of the path. If a query travels in a loop, it will be discarded. The reported states are the probability of successful fragment transmission ( $q_i$ ) and energy ( $E_i, M_i^{(e)}$ ) parameters on each hop. The information about the reliability and energy on each hop is used in the maximization of the network lifetime. We give more details on how  $q_i$ ,  $E_i$ , and  $M_i^{(e)}$  are collected and packed into three small query fields in Section III.

Each sensor node records its information in the query and forwards the query to the next set of nodes. The sensor nodes also keep track of each path that arrives at a prong. If a node needs to send back a reply to the query, it selects a set of *arc-disjoint*<sup>2</sup> paths to the prongs on which it can send the reply. For example, in Figure 1, node 3 can use path 2 or 3 to the prong on the right and path 1 to the prong on the left. In this case, node 3 should select paths 1 and 3.

A sensor node may select the paths with the shortest hop count, or with the highest reliability. The shortest path cost may not be the best metric for sensor networks. Indeed, [6] shows that the shortest path cost selects the nodes which use a few long hops, and therefore it decreases reliability. The selection methods which take reliability into account perform much better. In this paper, we do not present a path selection algorithm. The contribution of this paper is to maximize the network lifetime once the paths have been identified.

We note that, compared to a packet size used in regular networks, the size of the packets sent by the sensors may be quite small. However, our scheme can still work on the nodes which aggregate the data from other sensor nodes. The

aggregation is normally used to eliminate the redundant data reported by the nodes with coding techniques [7]. Our scheme amplifies the benefits of aggregation by increasing reliability. The aggregating nodes have more important data to send than the regular nodes since they are responsible for sending data from multiple child nodes. Our scheme allows the aggregating nodes to increase their reliability, making it less likely to lose their data.

### III. A MODEL FOR FAULT TOLERANT SENSOR NETWORKS

We assume that a sensor node generates a packet of size  $bM$  bits every  $D_M$  seconds. The packet is split into  $M$  fragments, each with size  $b$ , and  $K$  additional parity fragments with size  $b$  are generated using a linear erasure code [1]. The source node then distributes the fragments over  $n$  parallel paths, by allocating  $x_i, i = 1, \dots, n$ , fragments on path  $i$ . Since the total number of fragments is  $M + K$  we have  $\sum_{i=1}^n x_i = \mathbf{x}^T \mathbf{1} = M + K$ . We denote the allocation vector with  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  and use  $\mathbf{1}$  to denote a vector of all 1s.

The destination node needs to receive a total of  $M$  fragments in order to reconstruct the packet. We use random variables  $Z_i$  to indicate the number of fragments received on path  $i$  in  $D_M$  seconds. So, the probability that the packet can be reconstructed is given by:

$$P_{\text{succ}} = \Pr \left[ \sum_{i=1}^n Z_i > M \right]. \quad (1)$$

$P_{\text{succ}}$  is the measure of reliability in the network.  $P_{\text{succ}}$  is a function of  $\mathbf{x}$  the allocation of fragments on each path,  $\mathbf{q} = [q_1, \dots, q_n]$  the vector indicating the probability that a fragment will be successfully transmitted on each path, and  $K$  the number of parity fragments. We will use  $P_{\text{succ}}$  and  $P_{\text{succ}}(\mathbf{x}, \mathbf{q}, K)$  interchangeably in the rest of the paper.

If we approximate the loss of consecutive fragments on each path to be independent and identical to each other, we can approximate  $P_{\text{succ}}(\mathbf{x}, \mathbf{q}, K)$  with:

$$P_{\text{succ}} \geq Q(\mathbf{x}, \mathbf{q}, K) \quad (2)$$

where

$$Q(\mathbf{x}, \mathbf{q}, K) = \sum_{j=0}^K \frac{e^{-\lambda(\mathbf{x})} [\lambda(\mathbf{x})]^j}{j!} \quad \text{and} \quad \lambda(\mathbf{x}) = - \sum_{i=1}^n q_i \ln x_i. \quad (3)$$

We show that  $P_{\text{succ}}$  can be approximated with (2) in [2] by using the results of [8]. It was also shown in [8] that inequality (2) is single sided, that is  $P_{\text{succ}}$  is always greater than  $Q$ . This allows us to replace  $P_{\text{succ}}$  in our optimizations with  $Q(\mathbf{x}, \mathbf{q}, K)$  which is easier to analyze.

We measure the effectiveness of the scheme, in terms of the overhead introduced by the erasure code, as:

$$\eta = \frac{M}{M + K} \quad (4)$$

where  $\eta$  is the *efficiency* that we can achieve. The scheme is more effective as  $\eta$  approaches 1. We will use a lower bound

<sup>2</sup>Two arc-disjoint paths have no vertices or edges in common.

on  $\eta$  in the maximization of network lifetime to make sure that efficiency always stays above a certain threshold.

We assume that each sensor delivers information to the sink at the same rate, given by:

$$R = \sum_{i=1}^n R_i = \frac{b}{D_M}(M + K), \quad (5)$$

where  $R_i = bx_i/D_M$  is the rate offered on the  $i$ th path.

The sink sends a query to the sensor every  $D_M$  seconds in order to collect the fragment loss information on each path between the two nodes. Given the probability that a fragment transmission is successful on a link, we can calculate the probability that a fragment is successful on the path as follows:

$$q_i = \prod_{k=0}^{n_i-1} q_i^{(k)} \quad (6)$$

where  $q_i^{(k)}$  is the probability that a fragment is transmitted successfully on the  $k$ th link on path  $i$ , and  $n_i$  is the number of nodes on that path. We assume that each sensor node is keeping track of  $q_i^{(k)}$  using a passive monitoring technique similar to [9].

We assume that every node in the network has the ability to measure the amount of available energy<sup>3</sup> on the node  $E_i^{(k)}$ , and the average amount of energy it uses to transmit a bit of information  $e_i^{(k)}$ , where we index the node as ‘‘node  $k$  on path  $i$ ’’. To simplify the optimization, we assume that  $e_i^{(k)}$  and  $E_i^{(k)}$  do not change during the packet transmission.

A query sent by a prong to the network can also inform the sensors about the per-bit energy required to transfer a packet between each sensor and that prong,  $e_i$ . The per-bit energy consumption on a path can be determined by adding up the energy required to transfer a bit at every node on the path:

$$e_i = \sum_{k=0}^{n_i-1} e_i^{(k)} \quad (7)$$

where  $n_i$  is the number of nodes on path  $i$ . The vector of per-bit energy consumption is given by  $\mathbf{E}_b = [e_1, e_2, \dots, e_n]^T$ . So, the total amount of energy used to transmit the  $M + K$  fragments is given by:

$$E_{\text{Total}}(\mathbf{x}, \mathbf{E}_b) = b\mathbf{x}^T \mathbf{E}_b. \quad (8)$$

We also have access to the maximum number of fragments that can be transmitted on each path before the energy on the path runs out,  $M_i^{(e)}$ :

$$M_i^{(e)} = \min_{1 \leq k \leq n_i} \left\{ \frac{E_i^{(k)}}{be_i^{(k)}} \right\}. \quad (9)$$

$M_i^{(e)}$  is calculated from the information in the query at each node  $k$ , using

$$M_i^{(e)} \leftarrow \min \left\{ M_i^{(e)}, \frac{E_i^{(k)}}{be_i^{(k)}} \right\}. \quad (10)$$

<sup>3</sup>The available energy is a coarse estimate of the energy left after all the other sensors sharing the node transmit their packets. The accuracy of  $E_i^{(k)}$  depends on how often the sink sends queries to the nodes.

We denote with  $\mathbf{M}_e$  the vector of maximum number of fragments we can transmit on each path, i.e.  $\mathbf{M}_e = [M_1^{(e)}, M_2^{(e)}, \dots, M_n^{(e)}]^T$ .

#### IV. MAXIMIZATION OF NETWORK LIFETIME

In this section, we show how to maximize the network lifetime, while the network reliability, efficiency and energy consumption are bounded. We defined network lifetime earlier as the time until the first node loses all of its energy. We now show how the network lifetime can be calculated using the information collected by the source node, as shown in Sect. III. The lifetime of node  $k$  on path  $i$  is given by the amount of time it takes the source to consume all of the energy it has on that node:

$$T_i^{(k)} = \frac{1}{R_i} \frac{E_i^{(k)}}{e_i^{(k)}} = \frac{D_M}{bx_i} \frac{E_i^{(k)}}{e_i^{(k)}}. \quad (11)$$

The lifetime of each path can be found from the energy information carried in the query:

$$T_i = \min_{1 \leq k \leq n_i} \{T_i^{(k)}\} = \min_{1 \leq k \leq n_i} \left\{ \frac{D_M E_i^{(k)}}{bx_i e_i^{(k)}} \right\} = \frac{D_M}{x_i} M_i^{(e)} \quad (12)$$

where  $x_i$  is the number of fragments transmitted on the  $i$ th path and  $M_i^{(e)}$  is defined in (9) as the maximum number of fragments that can be transmitted on that path due to energy constraints. Using this definition of network lifetime we define the optimization of network lifetime as:

$$\text{Maximize: } T_{\text{net}}(\mathbf{x}) = \min_{\mathbf{x}, K} \left\{ \frac{D_M M_i^{(e)}}{x_i} \right\} \quad (13a)$$

$$\text{Subject to: } Q(\mathbf{x}, \mathbf{q}, K) \geq \epsilon \quad (13b)$$

$$\eta(\mathbf{x}, K) \geq \delta \quad (13c)$$

$$b\mathbf{x}^T \mathbf{E}_b \leq E_{\text{max}} \quad (13d)$$

$$0 \preceq \mathbf{x} \preceq \mathbf{M}_e \quad (13e)$$

$$\mathbf{x}^T \mathbf{1} - K = M \quad (13f)$$

where  $\preceq$  is pairwise vector comparison.

The first constraint, (13b) is to ensure the reliability is kept above a certain level. Increase in the reliability means that there will be less need for retransmissions in the network, thus decreasing total energy use. We use (2) to evaluate the reliability in the constraint. The second constraint, (13c), bounds the number of parity fragments. The third constraint, (13d), ensures that the total transmission energy is limited. There are two reasons to add  $E_{\text{max}}$  as the maximum amount of energy. First, the total energy consumption will depend on the number of parity fragments used to increase the reliability, as well as, the number of actual data fragments. Both of these factors affect energy consumption in the network. Second, even if the time until the first sensor runs out of energy is maximized, the time at which the last node runs out of power may not be maximized. So, it is possible that the optimal network load may consume energy more than the network load that minimizes total energy consumption. This would decrease

the time until all the nodes run out of energy, decreasing the operational time of the network. The sensor uses the energy restriction to prevent this from happening. The fourth constraint (13e) bounds the total number of fragments that can be transmitted on each path. The fifth constraint, (13f), ensures that total number of allocated fragments is  $M + K$ .

The optimization without constraints (13b)-(13e) is an approximation of the optimization in [4]. The optimization (13) does not consider all the possible paths between the sensor and the prongs. The sensor node uses the paths, which are arc-disjoint, in order to maximize the diversity in the network and increase reliability. However, the optimization in [4] is ideal and indeed even [4] presents an approximation of that optimization. Our formulation enhances the general problem of maximizing the network lifetime by adding reliability bounds.

The problem is solved by first replacing the constraint on the efficiency with:

$$0 \leq K \leq K_{\max} \quad (14)$$

where  $K_{\max} = \frac{1-\delta}{\delta}M$ .

Second, we can transform the problem into a linear programming problem using the fact that  $Q(\mathbf{x}, \mathbf{q}, K)$  is a monotonically decreasing function of  $\lambda(\mathbf{x})$  for a fixed  $K$  [2]. So, for a given  $K$  there exists  $\alpha_\epsilon(K)$  such that:

$$\lambda(\mathbf{x}) \leq \alpha_\epsilon(K) \leftrightarrow Q(\mathbf{x}, \mathbf{q}, K) \geq \epsilon \rightarrow P_{\text{succ}} \geq \epsilon. \quad (15)$$

We have show in [2] that  $\alpha_\epsilon(K)$  is almost linear for values of  $K > 5$ . So, we approximate  $\alpha_\epsilon(K)$  as a straight line for a given  $\epsilon$ :

$$\alpha_\epsilon(K) \approx s(\epsilon)K + c(\epsilon). \quad (16)$$

This allows us to simplify the reliability bound and replace (13b) with:

$$-\mathbf{x}^T \ln(\mathbf{q}) - s(\epsilon)K \leq c(\epsilon) \quad (17)$$

The slope  $s(\epsilon)$  and constant  $c(\epsilon)$  can be obtained with any number of techniques and stored on the nodes prior to field deployment of the sensors. In our simulations, we have used the least-square method.

Third, we convert the objective function so that (13) becomes a minimization problem with a linear objective function. We note that :

$$T_{\text{net}}^{\text{cvx}}(\mathbf{x}) \triangleq \max_{1 \leq i \leq n} \left\{ \frac{x_i}{D_M M_i^{(e)}} \right\} \quad (18)$$

is a linear function with the same optimum point as (13a) since the variables  $x_i$  are positive. In our simulation we have also converted the optimization into a standard linear program by introducing a new variable  $t$  which becomes the new objective function and by adding  $n$  new constraints  $x_i/M_i^{(e)} \leq t$  to the original problem (13).

## V. SIMULATION RESULTS

In this section, we present the simulation results. We use a network with four prongs and a single sink for our simulations. We study our method using two examples.

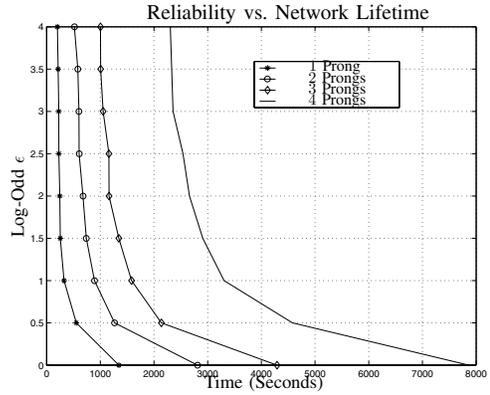


Fig. 2. The network reliability with the network lifetime. As the minimum reliability  $\epsilon$  increases, the network lifetime increases. However, the network lifetime is increased with the number of prongs.

TABLE I  
THE LOG-ODD TRANSFORM

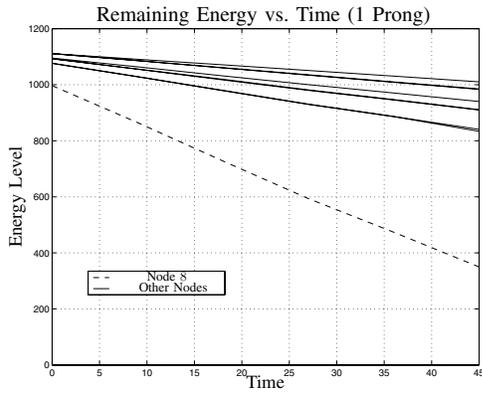
$\log_{10} \left( \frac{\epsilon}{1-\epsilon} \right)$	$\epsilon$	$\log_{10} \left( \frac{\epsilon}{1-\epsilon} \right)$	$\epsilon$
0.0	0.5	3.0	0.999
1.0	0.909	4.0	0.9999
2.0	0.990	5.0	0.99999

In the first example, the network consists of nine nodes forming a grid. We assume that the sink sends a query at regular intervals of  $D_M = 1$  sec to each of the nodes; the nodes answer by sending their data as described in Sect. II and Sect. III. The query carries information about the reliability ( $q_i$ ) and the energy ( $E_i, M_i^{(e)}$ ) as explained in Sect. III.

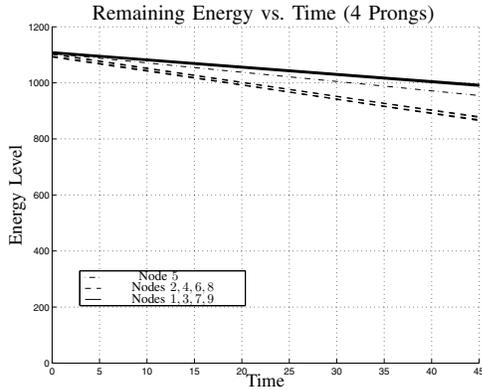
We have generated 100 query responses in which all links had a fragment success rate randomly distributed in the interval  $[0.85, 0.95]$  with the mean of 0.9. This makes the average path reliability 0.81 for two hop paths and 0.73 for three hop paths. Each sensor has enough energy to transmit 100,000 fragments and each sensor transmits packets of size  $M = 100$  fragments. Before transmitting a query response the sensor uses optimization (13) to calculate the number of fragments  $K$  required to achieve the reliability threshold  $\epsilon$  and the distribution of fragments on each path. We ran this simulation with a different number of active prongs.

Figure 2 shows the effect of minimum reliability  $\epsilon$  on network lifetime. The vertical axis is plotted in the “log-odd” format, where instead of  $\epsilon$  we show  $\log[\epsilon/(1-\epsilon)]$ . This allows us to see differences between values of  $\epsilon$  that are very close to 1. We give some representative values for the log-odd transformation in Table I. The horizontal axis shows the time at which the first node loses all of its energy, or network lifetime. Figure 2 shows that the network lifetime increases as we increase the number of prongs. The network lifetime also decreases as reliability increases because the sensors have to transmit more parity fragments to increase the reliability.

We show why the network lifetime increases with the number of prongs in Figure 3. The figure shows the energy



(a) Single Prong.



(b) Multiple Prongs.

Fig. 3. The left-over energy over time. We show the energy left on each of the nodes as time progresses. (a) The scenario with one prong. (b) The scenario with four prongs.

level at each node as a function of time. Figure 3(a) shows the energy levels of the nodes when only one prong is turned on. Figure 3(b) shows the energy levels when all four prongs are turned on. We observe that in the single prong scenario the node which spends the most energy is the node closest to the sink, node 8 in our simulation. In the four prong scenario, the available energy is distributed roughly the same on all of the nodes increasing the network lifetime.

In the second example, we simulate a network with 10,000 sensors randomly displaced in a rectangular area. We assume that every sensor can connect to four prongs located at the edge of the network. We simulate our scenario with four different values for the average path success rate  $\bar{q} = 0.6, 0.7, 0.8, 0.9$ . The energy consumption along each path is a random number with the mean 0.5.

Figure 4 illustrates the decrease of the network lifetime as a function of the minimum reliability. The top horizontal line with the value 0 corresponds to the lifetime when there is no erasure coding used in the system; this approximates [4]. The plots show the decrease of the network lifetime when  $K$  increases to achieve the reliability bound. In Fig. 4, we can argue that the lifetime is not substantially affected by the increased level of reliability. For example, even when

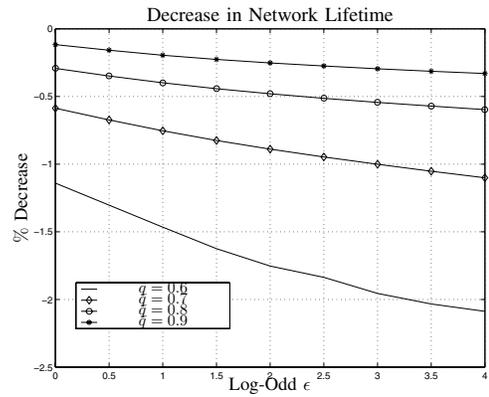


Fig. 4. The decrease in network lifetime versus network reliability. The increase in  $\epsilon$  has almost no significant impact on the network lifetime.

the average network reliability is  $\bar{q} = 0.6$  the lifetime does not decrease more than 2.5%, for the minimum reliability of  $\epsilon = 0.9999$ . This is due to the load balancing capability of our scheme.

## VI. CONCLUSION

We have proposed a new architecture for sensor networks that uses a sink with many prongs to increase load-balancing in the network. Load-balancing is achieved by using multiple paths to transmit information from each source to the sink. Each sensor also encodes the packets into fragments with an FEC code so that reliability in the network is increased. The distribution of fragments is calculated locally on every sensor, with the optimization we solve in the paper. The simulation results show that the network lifetime is not significantly affected when the reliability of the network is substantially increased and that the design with multiple prongs outperforms a single prong design.

## REFERENCES

- [1] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *Journal of the Association for Computing Machinery*, vol. 36, no. 2, pp. 335–348, April 1989.
- [2] P. Djukic, "Optimum resource allocation in multipath ad hoc networks," Master's thesis, University of Toronto, 2003.
- [3] P. Djukic and S. Valaee, "Minimum energy fault tolerant sensor networks," in *proceedings of IEEE GLOBECOM Wireless Ad hoc and Sensor Networks Workshop*, 2004, pp. 22–26.
- [4] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *INFOCOM*, vol. 1, 2000, pp. 22–31.
- [5] J. Gehrke and S. Madden, "Query processing in sensor networks," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 46–55, January-March 2004.
- [6] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges for reliable multihop routing in sensor networks," in *SenSys'03*. ACM Press, 2003.
- [7] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *INFOCOM*, 2004.
- [8] R. J. Serfling, "Some elementary results on Poisson approximation in a sequence of Bernoulli trials," *SIAM Review*, vol. 20, no. 3, pp. 567–579, July 1978.
- [9] A. Woo and D. Culler, "Evaluation of efficient link reliability estimators for low-power wireless networks," U.C. Berkeley Computer Science Division, Tech. Rep. UCB/CSD-03-1270, November 2003.