Optimizing Communication Energy Consumption in Perpetual Wireless Nanosensor Networks

Shahram Mohrehkesh and Michele C. Weigle
Department of Computer Science
Old Dominion University
{smohrehk, mweigle}@cs.odu.edu

***Abstract*—This paper investigates the effect of various parameters of energy consumption for communication in pulse-based wireless nanosensor networks that exploit energy harvesting to supply energy. Finding the optimum combination of parameters to minimize energy consumption while satisfying the QoS requirements (e.g. delay and reliability) of communication is a challenging task. We model this problem as a multiobjective function problem. We evaluate the effect of packet size, repetition and code weight on this optimization problem. Through simulation, the effect of network parameters, i.e. topology and energy for pulse transmission/reception, on the optimization problem is studied as well. The model enables optimum energy consumption design in wireless nanosensor networks.**

I. Introduction

Wireless nanosensor networks (WNSNs) [1] are a new generation of sensor networks at nano scale, which are envisioned to be produced in the coming years. Each nanonode is in the range of micro to nano meters in size. The nanoscale property of nanonodes opens exciting new applications in the sensing domain. For example, nanosensors could detect chemical compounds at the molecular level or the presence of different infectious agents, such as viruses or harmful bacteria [1]. Similar to sensor networks, they can be used in many fields, such as biology, medicine, chemistry, and in environmental, military, industrial, and consumer goods [1]. For example, nanosensors could be used to develop new touch surfaces or be added to standard office products (pens, papers, etc.), making the idea of a smart office a reality.

Nanosensors will collect useful information that must be sent outside of their sensing environment for storage and additional processing. In other words, they will need to communicate between themselves as well as with nodes in the micro and macro domain. Among all possible communication methods among nanonodes, studies [2] show that electromagnetic communication in the 0.1-10.0 terahertz (THz) frequency band is a promising approach for communication in WNSNs. Operating in the THz band can help nanosensors consume low energy while providing connectivity at the nano scale. A graphene-based nano-antenna has been studied [3] as the nano scale transceiver to transmit and receive pulses in the THz band.

Power has been a major challenge in wireless networks, as the lack of energy makes communication impossible. In WNSNs, with a nano battery, the problem is even more significant. It is not possible to provide nanoscale batteries that can last even for a month. Thus, energy harvesting through vibrations has been proposed [1], [4] as the method for providing power.

An initial model for the energy harvesting and consumption processes in WNSNs has been proposed by Jornet and Akyildiz [4]. However, their model does not evaluate the effect of various parameters to find the optimal performance. To be specific, several parameters are introduced in the modeling of energy harvesting and consumption that can affect the optimum energy utilization of a nanonode. Finding the optimum values for code weight, repetition (defined in Section II) and packet size, is an open question. Jornet and Akyildiz [4] argue that using code weight can save energy in transmission since the lower the code weight, the lower the energy for transmission. However, a deeper look shows that it depends on the energy ratio of reception to transmission of pulses as well as topology (i.e. number of neighbors). Moreover, the optimum packet size is dependent on code weight. Finally, the effect of code weight and repetition on communication reliability in combination with energy consumption needs to be considered. This paper attempts to find the optimum combination of these variables through solving an optimization problem.

Our optimization model for minimizing energy consumption is different from previous works in several major aspects. First, while previous work focused on methods such as minimizing the energy consumption through adapting the modulation [5], efficient routing protocols [6], or adaptive duty cycle of MAC [7], our method focuses on packet size and code weight for optimization. Since most previous work was developed based on carrier based transmission, code weight optimization as well as pulse-based communication optimization has not been investigated before. Second, our model introduces the concept of optimizing several communication objective functions simultaneously, while previous work mainly focus on one or two objective functions. Even models with several objective functions use methods such as weighted sum which are not efficient or accurate methods as comparison to the GA method that we use here. Finally, our model takes into account the communication energy consumption for both transmitter and receiver, where transmitter traffic interactions with several neighbors as receivers is considered.

The rest of paper is organized as follows. In the next section, we introduce the background for the communication model in WNSNs. In Section III we introduce our optimization problem. Simulation results are presented in Section IV, and the paper is concluded in Section V.

The work has been supported partially by NSF grant CNS-1116238.

II. Background and Related Work

In this section, we briefly introduce the communication model and energy consumption for communication in the THz pulse-based communication for nanosensors, mainly developed by Jornet and Akyildiz [1], [4].

A. *Communication Model*

Studies [2] have shown that molecular communication and electromagnetic (EM) communication in the 0.1-10.0 terahertz (THz) band are potential approaches for communication in WNSNs. Since EM communication provides a higher data rate [2] than molecular communication, in this work, we develop our communication model for EM communication only. Operating in the THz frequency allows nanosensors to consume low energy while providing connectivity in the nano to milli meter communication range. The nodes use pulse-based communication and Rate Division Time Spread On-Off Keying (RD TS-OOK) [8] as the modulation mechanism. A logical 1 is transmitted as a femto-second long pulse, and a logical 0 is transmitted as silence. The duration of each pulse is T_p and the time between two symbols, called inter-symbol time, is T_s. The probability of collision between symbols is extremely low due to the fact that there can be no collisions for 0 symbols (silences) and that the length of T_s is much longer than T_p (typically 1000 times larger).

Even though collisions are not much of an issue, there may be packet loss due to pulse absorption. To mitigate the effects of absorption, symbol repetition has been proposed [9]. Repetition is a simple mechanism for error detection and correction. With this method, the sender simply repeats each symbol several, typically 1 to 9, times.

As 0 symbols take no energy to transmit (because they are silences), it would be most energy-efficient to send as many 0s as possible. Applying a code weight to a packet can result in reducing the number of expensive 1s that are transmitted. The code weight is defined as the proportion of 1s to the total number of 1s and 0s in the packets [9]. A code weight of 0.5 means that, on average, there are an equal number of 1s and 0s in the packets. A lower weight, such as 0.4, means that there are fewer 1s. However, it also means that more bits should be used to send the same amount of information. For example, Table I shows how the number of 1s for sending two bits of information could be reduced by using three bits.

TABLE I. Code Weight Example

value	2 bits (weight = 0.5)	3 bits (weight = 0.25)
0	00	000
1	01	001
2	10	010
3	11	100

For a more realistic example, for sending $n = 64$ bits of information with a code weight of 0.4, at least $a = 6$ more bits will be added to each packet. In this case, the total number of encoded bits would be $m = 70$ and the number of 1s, u, is less than or equal to 28.

To make sure that for a target code weight, there are at most u 1s independent of the original bit values, for n-bits of information, the $\frac{m!}{(m-u)!u!} \geq 2^n$ condition must be satisfied with the minimum a additional bits, where the total number of bits $m = n + a$.

Note that sending fewer 1s consumes less energy in the sender while it consumes more energy in the receiver. Energy is consumed when receiving any bit, 0s or 1s. Decreasing the code weight necessarily increases the packet size, increasing the cost to the receiver. Depending on the ratio of energy required for reception to that transmission of a pulse, named as $\alpha = \frac{E_{pulse-rx}}{E_{pulse-tx}}$, and packet length, the code weight may or may not save energy in total. Here, the assumption is that α is small; therefore, the aim is to find the optimum values for *packet length* and *code weight*, which we address in Section III.

B. *Communication Energy Consumption Model*

Considering the code weight and energy for pulses, the energy required for transmission and reception of a packet can be computed. For a packet of size N, the energy consumed when transmitting or receiving a packet are given by

$$E_{packet-tx} = N \cdot W \cdot E_{pulse-tx} \tag{1}$$

$$E_{packet-rx} = N \cdot E_{pulse-rx} \tag{2}$$

where W represents code weight. Note that we assume that the energy for listening or idle mode of the wireless module of nanonode is not significant. Also, we do not include energy for other modules of the nanonode such as nano-processor, nano-memory, etc.

We assume that there is a grid of nanonodes, where each node has G neighbors. Each node senses its environment, creates a packet and broadcasts it. All neighbor nodes will receive the packet and, based on policies such as a random process, relative location, or the signal strength of received packet [10] and [11], one or several of neighbors will forward the packet. This step is repeated until the packet is delivered to the targeted destination, which is typically a sink for collecting the sensed information, or until the packet travels more than a defined maximum number of hops. Independent of the forwarding policy of traffic, the main parameter that affects the amount of energy for reception will be the number of neighbors. Our optimization problem will include this parameter in combination with other parameters.

III. Optimizing the Factors of Energy Consumption

Based on the joint model of energy harvesting and consumption, developed by Jornet and Akyildiz [1], [4], various factors are identified that can affect energy harvesting and consumption, and consequently network performance (e.g. delay, probability of successful transmission). Particularly, packet size, code weight, and repetition can affect the amount of energy that is consumed. We also must find the optimal achievable network performance in terms of various QoS requirements (e.g. delay, transmission reliability). Therefore, finding the optimum design point between energy usage efficiency and the network performance is the challenge here. We develop an optimization model that can find the best combination of these parameters. Then, we show how the best answer could be selected among a list of candidates when traffic load is taken into account.

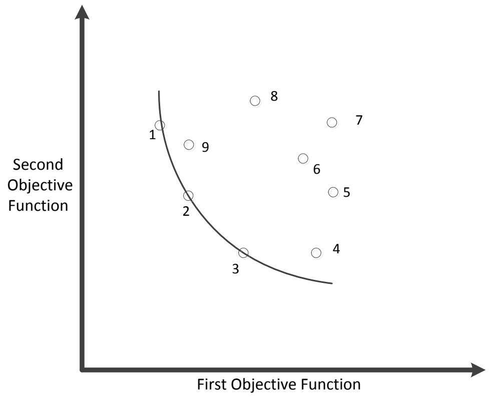

Fig. 1. Pareto points among other solution points for a two objective function problem

A. Optimization Model

We model the problem as a *Multi-Objective Combinatorial Optimization* (MOCO), a special form of Multi-Objective Optimization (MOP) [12], where variables can take discrete values. The general form of a MOCO is

$$\begin{aligned}&min_x\,[f_1(x), f_2(x), \ldots, f_n(x)]^T\\&\text{s.t.}\\&g(x) \leq 0\\&h(x) = 0\\&x_l \leq x \leq x_u\end{aligned}$$

where f_i is the i-th objective function, g and h are the inequality and equality constraints, respectively, and x is the vector of optimization or decision variables. The solution to the above problem is a set of Pareto points. In a MOP/MOCO problem, several functions need to be optimized at the same time. Then, instead of having a unique solution to the problem, the solution is a possibly infinite set of Pareto points. These points are called *Pareto optimal*. A design point in objective space f is termed *Pareto optimal* if there does not exist another feasible design objective vector f^* such that $f_i \leq f_i^*$ for all $i \in \{1, 2, \ldots, n\}$, and $f_j < f_j^*$ for at least one index of j ,$j \in \{1, 2, \ldots, n\}$.

Figure 1 represents the Pareto points among all possible solutions for a two objective function problem that requires the minimization of both objective functions. The curved line represents the Pareto-front, identified by non-dominated solutions that have labels 1, 2 and 3. The other points are not optimal Pareto points because they have a higher value in at least one of objective functions. For example, point 4 has higher value than point 3 for first objective function, and point 9 has a higher value than point 2 for second objective function. Clearly, representing and finding the solution of a MOCO problem when the number of variables, number of objective functions, and search space is larger is more challenging.

In our problem, the functions to be optimized for the N bit packet size, repetition R, and code weight W variables are defined as follows.

The first function is energy consumption, that is, the energy consumed for transmission (Eq. 1) plus reception (Eq. 2) of a packet by all the neighbors with N bit data.

$$f_1 = \frac{E_{packet-tx} + G \cdot E_{packet-rx}}{N} =$$
$$m' \cdot W \cdot E_{pulse-tx} + G \cdot m' \cdot \alpha \cdot E_{pulse-tx}$$

where $m' = N + a$, and a is the number of additional bits added to N that enables coding with code weight W. Recall that G is number of neighbors and α is the ratio of the energy required for reception to that of transmission of a pulse. We developed the model in the general form where there are G neighbors. Therefore, it will cover most unicast or broadcast scenarios where the packet will be received by one, some, or all of the neighbors. This function is set to be minimized, which means that the total energy that is consumed for transmission and reception per bit of information should be minimized.

The next objective function, f_2, concerns delay. Since N is larger than the information generation rate, the packet would contain several pieces of information together to avoid the overhead of packet transmission. However, this increases the delay in transmission of information. For example, if information is generated at 10 bits per second and the packet size is 1000 bits, it will take 100 seconds to prepare a packet. This may be acceptable for non-real time applications, or when the rest of packet can be filled with neighbors' forwarding data or can just be left empty. However, in our model, we are assuming that packets only contain information generated from one node. The simplest way to define the delay function is to model it in a linear relation with packet length, N. However, if delay has higher importance, the function could be modeled as a higher degree polynomial function of N. This function is set to be minimized.

$$f_2 = N \tag{3}$$

The next objective function, f_3, associates the chance of bit error rate with code weight. Lower code weight means transmission of fewer `1`s, which results in a lower probability of absorption as well as collisions between `1`s. This function is set to be minimized.

$$f_3 = W \tag{4}$$

The optimization problem can be formulated with only the $f1$, $f2$ and $f3$ functions, if repetition is not required to be considered as a variable. This could be the case if it is known that the environment would not affect the pulses significantly, and it is better to repeat the whole packet in case of error rather than consume energy with the repetition of symbols. However, we define the functions for repetition as a variable to have a comprehensive model.

The following function shows the effect of repetition. The higher the repetition, the higher the chance of error detection and recovery. Note that providing reliability depends on other parameters such as code weight, transmission power, and inter-symbol times. Here we are assuming that inter-symbol times and transmission power cannot be varied. Also, the effect of code weight on reliability is not significant as compared to repetition. Moreover, a lower code weight would result in reliable communication, which was considered in f_3.

$$f_4 = \lfloor \frac{\frac{R-1}{2}}{R} \rfloor \tag{5}$$

On the other hand, lower repetition means fewer bits to transmit and, consequently, less energy consumption.

$$f_5 = \frac{N}{R} \tag{6}$$

This function actually shows the efficient bit rate when repetition is used, and it should be maximized.

The constraint functions would be

$$\begin{aligned} g_1 &= m' \cdot W \cdot E_{pulse-tx} - E_{max} \leq 0 \\ g_2 &= m' \cdot E_{pulse-rx} - E_{max} \leq 0 \end{aligned}$$

This means that the energy for transmission or reception of one packet cannot exceed the maximum energy capacity of the nanonode, E_{max}.

The bounds on the variables of the problem are

$$\begin{aligned} &W \in (0.15 : 0.05 : 0.5) \\ &R \in (1, 3, 5) \\ &1 \leq N \leq 2500 \end{aligned}$$

Note that because the problem is a combinatorial problem, the bounds are actually the set of valid values that can be assigned to variables, i.e. W and R. For N, in addition to the bounds, the values should be discrete.

B. Optimization Problem Solution

Various methods are used to solve multiobjective optimization problems, such as the method of objective weighting or min/max formulation [13] , [12]. In some specific problems, it is possible to merge multiple objectives into one objective so that the resulting solution depends mainly on the weight vector assigned to each objective [13]. As a result, the same problem must be solved several times for different weight vectors.

Another way to solve multiobjective optimization problems is to use Genetic Algorithms (GA). Since GAs search for the optimal solutions based on a population of points instead of a single point, they can find multiple Pareto optimal solutions in a single run. It helps decision makers to choose the best solution from set of Pareto optimal points based on the situation without involving them in burden and common difficulty of balancing objective functions.

We use the controlled Non-dominated Sorting Genetic Algorithms II (NSGA-II) [13] to solve our MOCO problem. As mentioned before, the output of MOCO is a set of Pareto optimal points. Typically, the selection of one point depends on the context of the application.

IV. Simulation

We solved our defined MOCO problem with the optimization toolbox of MATLAB. We customized the creation, mutation, and crossover functions of model. Simulation parameters for our MOCO solution with the NSGA-II method are shown in Table II. We run the optimization with different values for α, G and $repetition$ to show the effect of these parameters on the points that are selected as optimum. The results for each of the configuration scenarios, listed in Table III, are presented in the following subsections. Note that Pareto optimal points are not unique and can even be different in several runs. However, the results that are presented here have a similar pattern for all runs and different runs give only non-significant bit differences in packet size.

TABLE II. MOCO Problem Parameters

Parameter	Value(s)
Population size	100
Pareto fraction	0.2
Generations	150
Selection	Uniform
Crossover fraction	0.8
Mutation function	Uniform
Crossover function	Three parents

TABLE III. Scenario Parameters

Scenario	G	α	max R
1	1	0.1	1
2	1	0.5	1
3	4	0.1	1
4	1	0.1	5
5	4	0.1	5

A. Scenario 1 (G=1, α=0.1, R=1)

In this scenario, we set G to 1. This scenario will evaluate the case of transmission between two adjacent nodes when broadcast will result only in reception by one neighbor. The α value is set to 0.1, based on the numerical values in [8] and modeling in [4], which is 1 pJ and 0.1 pJ for transmission and reception of pulse in 10 mm distance. Figure 2 shows the Pareto optimal points that are selected. The code weight and the packet length for each of the points is presented in the legend. This scatter plot represents the value of first and second objective functions for each of the Pareto points. Recall that the first objective function tries to minimize the amount of consumed energy per bit. On the other hand, the second function, minimizing delay is in relation to the packet length. Each of these points dominates the other one in one of the two objective functions. Therefore, depending on design priority, any of these points can be selected as the optimal solution. For example, if the priority is energy consumption, one of the points in the lower-right could be selected. If delay has priority, one of points in the left side would be the choice.

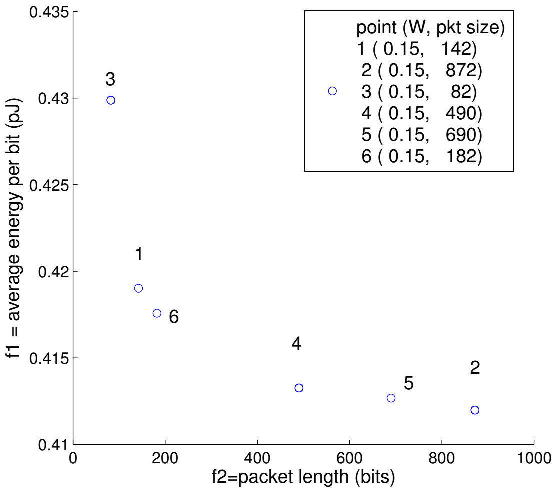

Fig. 2. Pareto Point and Function Values for Scenario 1

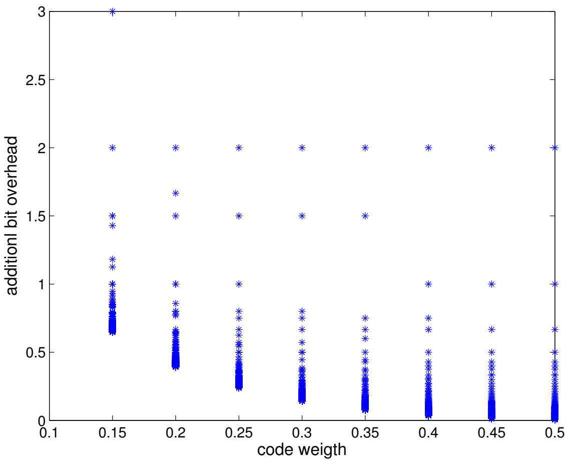

Fig. 3. Additional bit overhead for various code weights

Figure 2 also illustrates that various packet lengths are selected. A deeper look at the selected code weight for these points shows that all of them are equal to 0.15, which is the minimum code weight. It means that with this setting for G and α, it is better to choose the minimum code weight that is available. In fact, when one of the objective functions will have only optimum values for a specific value of variables, here W, then the problem focuses on the limited optimization of the other objective functions, in this case, packet length.

Figure 2 also shows that the difference in terms of efficient energy per actual information bit, f_1, is not significantly different among all the optimal points. This observation is confirmed in Figure 3, which shows that the overhead ($\frac{a}{N}$) from the code weight generally does not depend on the length of data. The figure illustrates data lengths that are in $[1..1000]$ range. Outliers occur when the number of original bits are very small, i.e. less than 10 bits. These short packets lengths are not applicable.

B. Scenario 2 - Effect of α

In this scenario, we increase the value of α to 0.5. Recall that α is $\frac{E_{pulse-rx}}{E_{pulse-tx}}$. Note that in reality, α is fixed. Here we are only evaluating scenarios for different α to show its effect on the optimization problem. The selected optimum points now cover a wide range of various code weights and packet sizes, as shown in Figure 4.

This easily shows that code weight is more effective for smaller values of α. In fact, when α becomes larger, the effect of code weight is reduced. This happens because the overhead bits from the lower code weight increase the reception energy, which eventually increases the average energy per bit. Because there are more than two objective functions, the optimal points cannot be chosen from the figure. Otherwise, if there were only these two functions, the optimum point would be one of the points in the bottom-left of Figure 4.

C. Scenario 3 - Effect of G

In this scenario, the effect of G on the selection of optimal points can be viewed in Figure 5, when G is changed from 1 to 4. α is set to 0.1 as described in Table III. The main observation is that more points with lower code weights are selected since there are more recipients, which makes it efficient to use a lower code weight. This effect can be viewed also in the average energy per bit function, $f1$, where it is almost twice that of Scenario 1 (with one neighbor) while the number of neighbors has increased four times.

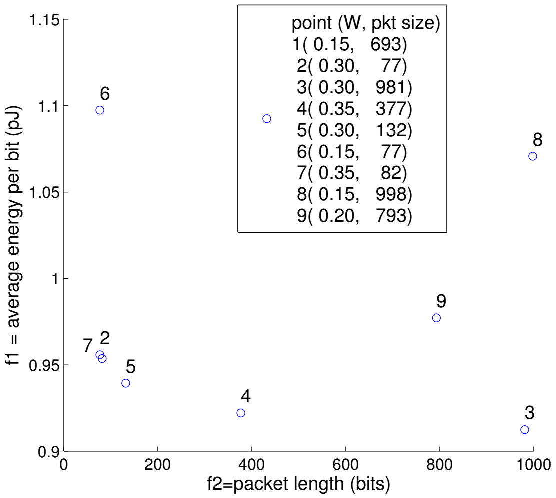

Fig. 4. Pareto Point and Function Values for Scenario 2

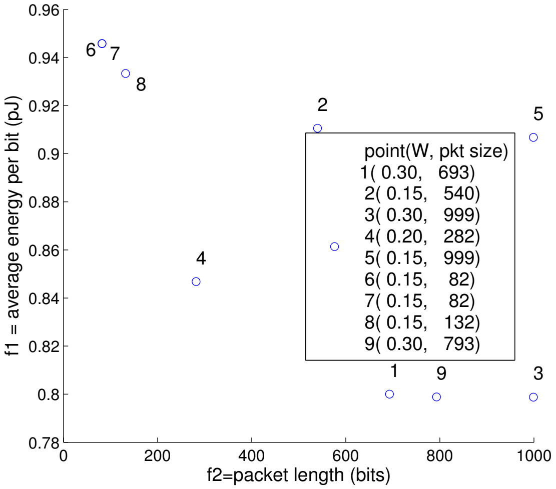

Fig. 5. Pareto Point and Function Values for Scenario 3

D. Scenario 4 - Effect of Repetition

This scenario takes into account the effect of repetition as another variable. 3 and 5-repetition is possible in this scenario. As shown in Table III, this scenario is similar to Scenario 1, as G and α are set to 1 and 0.1 respectively. Figure 6 shows the selection of optimum points. Points with various ranges of values for repetition, code weight and packet length are selected. This behavior is mainly due to the dominance of one of the objective functions. For example, point number 3 is selected because it provides a low average energy per bit. On the other hand, point 7 is selected because it provides high reliability with 5 repetition even though it has higher delay, i.e. packet length, and higher average energy per bit in comparison to the other points.

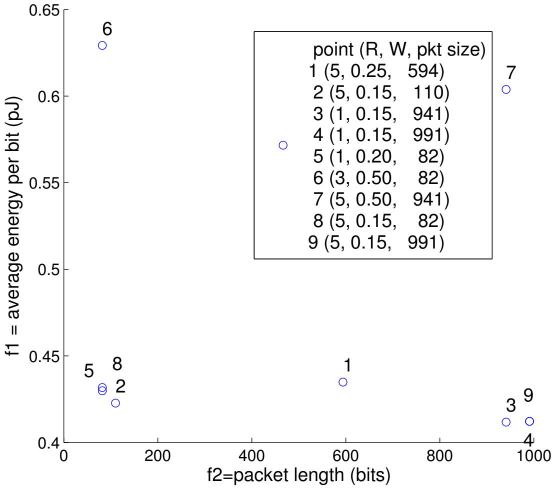

Fig. 6. Pareto Point and Function Values for Scenario 4

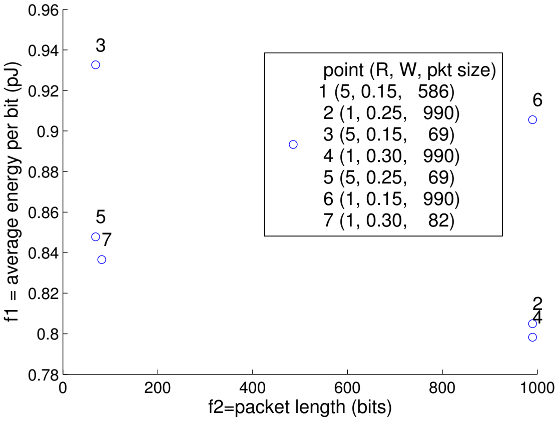

Fig. 7. Pareto Point and Function Values for Scenario 5

E. Scenario 5 - Repetition for Higher G

The fifth scenario evaluates the effect of repetition in combination with a higher number of neighbors (from one to four). As indicated in Table III, the maximum repetition and α are set to 5 and 0.1 respectively. In this scenario, the optimal points, as illustrated in Figure 7, are selected from almost all ranges of code weight and repetition. However, packet length values are mainly chosen from very short or very large packet sizes. The reason is that when a short packet size is selected, the energy bit efficiency and delay will be the dominant functions. On the other hand, for large packet sizes, code weight will be the dominant factor that leads to lower average energy per bit.

F. Selection Based on Performance

The final choice of optimum point will depend on the traffic load, i.e. packet generation rate from sensing. Using a model such as one developed by Jornet and Akyildiz [4], the various network performance metrics (e.g. delay, probability of success transmission) could be computed and the optimum point that satisfies the QoS requirement would be selected. For instance, the probability of success is affected when the node is out of energy due to high traffic load. In this case, even for an optimal point, the performance can be unacceptable if load is very high. So, a performance evaluation model in combination with this optimization model can find the optimum operation condition of a nanonode. Likewise, when the nanonode is out of energy, it takes time to harvest energy to transmit a packet. Therefore, if the traffic load is very high, it will result in a long delay due to lack of energy for the transmission of packets. These evaluations are part of our future work.

V. Conclusion

This paper investigated simultaneous optimization of energy consumption and the QoS requirements of perpetual wireless nanosensor networks. Code weight and repetition as parameters to reduce energy consumption and increase transmission reliability were studied in combination with packet size. The goal is to provide optimum energy consumption while the QoS requirements, i.e. delay and transmission reliability, are considered. The effect of energy for reception/transmission of pulse and network topology is shown in the model. The optimized model provide a guideline for optimal design of energy harvesting wireless nanosensor networks.

References

[1] I. F. Akyildiz and J. M. Jornet, "Electromagnetic wireless nanosensor networks," *Nano Comm. Networks*, vol. 1, no. 1, pp. 3–19, 2010.

[2] I. F. Akyildiz, F. Brunetti, and C. Blazquez, "Nanonetworks: A new communication paradigm," *Computer Networks*, vol. 52, no. 12, pp. 2260–2279, 2008.

[3] J. M. Jornet and I. F. Akyildiz, "Graphene-based nano-antennas for electromagnetic nanocommunications in the terahertz band," in *Proc. of EUCAP 2010, Fourth European Conference on Antennas and Propagation*, April 2010.

[4] J. Jornet and I. Akyildiz, "Joint energy harvesting and communication analysis for perpetual wireless nanosensor networks in the terahertz band," *IEEE Trans. on Nanotechnology*, vol. 11, pp. 570–580, May 2012.

[5] S. Cui, A. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *Wireless Communications, IEEE Transactions on*, vol. 4, no. 5, pp. 2349–2360, 2005.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pp. 10 pp. vol.2–, 2000.

[7] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of ACM SenSys*, pp. 171–180, 2003.

[8] J. M. Jornet and I. F. Akyildiz, "Information capacity of pulse-based wireless nanosensor networks," in *IEEE SECON*, pp. 80–88, 2011.

[9] J. Jornet and I. Akyildiz, "Low-weight channel coding for interference mitigation in electromagnetic nanonetworks in the terahertz band," in *IEEE Intl. Conference on Comm. (ICC)*, pp. 1 –6, Jun 2011.

[10] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of ACM MobiCom*, pp. 151–162, 1999.

[11] N. Wisitpongphan, O. Tonguz, J. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Comm.*, vol. 14, pp. 84–94, Dec. 2007.

[12] K. Deb, "Multi-objective optimization," in *Search Methodologies* (E. Burke and G. Kendall, eds.), pp. 273–316, Springer US, 2005.

[13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *International Conference on Parallel Problem Solving from Nature (PPSN)*, pp. 849–858, 2000.