

## Maximizing the Lifetime of Query Based Wireless Sensor Networks Using Ftqc Algorithm

N.Nagajothi<sup>1</sup>, A.S.Lakshmi<sup>2</sup>, V.Vimaladevi<sup>3</sup>, A.Rajeshwari<sup>4</sup>  
<sup>1, 2, 3, 4</sup>(AP/CSE, P.S.R.Rengasamy college of engineering for women, sivakasi, India)

---

**Abstract:** Data sense and recovery in wireless sensor systems have a general request in areas such as defense and inspection monitoring, and command and control in battlefields. In query-based wireless sensor systems, a user would subject a question and expect a answer to be returned within the limit. While the use of fault acceptance devices through idleness improves query reliability in the presence of changeable wireless message and sensor faults, it could reason the energy of the system to be quickly tired. Therefore, there is an inherent trade-off between query reliability versus energy consumption in query-based wireless sensor systems. In this paper, develop fault-tolerant quality of service (QoS) control algorithms based on hop-by-hop data delivery utilizing “source” and “path” redundancy, with the goal to satisfy application QoS requirements while prolong the duration of the sensor system. To develop a mathematical model for the lifetime of the sensor system as a function of structure parameters including the “source” and “path” redundancy levels utilized and discover that there exists optimal “source” and “path” redundancy below which the lifetime of the system is maximized while satisfying application QoS requirements. Numerical data are presented and validated through extensive simulation, with bodily understandings given, to show the possibility of our algorithm design.

**Keywords**—Wireless sensor networks, reliability, timeliness, query processing, redundancy, energy conservation, QoS, mean time to failure.

---

### I. Introduction

Over the last few years, a rapid increase in the amount of applications for wireless antenna networks (WSNs). WSNs are used in battleground application and a variety of medium health organization and condition-based protection applications on business, armed, and space platforms. For armed users, a main center has been area monitoring (security and surveillance applications). In query-based systems; a user would issue a query with quality of service (QoS) requirements in terms of consistency and timeliness. Retrieving sensor data such that QoS supplies are satisfied is a challenging difficulty and has not been considered until recently [4, 5, 6, 7, 8, 9]. The common approach is to relate redundancy to assure the QoS requirement. In this paper also interested in applying redundancy to satisfy application particular consistency and timeliness requirements. Moreover, to determine the optimal redundancy level that could satisfy QoS conditions while prolonging the duration of the WSN. Specifically, to develop the notion of “path” and “source” level redundancy. When given QoS requirements of a query, identify optimal path and basis idleness such that not only QoS requirements are fulfilled, but also the lifetime of the structure is prolonged. Develop a step-by-step data delivery method to achieve the beloved level of redundancy.

### II. Related Works

Existing research efforts related to applying redundancy to satisfy QoS requirements in query-based WSNs fall into three categories: traditional end-to-end QoS, reliability assurance, and application-specific QoS [4]. Conventional back-to-back QoS solutions are based on the idea of back-to-back QoS requirements. The difficulty is that it may not be possible to apply back-to-back QoS in WSNs due to the complexity and high cost of the protocols for resource constrained sensors. The Sequential Assignment Routing (SAR) [5] that utilizes path redundancy from an origin node to the go under node. Each sensor uses a SAR algorithm for trail selection. The goal of the SAR algorithm is to minimize the standard biased QoS metric all over the lifetime of the network. The algorithm does not consider the reliability issue.

ReInForM has been proposed [6] to addresses back-to-back reliability issues. ReInForm consider information responsiveness and flexibility to channel errors along with a differentiated portion approach of system resources based on the criticality of data. The procedure sends many copies of a small package along various paths from the source to the sink such that data is delivered with the desired reliability. It uses the concept of dynamic packet state in the context of sensor networks to control the number of paths required for the desired reliability, and does so using local information of the control error rate and topology. However, the protocol only concerns QoS in term of reliability.

In [7], M. Perillo et al. provide application QoS with the objective of maximize the duration of WSNs though satisfying a smallest level of reliability. This maximization is achieved through the combined optimization of

scheduling dynamic sensor sets and finding trails for data direction-finding. The natural life is defined as the amount of the instant that all sensor places are used. The approach employs the plan of turning off redundant sensors for episodes of time to save power while considering the tradeoff between energy consumption and reliability. Nevertheless, this approach is not scalable and QoS is limited to application reliability only.

Recently, a multi-path and multi-speed routing protocol called MMSPEED is proposed in [8] which takes into account both timeliness and reliability as QoS requirements. The aim is to provide QoS hold that allows packets to decide the most proper grouping of service options depending on their correctness and consistency requirements. The protocol provides back-to-back QoS provisioning by employing restricted geographic forward using instant neighbor information without end-to-end path discovery and protection. The procedure adapt to network dynamics. However, it does not believe power issues. Our hop-by-hop data delivery mechanism derives from geographical forwarding as considered in this paper. However our hop-by-hop data delivery mechanism specifically forms  $m$  redundancy paths for path redundancy and  $m_s$  sensors for source redundancy to satisfy the compulsory QoS requirements, facilitating the resolve of the best  $(m, m_s)$  that would maximize the lifetime of the WSN.

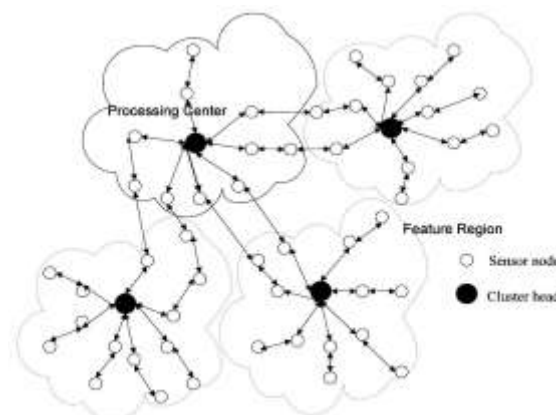
In [9], QoS is defined as the optimum number of sensors that should be transfer in sequence to the sinks at a few given time. The set of rules utilize the stand station to correspond QoS information to all of the sensors using a transmit control. It exploits the arithmetical model of the Gur sport to with dynamism adjust to the best number of sensors. The goal is to exploit the lifetime of the sensor network by having sensors periodically power-down to conserve energy, and at the same time have enough sensors powered-up and sending packets to the sinks to collect enough data. QoS metrics for data release such as consistency and timeliness are not considered.

Our approach of satisfying application reliability and timeliness conditions while maximizing the system natural life is to conclude the best level of job loss at the “source” and “path” levels. The basic level redundancy refers to the use of many antennas to return the requested sensor evaluation. The trail level job loss refers to the use of several paths to communicate the sense to the sink node. Since sensor networks are constrained with resources, develop a hop-by-hop data dissemination paradigm to dynamically form multiple paths for data delivery, instead of incurring additional fixed cost to first prepare various paths previous to data release

### III. System Model

A WSN consists of a set of low-power sensor nodes typically deployed through air-drop into a geographical area. Assume that sensors are indistinguishable with the similar first power level. Sensors group themselves into distinct clusters in the system for energy conservation purposes, with each cluster ideally responsible for a feature neighborhood, as illustrated in Figure 1. Each group elects a antenna to be the cluster head. The role of a cluster head is rotated fairly among sensors in the cluster based on a cluster head rotation algorithm such as HEED [1] or LEACH [2] so that sensors will not rapidly reduce their energy. The purpose of a group head is to manage the network within the cluster, and collect/aggregate sensor reading data from the sensors within the cluster. To save energy, the transmission power of a sensor even when it is a cluster head is reduced to a minimum level to enable it to communicate with its neighbor within one-hop radio range.

Assume that users (through a flying airplane or a moving vehicle) can issue a query through any cluster head, which we call it a processing center or a user monitoring node as labeled in Figure 1. A query may involve every or a split of clusters, say,  $k$  clusters, to react to the query for data sense and retrieval. These requested clusters are termed basis clusters. In this paper we assume  $k=1$  for simplicity.



“Fig 1-Cluster based WSN Architecture”

The WSN does not have a base station. In a cluster-based architecture, a source cluster head must relay sensor data information to the processing center in response to a user query, and thus will consume more energy than a pure sensor node. The energy consumed by a source cluster depends on the length of the path connecting the source cluster and the processing center. For fault tolerance reasons, two forms of redundancy are considered for which we analyze their effect on the lifetime of the WSN. Figure 1 illustrates the case in which  $m=2$  and  $m_s =5$ . Distinguish the collapse performance of a sensor due to environment conditions (i.e. hardware failure) by a failure prospect parameter  $q$ . Let the per hop channel error across the entire network be a constant  $e$  (where  $0 < e < 1$ ).

Assume that all sensors are deployed in a square sensor area of size  $A^2$  such that each side is of length  $A$  and homogeneous spatial Poisson process with intensity  $\lambda$ . Let  $n$  be the total number of nodes and  $n_s$  be the number of sensors in a cluster. The size of a cluster depends on the clustering algorithm employed and is a design parameter that will affect the lifetime of the system. The number of clusters in the system,  $N_c$  is given by  $n/n_s$ . A cluster head will perform the function of data collection, aggregation and communication. Let  $E_o$  be the initial energy of each sensor node in Joule. Within a cluster we assume that a cluster election protocol such as HEED [1] is in place that will more or less achieve a perfect rotation of the cluster head among all sensors in the cluster. Thus, given that the number of sensors in a cluster is  $n_s$ , the probability that a node will become a cluster head,  $p$ , is equal to  $1/n_s$ . Thus the total number of clusters in the network,  $N_c$  is given by  $n/n_s = np$ .

Assume a perfect rotation of sensor nodes within a cluster to assume the role of the cluster head, so each; sensor node would consume energy at about the same rate. Then, instead of considering each individual sensor energy level, we can consider the system energy whose initial energy level is given by  $E_{initial} = nE_o$ . When the energy level of the system falls below a threshold value, say  $E_{threshold}$ , the WSN is considered as having depleted its energy. For the energy model, we adopt the radio model in [1]. The energy dissipation to run the transmitter and receiver circuitry is denoted as  $E_{elec}$ . The energy used by the transmit amplifier to achieve an acceptable signal to noise ratio is denoted as  $E_{amp}$ . Also there is an  $r^2$  energy loss due to channel transmission where  $r$  is the transmission radius. Thus the energy spent by a SN to transmit a data packet of  $n_b$  bits a distance  $h$  is given by:

$$E_T = n_b (E_{elec} + E_{amp} r^2) \tag{1}$$

The energy spent to receive a message is given by:

$$E_R = n_b E_{elec} \tag{2}$$

The energy used by a SN for sensing  $E_C$  is assumed to be the same as  $E_R$ . We define the system lifetime or the mean time to failure (MTTF) as the total number of queries the system can answer correctly until it fails due to channel or sensor faults, or when the system energy reaches the energy threshold level  $E_{threshold}$ . We define a query's QoS requirements in terms of its reliability and timeliness requirements, denoted as  $R_{req}$  and  $T_{req}$ . The system must deliver query results within  $T_{req}$  and the reliability of data delivery must be at least  $R_{req}$ . Our objective is to determine the best path and source redundancy levels to satisfy QoS while maximizing MTTF.

#### IV. Probability Model

The Fault-Tolerant Quality of Service Control algorithm developed in this paper takes two forms of redundancy. The first form is path redundancy. That is, instead of using a single path to connect a source cluster to the processing center,  $m_p$  disjoint paths may be used. The second is source redundancy. That is, instead of having one sensor node in a source cluster return requested sensor data,  $m_s$  sensor nodes may be used to return readings to cope with data transmission and/or sensor faults. Fig. 1 illustrates a scenario in which  $m_p = 2$  (two paths going from the CH to the processing center) and  $m_s = 5$  (five SNs returning sensor readings to the CH). Below, we derive analytical expressions for  $R_q$  (query reliability) and  $E_q$  (energy consumption per query) resulting from the use of FTQC. First derive MTTF for the case in which only one source cluster is required to answer a query and only the reverse traffic is considered. Later, we generalize the result to the case in which the forward traffic for query dissemination is considered and in which multiple source clusters are required to answer a query.

##### 4.1 Query Reliability

Let  $R_q$  be the reliability of a query as a result of applying our proposed hop-by-hop data delivery mechanism with  $m$  paths for path level redundancy and  $m_s$  sensors for source level redundancy. Let  $E_q$  be the average energy consumption of the system to answer a query. Below we derive analytical expressions for  $R_q$  and  $E_q$ . Let  $d_{inter}$  be a random variable denoting the distance between a source cluster head (CH) and the

processing center and  $d_{intra}$  be the random variable denoting the distance between a sensor nodes (SN) to the CH. Then the number of hops between the processing center to the source CH, denoted by  $h$ , is given by:

$$h = \frac{d_{inter}}{r} - 1 \tag{3}$$

A query can be initiated by any CH which serves as the processing center for that query. Thus, the location of the processing center varies on query by query basis. For derivation convenience without loss of generality, let the processing center be located in the center of the sensor area with the coordinate at (0, 0) and the source CH be randomly located at (Xi, Yi) in the square sensor area with  $-A/2 \leq Xi \leq A/2$  and  $-A/2 \leq Yi \leq A/2$  and. Then, the expected value of  $d_{inter}$  is given by:

$$E[d_{inter}] = \int_{-A/2}^{A/2} \int_{-A/2}^{A/2} \sqrt{(X_i^2 + Y_i^2)} \left(\frac{1}{A}\right) \left(\frac{1}{A}\right) dX_i dY_i = 0.3825A \tag{4}$$

For notational convenience, let  $N_{inter}^h$  represent the average number of hops (or sensors) to forward sensor data from a source CH to the processing center. Then  $N_{inter}^h$  is given by:

$$N_{inter}^h = [E[h]] = \left[ \frac{0.3825A}{r} - 1 \right] \tag{5}$$

Since a sensor becomes a CH with probability  $p$  and all the sensors are distributed in the area in accordance with a spatial Poisson process with intensity  $\lambda$ , the CH and non-CH sensors will also be distributed in accordance with a spatial Poisson process with rates  $p\lambda$  and  $(1-p)\lambda$ , respectively. The expected distance from a non-cluster-head sensor to the CH is given by:

$$E[d_{intra}] = \frac{1}{2(p\lambda)^{1/2}} \tag{6}$$

If this distance is more than per-hop distance  $r$ , a sensor will take a multi-hop route to transmit sensor data to the CH. The average number of intermediate sensors is the quantity above divided by per-hop distance  $r$ . Let  $N_{intra}^h$  denote the average number of hops to forward sensor data from a SN responsible for a reading to its CH. Then  $N_{intra}^h$  is given by:

$$N_{intra}^h = \left\lceil \frac{1}{2(p\lambda)^{1/2}} \right\rceil \tag{7}$$

Let  $Q_r$  be the probability of a SN failing to receive sensor data.  $Q_r$  is given by:

$$Q_r = [(1 - q)(1 - e)] \tag{8}$$

Let the deadline requirement of a query be  $T_{req}$ . The speed requirement to satisfy the timing constraint is given by [8]:

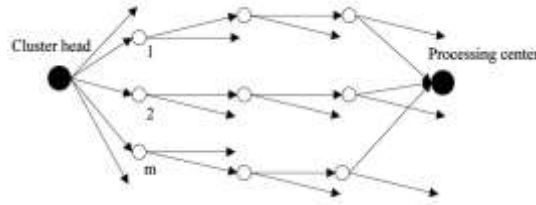
$$X_{set} = \frac{d_{inter} + d_{intra}}{T_{req}} \tag{9}$$

The expected speed requirement is given by:

$$E[X_{set}] = \frac{0.3825 + \frac{1}{2(p\lambda)^{1/2}}}{T_{req}} \tag{10}$$

Let  $Q_t$  be the probability that a next-hop sensor fails to satisfy the speed requirement. Assume that the speed of a sensor is uniformly distributed within a range  $[a, b]$ .  $Q_t$  is given by:

$$Q_t = \text{cdf}(x \leq E[X_{set}]) = \frac{E[X_{set}] - a}{b - a} \tag{11}$$



“Fig.2. Hop-by-hop data delivery”

We develop a hop-by-hop data delivery scheme to implement the desired level of redundancy to achieve QoS. For path redundancy, we want to form  $m$  paths from a source cluster head to the processing center, as illustrated in Figure 2. This is achieved by having  $m$  nodes in hop one relay the data through broadcasting, and only one single node relay the data per receiving group in all subsequent hops. For source redundancy, we want each of the  $m_s$  sensors to communicate with the source cluster head through a distinct path. This is achieved by having only one node relay the data through broadcast in each of the subsequent hops in each path.

Let  $n_k$  be the average number one-hop neighbors, calculated as  $\lambda\pi r^2$ . It has been reported that the number of edge-disjoint paths between nodes is equal to the average node degree with a very high probability [6]. Thus when the density is  $n_k$  sufficiently high such that is sufficiently larger than  $m$  and  $m_s$ , this hop-by-hop data delivery scheme can effectively result in  $m$  redundant paths for path redundancy and  $m_s$  distinct paths from  $m_s$  sensors for source redundancy.

The probability of a next SN fails to relay a broadcast packet because of either channel/node failure or speed violation, denoted by  $Q_{rt}$ , is given by:

$$Q_{rt}=1-[(1 - Q_r)(1 - Q_t)] \tag{12}$$

The probability that at least one next-hop SN (among the one-hop neighbors) is able to receive the broadcast message and satisfy the speed requirement is given by:

$$\Theta=1-Q_{nk}^{rt} \tag{13}$$

Thus, the probability that at least one next-hop sensor receives the broadcast message and satisfies the speed requirement in each of subsequent hops in a path between the source cluster head and the processing center, i.e., the probability that a path is formed, is given by  $\Theta^{N_{inter}^h - 1}$

Where  $N_{inter}^h$  is the number of hops between the source cluster head and the processing center. Since there are  $m$  paths between a source cluster and the processing center, the source cluster will fail to deliver data to the processing center if one of the following happens:

- None of the SNs in the first hop receives the message. The probability for this case is  $1-\theta$ .
- In the first hop,  $j$  ( $1 \leq j \leq m$ ) SNs receives the message, but all  $j$  paths fail to deliver the message because the subsequent hops fail to receive the broadcast message. The probability for this case is:

$$\sum_{j=1}^{m-1} C \binom{nk}{j} Q_{rt}^{nk-j} (1 - Q_{rt})^j (1 - \theta^{N_{inter}^h - 1})^j$$

- In the first hop, at least  $m$  SNs receive the broadcast message from the source CH, but all  $m$  paths fail to deliver the message because the subsequent hops fail to receive the broadcast message. The probability for this case is:

$$\sum_{j=m}^{m-1} C \binom{nk}{j} Q_{rt}^{nk-j} (1 - Q_{rt})^j (1 - \theta^{N_{inter}^h - 1})^m$$

Thus, the probability of the source cluster failing to deliver data to the processing center is given by:

$$Q_{fp}^m=(1 - \theta) + \sum_{j=1}^{m-1} C \binom{nk}{j} Q_{rt}^{nk-j} (1 - Q_{rt})^j (1 - \theta^{N_{inter}^h - 1})^j + \sum_{j=m}^{m-1} C \binom{nk}{j} Q_{rt}^{nk-j} (1 - Q_{rt})^j (1 - \theta^{N_{inter}^h - 1})^m \tag{14}$$

For source redundancy, instead of using one sensor, we assign  $m_s$  sensors in each cluster to return sensor readings to their CH to cope with incorrect readings and channel/sensor faults. To implement source redundancy, SNs also use hop-by-hop data delivery scheme to send sensor data to their CH. A SN will fail to



return its reading to the CH when any hop in the path from the SN to the CH fails to forward the data, so the failure probability<sup>1</sup> that all  $m_s$  sensors within a cluster fail to return sensor reading to the CH is given by:

$$Q_{fs}^{m_s} = (1 - \theta^{N_{intra}^h - 1})^{m_s} \quad (15)$$

Combining results from above, the failure probability of a cluster not being able to return a correct response, because of either path or source failure, or both, is given by:

$$Q_f = 1 - (1 - Q_{fp}^m)(1 - Q_{fs}^{m_s}) \quad (16)$$

Therefore, the query success probability is given by:

$$R_q = 1 - Q_f \quad (17)$$

#### 4.2 Energy consumption

For source redundancy, in response to a query, a sensor assigned would transmit a data packet to their respective source CH. Since the average number of hops between a sensor and its CH is given by  $N_{intra}^h$  as derived above, and in general a query requires the use of  $m_s$  sensors for source redundancy, the total energy required to forward data to the CH is given by:

$$E_s = m_s (N_{intra}^h - 1) [E_T + \lambda(\pi r^2)E_R] \quad (18)$$

For path redundancy, let  $E_{ch}$  be the total energy consumed by the WSN to transmit sensor data from the source CH to the processing center with  $m$  paths connecting the CH to the processing center. The source CH would broadcast a copy of the data packet and all first-hop neighbors would receive. Then, among the first hop neighbors,  $m$  nodes would broadcast again and all 2nd-hop neighbors would receive. In each of the subsequent hops on a path, only one node would broadcast and the neighbors on the next-hop would receive. Consequently,  $E_{ch}$  is given by:

$$E_{ch} = E_T + [E_T + \lambda(\pi r^2)E_R] + m(N_{inter}^h - 1)[E_T + \lambda(\pi r^2)E_R] \quad (19)$$

The amount of energy spent by the system,  $E_q$ , to answer a query that demands a source cluster to respond, using  $m_s$  sensors for source redundancy and  $m$  paths for path redundancy, is given by:

$$E_s \quad E_q = E_{ch} + \quad (20)$$

### V. Performance Evaluation

Our objective is to find the best redundancy level represented by  $m$  and  $m_s$  that would satisfy query reliability and timeliness requirements as defined by equation (17) while maximizing MTTF, when given a set of system parameter values characterizing the application and network conditions. That is, if  $T_{req}$  and  $R_{req}$  are the timing and reliability requirements of a query, then we determine the best combination of  $(m, m_s)$  such that the MTTF is maximized, subject to the constraint:

$$R_q > R_{req} \quad (21)$$

Note that  $R_q$  the constraint given above implies the timing requirement is satisfied based on how we derive in equation (17). From a user's perspective, the user rarely concerns if a response to a query would satisfy condition (21) above. Even in cases a response is formulated properly by the sensor system such that condition (21) is satisfied, if the user does not see a response returned within the specified real-time constraint, the system is considered as having failed. We define a metric called the mean time to failure (MTTF) of the sensor system that considers this failure definition.

MTTF of a sensor data system as the average number of queries that the system is able to answer correctly before it fails, with the failure caused by either channel or sensor faults (such that a response is not delivered within the real-time deadline), or energy depletion. When  $m$  paths and  $m_s$  sensors are used to achieve  $R_q$  in order to satisfy condition (19), the amount of energy consumed is given by  $E_q$  given in equation (20)

above. Thus, the average number of queries that the system is able to sustain before running out its energy is given by:

$$N_q = \frac{E_{\text{initial}} - E_{\text{threshold}}}{E_q} \tag{22}$$

Since the system is able to answer  $N_q$  queries before energy depletion, each with the reliability of  $R_q$ , the MTTF of the system is the expected number of queries that the system can answer without experiencing a failure with the upper bound of  $N_q$ , i.e.,

$$\text{MTTF} = \sum_{i=1}^{N_q-1} i R_q^i (1 - R_q) + N_q R_q^{N_q} \tag{23}$$

## VI. SIMULATION

We present numeric data to demonstrate the tradeoff between  $R_q$  and  $E_q$  and that there exists an optimal  $(m, m_s)$  set that would maximize the MTTF of the sensor system while satisfying Condition (21). Table 1 lists the parameters used along with their default parameters. We vary key parameters to study their effect on optimal  $(m, m_s)$  and MTTF.

Parameter	Default value
M	[1-4]
$m_s$	[1-7]
N	4000
$n_s$	300
Q	$10^{-6}$
E	[0.3-0.6]
R	1unit (1 unit=25m)
(a,b)	(10,100)
$\lambda$	10 nodes/sq.unit
A	20 units
$n_b$	50 bytes
$E_{elec}$	50 nJ/bit
$E_{amp}$	10Pj/bit/ $m^2$
$E_0$	2 joule
$E_{threshold}$	0 joule
$T_{req}$	[0.2-0.5]sec

“Table 1: parameters”

Table 2 summarizes the optimal  $(m, m_s)$  set that would maximize the MTTF of the sensor system under the environment characterized by the set of parameter values listed in table 1. Other parameter values may generate different  $(m, m_s)$  but the trend remains the same. We see that as  $e$  increases, the system tends to use more redundancy to satisfy condition (21) and to maximize the MTTF of the sensor system. Conversely as the real-time deadline increases, the system tends to allocate less redundancy. Most importantly, there always exists an optimal  $(m, m_s)$  set that would maximize the MTTF of the sensor system

	e=0.35	0.4	0.45	0.5	0.55
$T_{req}=0.25\text{sec}$	2,3	3,3	3,3	4,7	4,7
0.30 sec	1,1	2,3	2,3	3,3	4,7
0.35 sec	1,1,	1,1	2,3	2,3	3,3
0.40 sec	1,1	1,1	2,1	2,3	2,3
0.45 sec	1,1	1,1	1,1	2,3	2,3

“Table 2: Optimal  $(m, m_s)$  with varying  $e$  and  $T_{req}$ ”

## VII. Conclusion

In this paper described path and source redundancy fault tolerance mechanisms, which, when properly employed, could achieve quality of service (QoS) requirements while maximizing the lifetime of query-based sensor networks. We discussed how these mechanisms can be realized using hop-by-hop packet broadcasting and derived the probability of successful data delivery within a real-time constraint  $R_q$  (as well as the amount of energy consumed  $E_q$  per query. When given a set of parameter values characterizing the operating and workload conditions of the environment, we identified the optimal  $(m, m_s)$  setting that would maximize the MTTF while satisfying the application QoS requirements. To apply the results derived in this paper, one could build a table at static time listing  $R_q$ ,  $E_q$  and MTTF as a function of  $(m, m_s)$  covering a perceivable set of parameter values, as well as per-query QoS requirements in terms  $R_{req}$  and  $T_{req}$ . Then, a simple table lookup could be performed at runtime to determine the optimal  $(m, m_s)$  that could satisfy the QoS requirements and maximize the MTTF.

In the future, we plan to look at the implementation issues. We also plan to extend the work to the case when there are multiple QoS levels from multiple query classes that may query the sensor network concurrently. The analysis work performed in this paper is based on the average behavior without considering dynamics such as more energy may be consumed by some sensors over others or some sensors may fail earlier than others. In the future, we plan to factor network dynamics into the model, develop fault tolerance mechanisms and algorithms associated with them, and analyze the effect of  $(m, m_s)$  on the MTTF of the sensor networks.

## References

- [1] O. Younis and S. Fahmy, "HEED: A Hybrid Energy Efficient, Distributed Clustering Approach for Ad Hoc Sensor Network", IEEE Transaction on Mobile Computing, Vol. 3, No. 3, October-December 2004, pp. 366-379.
- [2] W. Heinzelman, C. Chandrakasan and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Micro sensor Networks," IEEE Transactions on Wireless communication, Vol. 1, No. 4, 2002, pp. 660-670.
- [3] P. Mhatre, et al. "A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint," IEEE Transactions on Mobile Computing, Vol. 4, No. 1, 2005, pp. 4-15.
- [4] D. Chen, and P. Varshney, "QoS Support in Wireless Sensor Networks: A Survey", International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, USA, June 21-24, 2004
- [5] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocol for Self-Organization of a wireless Sensor Network", IEEE Personal Communications, pp. 16-27, October 2000
- [6] B. Deb, S. Bhatnagar and B. Nath, "ReInForM: Reliable Information Forwarding using Multiple Paths in Sensor Networks", 28th Annual IEEE Conference on Local Computer Networks (LCN 2003), Bonn, Germany, October 2003
- [7] M. Perilo, and W. Heinzelman, "Providing Application QoS through Intelligent Sensor Management", Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA 03), May 2003.
- [8] E. Felemban, C. Lee, E. Ekici, R. Boder, and S. Vural, "Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks", Proceedings of IEEE INFOCOM 2005, vol. 4, March 2005, pp. 2646-2657.
- [9] R. Iyer, and L. Kleinrock, "QoS Control for Sensor Network", IEEE International Conference on Communications (ICC 2003), May 2003