Optimal Power Allocation in Wireless Networks with Transmitter-Receiver Power Tradeoffs

Sudarshan Vasudevan[†], Chun Zhang[†], Dennis Goeckel[‡], Don Towsley[†]

[†]Dept. of Computer Science,
University of Massachusetts,
Amherst, MA 01003
{svasu,czhang,towsley}@cs.umass.edu

[‡]Dept. of Electrical and Computer Engg.
University of Massachusetts,
Amherst, MA 01003
goeckel@ecs.umass.edu

***Abstract*— For many wireless communication links, such as those employing turbo codes or sequentially-decoded convolutional codes, the power consumption of the decoder at the receiver depends on the received signal power and, hence, on the transmitted signal power. By transmitting a signal at a power higher than the minimum required for successful reception, the transmitter can "assist" the receiver by reducing the receiver decoding effort and thereby reducing the overall power consumption on the link.**

In this paper, we consider sender-receiver power tradeoffs in data-gathering trees. We formulate an optimization problem to optimally assign power to the nodes in the tree for maximizing the lifetime of the data-gathering tree, which is equivalent to the time until network partition due to battery outage. We propose a *Binary Search Algorithm* for optimal power assignment among nodes that maximizes the tree lifetime. Our *Binary Search Algorithm* can be easily extended to handle practical considerations such as node mobility and peak power constraints at nodes. Using turbo codes as an example of a channel coding technique, we demonstrate significant improvements in network lifetimes as a result of sender-receiver power tradeoffs. These improvements are observed under a wide variety of network conditions and are more pronounced in densely deployed networks and networks with asymmetric power costs.

I. INTRODUCTION

Wireless ad hoc networks have attracted considerable attention in the recent years, both in academia as well as industry. One of the main reasons for their popularity can be attributed to the various applications they enable. Habitat monitoring [13], environmental observation and forecasting [1], organ monitoring and health monitoring [17] and target tracking [23] are just a few examples of the many applications of wireless ad hoc networks. A wireless ad hoc network comprises a large number of nodes scattered in a region of interest. Each node is equipped with a transceiver used for transmitting and receiving signals. Each node also has some on board memory for data storage. The deployed nodes not only originate data but can also act as data forwarders, storing and forwarding data originated by other nodes.

One of the most important challenges in the design of such ad hoc networks is to reduce the energy consumption of the network. The nodes in an ad hoc network are typically battery-powered and, hence, have a limited lifetime. As a result, the lifetime of the network is also limited. Careful design to maximize the network lifetime is of utmost importance. For many ad hoc network applications, techniques such as aggregation and in-network processing [9], energy-aware routing [8], [10] and energy-aware medium access protocols [22] have been proposed in order to reduce communication. All these works assume that communication is the dominant contributing factor in limiting the lifetime of an ad hoc network. Furthermore, there have been works [7], [21], [18] that focus on the physical aspects of communication where the emphasis is on minimizing transmission energy.

An interesting observation was made in [4] in which the authors argue that, in dense ad hoc networks, the average inter-node distances will be small ($< 10m$). For small distances, the circuit energy consumption along the signal path becomes comparable to or even dominates the transmission energy in the total energy consumption. The authors then propose an energy consumption model that accounts for the circuit power consumption together with the transmission energy. Another work with a similar flavor is [14], in which the authors account for the energy expended by other processes that run when the transmitter is in the 'on' state and then find the optimal transmission strategies under this energy model.

In [19], we propose the idea of increasing transmit power to reduce the receiver power consumption. In particular, we observe that there are many coding techniques in which the power consumption at the receiver to decode a signal is a function of the transmitter power. Hence, by increasing the power used by a transmitter to transmit a signal, the decoder can decode the signal faster and expend less energy. Examples include turbo decoders, where the number of decoder iterations decreases as the received SNR increases and sequential decoding of convolutional codes. We then show that such a tradeoff has the potential for significant energy savings and hence, an improvement in network lifetime. However, several simplifying assumptions were made in [19]. In particular, we consider a homogeneous tree network, i.e. a complete, balanced tree with fixed length edges. Node data rates and remaining energies were same for all the nodes. Furthermore, all nodes were constrained to use a single power setting, which as we will see in this paper, does not always yield the optimal system lifetime.

In this paper, we consider the *Lifetime Maximization Problem* (LMP) in an arbitrary data-gathering tree of wireless nodes. Each node in the tree generates data at a constant rate, destined to a sink node. The inter-node distances are assumed to be arbitrary. We first consider a simpler version of LMP, in which nodes are constrained to use a single power setting throughout their lifetime. We then present the *Binary Search Algorithm* that achieves a power allocation among nodes to maximize the system lifetime. Subsequently, we show that a single power setting per node is insufficient to maximize the system lifetime. We then formulate a variant of LMP in which we allow multiple power settings for each node. We show how the *Binary Search Algorithm* can be modified to optimally solve the multi-power variant of LMP. Our algorithm is amenable for implementation in practical scenarios. Furthermore, the algorithm can be easily extended to adapt to changes in the tree structure induced by node mobility. Other relaxations such as peak power constraints at nodes and arbitrary node data rates can also be handled. Using turbo codes as an example, we demonstrate that by employing the power tradeoff techniques suggested in this paper, significant improvements in network lifetimes are achievable under a wide range of operating conditions, over schemes in which nodes always transmit at minimum power. In particular, we observe that the improvements in lifetime are more pronounced in dense networks and networks with asymmetric power costs.

We note that the problem of lifetime maximization of an ad hoc wireless network has been investigated earlier [2], [16]. However, the tradeoff between transmit and receive powers was not considered in these problem formulations. This tradeoff is the main subject of this paper and is considered in detail here.

The rest of the paper is organized as follows. In Section II, we describe our power model. In Section III, we describe the *Lifetime Maximization Problem* which is the subject of this paper. We first formulate a simpler version of this problem in which nodes are constrained to use only a single power setting and then solve this problem optimally. In Section IV, we show that under certain conditions a single power setting per node does not yield the optimal system lifetime. We, therefore, formulate the *Multi-Power Lifetime Maximization Problem* which allows nodes to employ multiple power settings and describe how this problem can be solved optimally. In Section V, we discuss numerical results to demonstrate the benefits of power tradeoffs, using turbo codes as an example of a communication system with transmitter-receiver power tradeoff. We discuss some practical considerations in Section VI. Finally, we conclude and present some potential future directions in Section VII.

II. Power Consumption Model

Let E_m denote the minimum received energy (in Joules) required to successfully decode an information bit conveyed by a transmitter. Now, if the transmitter and receiver are separated by a distance d, the minimum transmission energy required to convey an information bit is $E_m d^\alpha$, where α denotes the path-loss exponent. However, as noted in the introduction, often it is advantageous to transmit at an energy (or at a power level) larger than this required minimum and, thus, the actual transmission energy invested in the signal is scaled by the factor P_g ($P_g \geq 1$) to yield a total transmitter energy per information bit of:

$$P_t = P_g E_m d^\alpha \tag{1}$$

The circuit energy per bit consumed by the transmitter contributes a constant additive term to equation (1). However, we do not consider the circuit energy consumption of the transmitter in our model since it does not contribute to the problem of optimal transmit-receive power tradeoff considered in the paper.

The receiver energy consumption P_r per information bit is given by

$$P_r = E_u f(P_g) \tag{2}$$

where E_u is the amount of energy per unit operation per bit to run the decoder, and $f(P_g)$ is a (non-linear) function of P_g that returns the number of unit operations required at the receiver to decode the signal. For example, in the case of turbo codes, E_u is the energy in Joules per bit per iteration to run the iterative decoder, and $f(P_g)$ is the number of decoder iterations required. Note that $f(P_g)$ is a monotonic non-increasing function of P_g.

The receive energy consumption per information bit P_r can be rewritten as:

$$P_r = F(P_t) = E_u f(\frac{P_t}{E_m d^\alpha}) \tag{3}$$

It is easy to see that F is a monotonic non-increasing function of P_t.

The rest of this paper deals with the issue of determining the optimal transmission energy per bit, P_t, for each node in a network so as to maximize the network lifetime.

Although P_t represents the transmission energy (in Joules) per bit, throughout this paper, we use the term "power setting" to refer to the quantity P_t, for ease of discussion. We note that the transmission energy P_t is varied by changing the transmission power (through the quantity P_g in equation (1)).

III. Lifetime Maximization Problem

There are a number of scenarios where the ability to trade transmit power for receive power should be effective, particularly in networks with asymmetric power costs. Consider the following example: a "hot spot" node (or set of nodes) sits on a critical path between two large clusters of nodes that frequently communicate. If all nodes start with the same amount of energy, standard transmission schemes will result in such a node running out of energy well before its counterparts; hence, network lifetime will be limited. It is readily apparent how this problem can be ameliorated with the framework presented in this paper. Nodes transmitting to the critical node can increase their transmission energy significantly, thus allowing the critical node to reduce its energy used for decoding. Likewise, the critical node can operate at minimum transmission energy, thus placing the burden of decoding on its more energy-rich receive counterparts to expend energy. It is clear then that we want to determine the optimal power allocation among the nodes in order to maximize the system lifetime.

We, therefore, formulate the *Lifetime Maximization Problem* (LMP) for a wireless network, which is the main subject of this paper. The scenario that we consider in this paper is that of a network of wireless nodes and a data sink. The nodes generate data at some rate and disseminate the data to the data sink. For the purpose of data dissemination, nodes organize themselves into a tree rooted at the data sink. In this paper, we do not address the problem of how to construct such a tree. We instead assume that such a tree is already available. Indeed, algorithms for constructing dissemination trees have been proposed in [10], [11]. Our goal, then, is to maximize the lifetime of the tree, i.e., the time until the first node in the tree runs out of energy. As mentioned earlier, we are interested in a tree as it is considered to be an efficient dissemination structure for the scenario that we are interested in.

In this section, we consider a simpler version of LMP, called the *Single Power Lifetime Maximization Problem*, in which each node is constrained to use a single power setting throughout its lifetime. We first formulate this problem and then provide a solution to this problem. In the next section, we show that in some cases a larger lifetime can be achieved by allowing each node to have multiple power settings. We, then, formulate the *Multi-Power Lifetime Maximization Problem* to allow nodes to have multiple power settings and show how this problem can be solved optimally.

A. Single Power Lifetime Maximization Problem

Let n be the number of wireless nodes, excluding a data sink S, in the network, which are organized into a tree rooted at S. Each wireless node i ($1 \leq i \leq n$) generates data (destined for sink S) at a constant rate R_i and has a total remaining energy denoted by E_i. The sink S is assumed to be "plugged into the wall" and, hence, has an infinite amount of energy. We assume that *flow-conservation* is maintained at all interior nodes in the tree. More precisely, an interior node forwards traffic at a rate R'_i, which is the sum of the incoming flow rates from all its children (denoted by the set C_i) and its own data generation rate R_i. Each node i uses a transmission energy P_t^i in order to transmit data to its parent, denoted by $p(i)$. Let τ denote the **lifetime** of the network, which is defined as *the time until the first node in the tree runs out of energy*. Our goal, then, is to find the optimal vector of power settings of nodes, $\mathbf{P_t} = (P_t^1, \ldots, P_t^n)$, that maximizes the network lifetime τ.

The *Single Power Lifetime Maximization Problem* (SPLMP) is then formulated as follows:

$$\max_{\mathbf{P_t}} \quad \tau \qquad (4)$$

subject to:

(1) *Flow Conservation:*

$$R'_i = \sum_{j \in C_i} R'_j + R_i, \quad \forall i, 1 \leq i \leq n$$

(2) *Energy Constraints:*

$$R'_i P_t^i \tau + \sum_{j \in C_i} F(P_t^j) R'_j \tau \leq E_i \qquad \forall i, 1 \leq i \leq n$$

(3) *Transmission Energy Constraints:*

$$P_t^i \geq E_m d_{i,p(i)}^{\alpha}, \qquad \forall i, 1 \leq i \leq n$$

The first constraint represents the *flow conservation* property, as mentioned earlier. Since each node generates data at a constant rate, the constraint guarantees that *instantaneous* flow conservation is maintained at all times. The second constraint states that the total energy consumption of a node i over the lifetime of the network is less than or equal to its initial total remaining energy E_i. The third constraint specifies that the transmission energy per bit of node i must be greater than or equal to the minimum transmission energy required for successful decoding at node $p(i)$ located at a distance $d_{i,p(i)}$.

Having formulated the optimization problem, we next proceed to describe how to solve this problem.

B. Binary Search Algorithm

We show in Appendix A that the optimal solution to SPLMP can always be achieved by choosing power settings of nodes such that all nodes die out at the same time. Intuitively, this is because there is no advantage to a node having energy left when the network dies. As a result, the inequality in the energy constraints (constraint (2)) of the optimization problem (4) in Section III-A can be replaced with equality constraints.

We also observe that SPLMP can be solved iteratively in a bottom-up manner. During each iteration, the root node chooses a target lifetime τ that the network seeks to achieve. Initially, the leaf nodes have determined their transmit power setting in order to achieve the target τ simply by solving the equality constraint (2) of the optimization problem. Given the transmit power settings of the leaf nodes, the parent nodes, in turn, compute their transmit power settings in order to achieve τ, again by solving constraint (2). This process repeats until all children of the root node determine their power settings. However, the resulting power assignment may not necessarily be feasible. In particular, the power assignment may require a node (say i) to transmit at a power smaller than the required minimum i.e., $P_t^i < E_m d_{i,p(i)}^{\alpha}$. When this happens, we conclude that the lifetime τ is too large to be achieved by the network and a smaller lifetime needs to be sought. If, however, the resulting power settings of all the nodes do not violate any constraint of the optimization problem, then the lifetime τ is feasible and the optimal lifetime of the network must be greater than or equal to τ. As a result, a higher lifetime is sought in the next iteration. Thus, it can be seen that the process of finding the optimal lifetime is similar to a binary search process.

A formal specification of the algorithm is shown in Figure 1. This algorithm takes as input the data gathering tree T and an error margin Δ that determines the desired accuracy of the solution. The output of the algorithm is a lifetime within Δ of the optimal lifetime and the power setting of each node corresponding to the optimal lifetime. τ_u and τ_l represent the lower and upper bounds of the optimal lifetime and are updated after each round of algorithm execution. The algorithm returns a lifetime within Δ of the optimal lifetime, when the condition $\tau_u - \tau_l < \Delta$ is satisfied.

Let τ^* denote the optimal lifetime of the optimization problem (4) and τ denote the lifetime returned by the algorithm in Figure 1. From the algorithm specification, it is easy to see that the optimal lifetime, τ^* always satisfies the condition $\tau_l \leq \tau^* \leq \tau_u$. The variable τ in the algorithm, also, always satisfies the condition $\tau_l \leq \tau \leq \tau_u$. When $\tau_u - \tau_l < \Delta$, we can conclude that $|\tau^* - \tau| < \Delta$.

function COMPUTE-LIFETIME(T, Δ)
Input: Data-gathering Tree (T) and error-margin (Δ)
Output: Optimal network lifetime τ

$\tau_l \leftarrow 0$
$\tau_u \leftarrow C$
while FEASIBLE(T,τ_u) **do**
    $\tau_l \leftarrow \tau_u$
    $\tau_u \leftarrow 2\tau_u$
endwhile
while$((\tau_u - \tau_l) > \Delta)$ **do**
    $\tau \leftarrow \frac{\tau_l + \tau_u}{2}$
    if FEASIBLE(T,τ) **then**
        $\tau_l \leftarrow \tau$
    else
        $\tau_u \leftarrow \tau$
    endif
endwhile
return τ

function FEASIBLE(T, τ)
Input: Data-gathering Tree (T) and target lifetime (τ)
Output: **true** if τ is feasible, **false** otherwise

for $i = 1, \ldots, n$ **do**
    $P_t^i \leftarrow \frac{E_i - E_u \sum_{j \in C_i} R'_j F(P_t^j)\tau}{E_m d^{\alpha}_{i,p(i)} R'_i \tau}$
    if $P^i_{gap} < 1$ **then**
        return false
    endif
endfor
return true

Fig. 1. *Binary Search Algorithm* to achieve a power assignment among nodes that maximizes the lifetime of the data-gathering tree T.

Our analysis in Appendix B shows that the algorithm requires $O(\max(0, \log_2 \frac{\tau^*}{C}) + \log_2 \frac{\max(\tau^*, C)}{\Delta})$ iterations to converge to a lifetime within Δ of the optimal lifetime.

IV. Multi-Power Lifetime Maximization Problem

So far, we have considered the problem of maximizing the lifetime of a data-gathering tree when nodes are constrained to a single power setting. We next ask the question: can the lifetime of a network be improved by allowing each node to operate at multiple power settings? We answer this question by first formulating the *Multi-Power Lifetime Maximization Problem*, which allows each node to have multiple power settings.

A. Problem Formulation

Let k denote the maximum number of transmit power settings available to each node. Let $\mathbf{P^i_t} = (P^i_{t_1}, \ldots, P^i_{t_k})$ be k arbitrary transmit power settings used by node i for durations $\tau^i_1, \ldots, \tau^i_k$ respectively. Note that a node i may use fewer than k power settings. In other words, $\tau^i_j = 0, j \in [1, k]$, if power setting $P^i_{t_j}$ is not used by node i. The goal is to determine the durations $\tau^i_1, \ldots, \tau^i_k$ that each power setting is used so as to maximize the network lifetime τ. The *Multi-Power Lifetime Maximization Problem* (MPLMP) is formulated as follows:

$$\max \quad \tau$$

subject to:

(1) *Flow Conservation:*

$$R'_i = \sum_{j \in C_i} R'_j + R_i, \quad \forall i, 1 \leq i \leq n$$

(2) *Energy Constraints:*

$$\sum_{l=1}^{k} P^i_{t_l} R'_i \tau^i_l + \sum_{j \in C_i} \sum_{l=1}^{k} F(P^j_{t_l}) R'_j \tau^j_l \leq E_i, \forall i, 1 \leq i \leq n$$

(3) *Transmission Energy Constraints:*

$$P^i_{t_l} \geq E_m d^{\alpha}_{i,p(i)}, \forall i, 1 \leq i \leq n, \forall l, 1 \leq l \leq k$$

(4) *Lifetime Constraints:*

$$\sum_{l=1}^{k} \tau^i_l \geq \tau, \forall i, 1 \leq i \leq n, \forall l, 1 \leq l \leq k$$

(5) *Default Constraints:*

$$\tau^i_l \geq 0, \forall i, 1 \leq i \leq n, \forall l, 1 \leq l \leq k$$

Having formulated the problem, the rest of this section discusses its solution. Using a two-node network as an example, we first show that a single power setting may not always yield the optimal system lifetime. For a two-node network, we identify conditions under which multiple power settings become necessary to achieve optimal lifetime and also determine what those power settings should be. Finally, we use the results established for the two-node network to solve MPLMP.

B. Two-Node Network

A node i is transmitting data with a constant rate R_i to a receiver node j which is at a distance d from node i. Let E_i and E_j be the remaining energies of nodes i and j respectively. The transmitting node i is no longer constrained to use a single power setting. Let $\mathbf{P^i_t} = (P^i_{t_1}, \ldots, P^i_{t_k})$ be the transmit power settings used by node i for durations $\tau_1, \ldots, \tau_k$ respectively. We have the following optimization problem:

$$\max \quad \sum_{l=1}^{k} \tau_l \tag{5}$$

subject to:

(1) *Sender Constraints:*

$$\sum_{l=1}^{k} P^i_{t_l} R_i \tau_l \leq E_i$$

(2) *Receiver Constraints:*

$$\sum_{l=1}^{k} F(P^i_{t_l}) R_i \tau_l \leq E_j$$

(3) *Transmission Energy Constraints:*

$$P^i_{t_l} \geq E_m d^\alpha, \forall l, 1 \leq l \leq k$$

(4) *Default Constraints:*

$$\tau_l \geq 0, \forall l, 1 \leq l \leq k$$

Having formulated the two-node lifetime maximization problem, we first show why a single power setting is insufficient to achieve the optimal lifetime using a specific choice of F.

Consider $F(P^i_t) = \sqrt{100 - (P^i_t)^2}$. It is easy to see that F is non-increasing and concave. Furthermore, assume that $E_i = E_j = 10$ and $R_i = 1$. It can be seen that the optimal lifetime of this system is two seconds. This lifetime can be achieved using a power setting $(0, 10)$ for one second and $(10, 0)$ for one second. It is obvious that a lifetime of two seconds cannot be achieved using a single power setting.

We next observe that the number of power settings required to achieve the optimal lifetime depends on the properties of F. We also provide optimal solution for the two-node problem, when F is convex or concave.

1) Optimal Solution for the Two-Node Network:

Theorem 1: At most two transmit power settings $P^i_{t_1}$ and $P^i_{t_2}$ are required to achieve the optimal system lifetime in optimization problem (5), for any monotonic non-increasing F.

Proof: From (5), we know that transmitting node i uses k transmit power settings, determined by the vector $\mathbf{P^i_t} = (P^i_{t_1}, \ldots, P^i_{t_k})$ in order to achieve the optimal system lifetime. Let $\tau_1, \ldots, \tau_k$ be the corresponding durations of time that each power setting is used.

From Linear Programming theory, we know that the objective function $\sum_{l=1}^{k} \tau_l$ achieves its maximum at an extreme point $(\tau_1^*, \ldots, \tau_k^*)$. In our problem, an extreme point $(\tau_1, \ldots, \tau_k)$ corresponds to the intersection of k planes out of a possible $k+2$ planes (k planes due to the default constraints, and 2 planes due to the sender and receiver constraints). Therefore, the k planes that determine an extreme point must be composed of at least $k-2$ planes due to the default constraints. Hence, at most two elements of the extreme point $(\tau_1, \ldots, \tau_k)$ are greater than 0. This completes the proof. ■

We next show that a single power setting is sufficient to maximize lifetime, if F is a non-increasing linear function.

Lemma 1: A single transmit power setting P^i_t is sufficient to achieve optimality, when $F(P^i_t) = -k_i P^i_t + b_i$. The optimal lifetime is given by $(k_i E_i + E_j)/(b_i R_i)$. This lifetime is achieved at the power setting $P^i_t = (E_i b_i)/(k_i E_i + E_j)$.

Proof: When $P^i_t = (E_i b_i)/(k_i E_i + E_j)$, the receiver decoding cost is given by $F(P^i_t) = (E_j b_i)/(k_i E_i + E_j)$. The system lifetime evaluates to $(k_i E_i + E_j)/(b_i R_i)$. Next, we prove that the achieved lifetime is optimal.

We first formulate the lifetime maximization problem as a linear programming problem. From Theorem 1, we know that two power settings are sufficient to achieve the optimal lifetime. Let τ_1 and τ_2 be the time durations for which each of the two power settings are used. The maximization problem is then formulated as follows:

$$\max \quad \tau_1 + \tau_2$$

subject to:

(1) *Sender Constraints:*

$$R_i(P^i_{t_1}\tau_1 + P^i_{t_2}\tau_2) \leq E_i$$

(2) *Receiver Constraints:*

$$R_i[(-k_i P^i_{t_1} + b_i)\tau_1 + (-k_i P^i_{t_2} + b_i)\tau_2] \leq E_j$$

(3) *Transmit Energy Constraints:*

$$P^i_{t_l} \geq E_m d^\alpha, l = 1, 2$$

(4) *Default Constraints:*

$$\tau_1, \tau_2 \geq 0$$

From receiver constraints we have,

$$-k_i R_i (P^i_{t_1}\tau_1 + P^i_{t_2}\tau_2) + R_i b_i (\tau_1 + \tau_2) \leq E_j \tag{6}$$

Combining equation (6) with the sender constraints, we have $\tau_1 + \tau_2 \leq (k_i E_i + E_j)/(b_i R_i)$. This completes the proof. ■

We next use Lemma 1 to show that a single power setting is sufficient to achieve optimal system lifetime, even if F is a non-increasing convex function.

Theorem 2: A single power setting, P^i_t, suffices to achieve optimal lifetime, when F is a non-increasing convex function. The optimal power setting is the solution of $y = F(x)$ and $E_j x = E_i y$.

Proof: Let (P^i_t, P^j_r) be the solution of $y = F(x)$ and $E_j x = E_i y$. Let $y = -k_i x + b$ be a line tangent to $y = F(x)$ at (P^i_t, P^j_r). From Lemma 1, we know that the point

(P_t^i, P_r^j) achieves the optimal lifetime, if $F(x) = -k_i x + b$. However, for a non-increasing convex F, we know that $F(x) \geq -k_i x + b$. Thus, the optimal lifetime achieved with a non-increasing convex F must be less than or equal to that achieved when $F(x) = -k_i x + b$. Since the power setting (P_t^i, P_r^j) achieves the optimal lifetime when $F(x) = -k_i x + b$, it must necessarily achieve the optimal lifetime when F is a non-increasing convex function. This completes the proof. ■

Theorem 3: Two end point power settings suffice to achieve optimal lifetime, when $F(x)$ is a non-increasing concave function.

Proof: We first prove Theorem 3, if $F(x) = -k_i x + b$, i.e.F is a non-increasing linear function. We use the fact that any point on a line can be expressed as a convex combination of its two end points (P_{t1}^i, P_{r1}^j) and (P_{t2}^i, P_{r2}^j). Formally, any point (P_t^i, P_r^j) on the line can be expressed as $P_t^i = \alpha P_{t1}^i + (1-\alpha)P_{t2}^i$, $P_r^j = \alpha P_{r1}^j + (1-\alpha)P_{r2}^j$, where $\alpha \in [0, 1]$.

We know from Lemma 1 that a single power setting (P_t^i, P_r^j) suffices to achieve the optimal lifetime, τ, if F is a non-increasing linear function. Thus, a system that uses the operating point (P_t^i, P_r^j) for duration τ achieves the same lifetime as a system that uses the end point (P_{t1}^i, P_{r1}^j) for duration of $\alpha\tau$ and end point (P_{t2}^i, P_{r2}^j) for a duration of $(1-\alpha)\tau$. Since any point on the line can be replaced by two end points, the two end point settings are enough to achieve the optimal lifetime.

Next, we prove Theorem 3. For a non-increasing concave F, $F(x) \geq -k_i x + b$. Hence, the optimal lifetime achieved with the function $F(x) = -k_i x + b$ must be greater than or equal to that achieved with a non-increasing concave $F(x)$. As we have shown for a non-increasing linear F, the optimal lifetime is achieved using the power settings defined by the end-points (P_{t1}^i, P_{r1}^j) and (P_{t2}^i, P_{r2}^j) of $F(x)$. Hence, the optimal lifetime achieved by $F(x) = -k_i x + b$ must be achievable with a concave $F(x)$ with the same end-points. This completes the proof. ■

C. Optimal Solution for a Tree Network

We now use the results established for the two-node optimization problem to solve MPLMP described in Section IV-A.

We first observe that the optimization problem in IV-A can be solved iteratively in a bottom-up manner using an approach similar to that used in the *Binary Search Algorithm* in Figure 1. At a high level, the approach consists of identifying a target lifetime τ that the network seeks to achieve. Given τ, the leaf nodes determine the durations that each of their transmit power settings must be used in order to achieve τ and then their parent nodes in turn determine the durations for their transmit power settings and so on. If the lifetime τ is feasible, then a higher lifetime is sought by the network. Otherwise, a smaller lifetime is sought.

However, unlike SPLMP, for a given lifetime τ, a node now has multiple transmit power settings that it must employ in order to achieve the lifetime τ. We observe that for a given target lifetime τ, each node in the tree must choose the durations for each of its available transmit power settings so as to minimize the receive energy consumption of its parent node while achieving a lifetime τ. Thus, for a given τ, each leaf node first determines the durations for each of its transmit power settings to minimize the energy consumption of their parent nodes, while achieving a lifetime τ or longer. With the energy remaining after setting aside energy to receive data from its children, (denoted by E_i') each parent node i, in turn, determines the durations for each of its transmit power settings to minimize the receive energy consumption of its own parent, while achieving a lifetime of at least τ.

Based on the above discussion, we observe that once all of its children have determined their transmit power settings, each node i solves the following optimization problem to minimize the receive energy of its parent:

$$\min \quad \sum_{l=1}^{k} \tau_l F(P_{t_l}^i) R_i' \tag{7}$$

subject to:

(1) *Lifetime Constraints:*

$$\sum_{l=1}^{k} \tau_l \geq \tau$$

(2) *Sender Constraints:*

$$\sum_{l=1}^{k} P_{t_l}^i \tau_l R_i' \leq E_i'$$

(3) *Transmit Energy Constraints:*

$$P_{t_l}^i \geq E_m d_{i,p(i)}^{\alpha}, \forall l, 1 \leq l \leq k$$

(4) *Default Constraints:*

$$\tau_l \geq 0, \forall l, 1 \leq l \leq k$$

As in the two-node case, the number of power settings required to achieve optimality in (7) depends on the properties of F. We first claim that, for a non-increasing F, at most two transmit power settings are required to minimize the receiver energy in (7). We shall use the results established for the two node case in Section IV-B.1 in order to prove the claim. From Theorem 1, we know that at most two power settings are required to achieve optimal lifetime for any combination of values of E_i and E_j (sender and receiver energies). Therefore, it must be the case that at most two power settings are required to achieve a specific target lifetime τ (smaller than or equal to the optimal lifetime) in (7).

Similarly, from Theorem 2 and Theorem 3, we can argue that only one power setting is required to achieve optimality in (7) for the case when F is convex and two power settings suffice for the case when F is concave.

Based on the above, we note that, when F is convex, the optimal lifetime is achieved using only a single power setting per node. Therefore, the *Binary Search Algorithm* specified in Figure 1 can be used to obtain the optimal power allocation among nodes.

When F is concave, however, the optimal power settings are given by the two end-points of F as shown in Theorem 3. Let $P^i_{t_1}$ and $P^i_{t_2}$ represent the two end-point power settings of node i. We need to determine the durations that each power setting is used. The approach to finding the durations is again based on the idea of a binary search. The root node first chooses a lifetime τ that the network seeks to achieve. Since there is no advantage of a node having energy left when the network dies, the inequalities in constraints (1) and (2) of optimization problem (7) can be replaced with equalities. Hence, for a target lifetime τ, each node essentially needs to determine the durations $\alpha\tau$ and $(1-\alpha)\tau$ ($\alpha \in [0,1]$), that $P^i_{t_1}$ and $P^i_{t_2}$ are used respectively. The fraction α can be determined as follows:

After replacing the inequality in constraint (2) with an equality, we obtain:

$$P^i_{t_1}\alpha\tau R'_i + P^i_{t_2}(1-\alpha)\tau R'_i = E'_i$$

which yields:

$$\alpha = \frac{E'_i - P^i_{t_2}\tau R'_i}{(P^i_{t_1} - P^i_{t_2})\tau R'_i}$$

Without loss of generality, we assume $P^i_{t_1} > P^i_{t_2}$. Then, if $\alpha < 0$, it means that the target lifetime τ is too large and hence, a smaller lifetime is sought in the next iteration. Otherwise, a larger lifetime is sought in the next iteration. The process repeats until the lifetime returned by the algorithm is "sufficiently" close to the optimal lifetime.

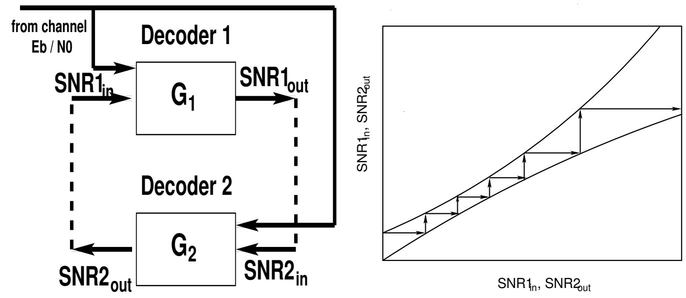
V. Numerical Results

In this section, numerical results are presented for channel coding using turbo codes. We first characterize the decoding costs of a turbo decoder and subsequently, demonstrate improvements in lifetime obtained by employing the power tradeoffs suggested in this paper.

A. Background

A turbo decoder performs iterative decoding and can be viewed as a nonlinear dynamical system with feedback [5], as shown in Figure 2(a). The system consists of two decoder components, decoder 1 and decoder 2. The input and output signal-to-noise ratios of decoder 1 are denoted by $SNR1_{in}$ and $SNR1_{out}$ respectively, while those of decoder 2 are denoted by $SNR2_{in}$ and $SNR2_{out}$. For a given E_b/N_0 (which represents the ratio of bit energy to noise of the received signal), the output of each decoder is a nonlinear function of its input signal-to-noise ratio, SNR. This nonlinear function is denoted by G_1 for decoder 1 and G_2 for decoder 2, as shown in Figure 2(a). Thus, $SNR1_{out} = G_1(SNR1_{in}, E_b/N_0)$ and $SNR2_{out} = G_2(SNR2_{in}, E_b/N_0)$. Also, $SNR1_{out} = SNR2_{in}$ and hence, $SNR2_{out} = G_2(G_1(SNR1_{in}, E_b/N_0), E_b/N_0)$.

Figure 2(b) illustrates the process of iterative decoding in a turbo decoder. Each curve in the figure represents the SNR output of a decoder component, which is input to the other decoder component. The arrows between the two curves



(a) Turbo decoding as a nonlinear system with feedback

(b) Illustration of turbo decoding process

Fig. 2. Turbo Decoding

represent the iterations of the decoder. It can be seen that the number of iterations to decode the received signal depends on how close the two decoder SNR curves are to each other. The narrower the separation between them, the greater the number of iterations required to decode and hence the greater the power consumption of the receiver. If the sender transmits at a higher power, the separation between the SNR curves increases, thereby, requiring fewer iterations at the receiver.

B. Determining Decoder Effort

Next, the dependence between the number of iterations in the decoding process and the transmit power is explored. The number of iterations required for turbo decoding is estimated using the techniques of [6], which employs the powerful method of density evolution [15]. In Figure 3, the number of decoder iterations as a function of P_g is plotted from the simulation of a rate-1/2 turbo decoder with block length $B = 32768$. The large block length is employed only to aid with accurate code characterization, and it should be noted that the power tradeoff techniques described in this paper apply to turbo codes with short block lengths - only the characterization technique to find $f(P_g)$ changes. From Figure 3, we see that a second-order exponential fit, as defined in equation (8), seems to capture the behavior well for P_g ranging from 1 to 19. For higher values of P_g, simulations indicate that $f(P_g)$ is equal to 1.

Based on these results, $f(P_g)$ is defined as follows:

$$\begin{aligned} f(P_g) &= 10^{(0.0008P_g^2 - 0.0659P_g + 0.9792)}), P_g \in [1, 19] \\ &= 1, P_g > 19 \end{aligned} \quad (8)$$

It is easy to see that f is **convex**. Hence, the function F, defined in equation (3) is also convex. As a result, the *Binary Search Algorithm* specified in Figure 1 is used to determine the optimal system lifetime.

C. Performance Evaluation

We now present some numerical results to illustrate the benefits of using the power tradeoff scheme over a scheme in which nodes always transmit at the minimum power.

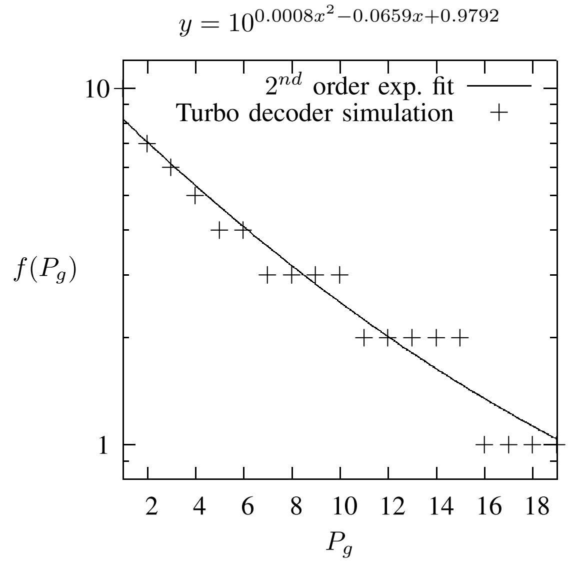


Fig. 3. Characterizing $f(P_g)$ for Turbo Codes

Recall from Section III-A that the energy constraint for a node i in SPLMP is given by the following expression:

$$E_m d_{i,p(i)}^{\alpha} R_i' P_g^{(i)} \tau + \sum_{j \in C_i} E_u f(P_g^j) R_j' \tau \leq E_i$$

Normalizing the above expression with respect to $E_m d_{i,p(i)}^{\alpha}$ yields:

$$R_i' P_g^{(i)} \tau + \sum_{j \in C_i} A_{i,p(i)} f(P_g^j) R_j' \tau \leq \frac{E_i A_{i,p(i)}}{E_u}$$

where $A_{i,p(i)} = \frac{E_u}{E_m d_{i,p(i)}^{\alpha}}$.

The parameter $A_{i,p(i)}$ represents the asymmetry between the transmit power of node i and the decoding cost of its parent node $p(i)$. It is a convenient representation since it groups together trends in technology (viz. E_u and E_m) as well as trends in the transmitter-receiver distance into a single parameter. It can be seen that the larger the value of $A_{i,p(i)}$, the greater is the decoding cost in comparison with the transmission cost and hence, higher the energy savings obtained by increasing the transmit power.

To get an idea of some typical values for $A_{i,p(i)}$, consider the following example. Assume a total (ambient plus receiver) noise temperature of 300K, a 10 dB SNR requirement for successful signal reception, a 10 KHz bandwidth, and a data rate of 10 kbits/s. Assuming omnidirectional antennas and the free space path loss exponent $\alpha = 2$, typical values of $A_{i,p(i)}$ can range roughly from 0.06 (carrier frequency 5 GHz, distance 3 m) to 0.558 (carrier frequency 5 GHz, distance 1m) to 1.55 (carrier frequency 1 GHz, distance 3 m). The value of E_u required for these sample numbers is estimated from [12]. This indicates that wide ranges of $A_{i,p(i)}$ are of interest for dense sensor networks. Furthermore, even small changes in the system requirements (data rate, bandwidth, required received SNR), can change the value of $A_{i,p(i)}$ significantly, with lower data rate systems (e.g., sensor network applications) yielding higher values of $A_{i,p(i)}$ and higher data rate systems the opposite.

We next illustrate the improvements in the lifetime of the data gathering network by employing the power tradeoff mechanisms described in this paper over the network when nodes always transmit at minimum power. Let τ^* denote the optimal lifetime of the network as obtained using the power assignment returned by the *Binary Search Algorithm*. Let τ_m denote the lifetime of a sensor system obtained using a "minimum-power-always" scheme. Then the lifetime improvement is given by the ratio $\rho = \frac{\tau^*}{\tau_m}$.

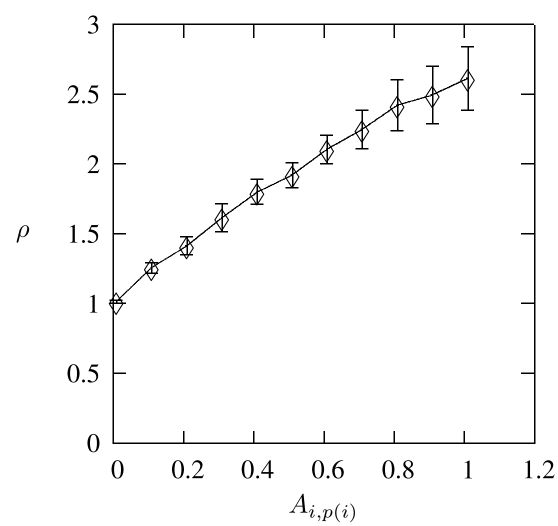


Fig. 4. ρ vs $A_{i,p(i)}$

1) Effect of $A_{i,p(i)}$: In Figure 4, we plot ρ as a function of the parameter $A_{i,p(i)}$. The parameter $A_{i,p(i)}$ is the same for all links in the tree. Each data point is obtained by averaging over 20 different runs of the *Binary Search Algorithm*. We also plot the 98% confidence intervals on the graph. In each run, a tree comprising 5000 nodes is constructed with each node having 2-7 children. Node data rates are chosen uniformly at random from the interval [10,100]. The energies of the nodes are chosen at random from the interval [2,3] $\times 10^5$.

As expected, as $A_{i,p(i)}$ increases, the energy savings also increase. This is because the cost of decoding increases in comparison to the transmission cost and so trading transmit power for receive power yields greater benefits. We see from Figure 4 that 1.5-2.5 times improvement in lifetimes are observed depending on the value of $A_{i,p(i)}$.

2) Effect of node degree: We next study the effect of node degree on lifetime improvement ratio ρ. Keeping the number of nodes in the tree fixed, we vary the average number of children per node in the tree from 3 to 11. The parameter $A_{i,p(i)}$ for each link in the tree is drawn randomly from the interval [0.01,3]. All other parameters are chosen exactly as described in Section V-C.1. We observe that as the average node degree increases, ρ also increases. This is because as node degree increases, the incoming data rate of a node increases and hence, a node has to expend a large amount of energy in decoding the received data. Thus, there is greater benefit in increasing the transmit power in order to reduce the burden of the energy-starved receiver. In particular, we observe 2-3 times improvement in network lifetime depending on the average node degree.

3) Effect of number of nodes and data rates: We also studied how ρ is impacted by varying the number of nodes in the tree. We vary the number of nodes, n, from 500 to 5000 without changing the node degree distribution i.e., each node in the tree has a random number of children chosen uniformly in the interval [2,7]. The parameter $A_{i,p(i)}$ is chosen from the interval [0.01,3]. All other parameters are chosen exactly as described in Section V-C.1. We observed 2.3-2.4 times improvement in the system lifetime over the chosen range of values of n.

We also evaluated the lifetime improvement ratio ρ as a function node data rate (varied from 10 to 100), for a fixed n. Again, we observed 2.3-2.4 times improvement in the lifetime over the range of values of R considered.

VI. Practical Considerations

We next discuss several practical issues concerning the *Binary Search Algorithm*.

A. Implementation Issues

There are two ways of implementing the *Binary Search Algorithm*. One is a distributed implementation. The sink node S broadcasts the targeted lifetime, τ, to all nodes in the tree. Each node then computes its power settings to achieve the lifetime and announces its power settings to its parent node, which in turn repeats the process. If a power constraint is violated at any node, then that node sends out a message to the sink. The sink then chooses a smaller lifetime and broadcasts it out to the nodes in the tree. The problem with this approach is that it introduces communication overhead. In particular, each node sends out one message and receives one message per iteration of the algorithm.

In order to alleviate the communication overhead, we advocate a centralized approach by shifting the computation of the optimal lifetime to the power-rich data sink S. However, the sink now needs to know the structure of the data-gathering tree. It also needs information about the remaining energies of the nodes, the node data rates and the length of the edges in the data-gathering tree. This is not a serious problem, since this information can be piggybacked on the actual data messages from the nodes to the sink, after the data-gathering tree has been constructed. Since this information needs to be sent only once to the sink node, it introduces very little overhead. The sink node then executes the *Binary Search Algorithm* locally and broadcasts out the optimal lifetime value. The nodes in the tree then compute their power settings in order to achieve the optimal lifetime.

B. Node Additions and Deletions

We next consider the issue of node additions and deletions from the data-gathering tree as a result of node mobility. Node additions and deletions result in a change in data rates and, hence, necessitate a change in power assignments among nodes.

Whenever the sink S detects that a node is deleted from the tree (notified either by the parent of the deleted node or one of the children of the deleted node), it executes the *Binary Search Algorithm* with the modified data-gathering tree (without the deleted node). Note that the sink requires no additional information from the nodes in the tree. It can compute the remaining energy levels of nodes in the tree by computing the energy consumption of each node until the instant when the deletion of the node occurred.

When a node is added to the tree, the newly added node announces its data rate, remaining energy and its parent node to the sink. The sink updates the tree structure and re-computes the optimal power assignment and broadcasts the results to the nodes in the tree.

Node movement also results in changes in length of the edges in the tree, which changes the power required by a node to communicate with its adjacent node. As a result, the optimal power allocation among nodes can potentially be different. Whenever there is a change in the length of an edge, one of the nodes adjacent to the edge can piggyback this information in its data messages to the sink node. Alternately, distance information can be piggybacked by each node to the sink on a periodic basis. The sink re-computes the optimal power assignment among nodes and announces the changes to the nodes.

C. Peak Power Constraints

In the SPLMP and MPLMP formulations, nodes have no constraints on the maximum power they can transmit at. In practice, there is a peak power constraint on each node's transmission. In this case, the power constraints in SPLMP and MPLMP changes to $P_{max} \geq P_t^i \geq E_m d^\alpha$. However, this requires only a minor modification in the *Binary Search Algorithm*. If the targeted lifetime τ results in a $P_t^i > P_{max}$ for node i, then node i sets its P_t^i to P_{max}. No other change to the algorithm is needed. Thus, with peak power constraints, there is a minimum decoding cost (equal to $F(P_{max})$) at the receiver.

D. Effect of Interference

In the SPLMP and MPLMP formulations, we do not explicitly account for interference caused due to nodes increasing their transmission powers. Increasing the transmission power of a node increases interference which, in turn, results in a decrease in node data rates. We assume that the node data rates are small, as is the case in low duty cycle sensor networks, and that, there are enough time slots available for transmission to each node to acheive its required data rate (despite the increased interference). Explicitly factoring interference into our optimization framework and studying the impact of interference on medium access control and scheduling is an interesting future direction.

E. Arbitrary Node Rates

At low signal-to-noise and interference ratio (SIR), which is the case in dense wireless networks, the achieved data rate on the link scales linearly with the SIR. Under these conditions, the authors in [3] obtain the optimal transmission strategy to be used by nodes: assuming a slotted system, in each time slot, a node either transmits at the maximum achievable link rate or remains idle. Using this transmission strategy in our scenario, it is possible to further reduce the energy consumption of the network. When node data rates are fixed, by transmitting at the maximum achievable link rate, a node can remain active for a fraction of time slot and can idle for the remaining fraction, thus saving energy. However, using this strategy instantaneous flow conservation cannot be maintained; flow conservation is only satisfied over the lifetime of the network. If nodes have arbitrary (or variable) data rates, however, nodes must still transmit at the maximum achievable link rate, whenever a slot is available for transmission and idle otherwise. However, because of the variable rate each node must have a large buffer in order to guarantee that no data is lost.

VII. Conclusions and Future Work

In this paper, the problem of optimal power allocation in a wireless network that allows for the tradeoff of transmitter power for receiver power has been considered. We considered the power allocation problem in a general setting common in ad hoc networks, viz. a network of wireless nodes organized into a tree rooted at a data sink. We formulated the *Lifetime Maximization Problem* (LMP) to determine the power levels of nodes that maximizes the lifetime of the tree, i.e. the time until the first node in the tree dies out. We first formulated a simpler version of LMP, called the *Single Power Lifetime Maximization Problem* (SPLMP), in which nodes are constrained to use a single power setting and proposed a *Binary Search Algorithm* to solve this problem optimally. We then showed that single power setting per node is insufficient to achieve the optimal lifetime and formulated the *Multi-Power Lifetime Maximization Problem* (MPLMP). We also showed how this problem can be solved optimally.

Using turbo codes as an example of channel-coding, significant energy savings were demonstrated when power tradeoffs are employed over the case when nodes always transmit at the minimum allowable power. In particular, 2-2.5 times improvement in network lifetimes were observed over a wide range of parameter values. The improvements will be even higher in networks with asymmetric power costs, for instance, networks with “hot spot” nodes which are on the critical path between two large clusters of nodes.

The encouraging results obtained in this paper point to a number of interesting future directions. While we considered a tree topology in this paper, the problem of optimal power allocation in a general network setting is an interesting future direction. In particular, the problem of joint routing and power allocation in such channel-coded wireless networks is a very interesting and challenging future direction. We would also like to explore transmitter-receiver power tradeoffs for objective functions other than the one considered in this paper. For instance, determining the optimal power settings to maximize the amount of data transmitted to a data sink is another interesting future direction.

Acknowledgements

This research was supported in part by the National Science Foundation under NSF grants ANI-0085848, EEC-0313747001 and ECS-0300130. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

We are extremely thankful to Prof. Wayne Burleson of the Electrical and Computer Engineering Dept. at UMass for numerous helpful discussions throughout the course of this work.

References

[1] A. Baptista, T. Leen, Y. Zhang, A. Chawla, D. Maier, W. Feng, W. Feng, J. Walpole, C. Silva, J. Freire. ”Environmental Observation and Forecasting Systems: Vision, Challenges and Successes of a Prototype”. *Encyclopedia of Physical Science and Technology (R. A. Meyers, Ed.)*, Academic Press, Third Edition, Vol. 5., pp 565-581.

[2] J. Chang and L. Tassiulas. ”Energy-conserving routing in Wireless Ad Hoc Networks”. In *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[3] R. Cruz and A. Santhanam. ”Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks”. In *Proceedings of IEEE INFOCOM*, San Francisco, Mar. 2003.

[4] S. Cui, A. Goldsmith and A. Bahai. ”Modulation Optimizations Under Energy Constraints”. *Proceedings of ICC’03*, Alaska, U.S.A, May, 2003.

[5] D. Divsalar, S. Dolinar and F. Pollara. ”Iterative turbo decoder analysis based on density evolution”, In *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 891-907, May 2001.

[6] H. El Gamal and A. Hammonds, Jr. ”Analyzing the turbo decoder using the Gaussian approximation”. In *IEEE Transactions on Information Theory*, vol. 47, pp. 671-686, Feb. 2001.

[7] A. Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu and S. Zahedi. ”Energy-efficient scheduling of packet transmissions over wireless networks”. In *Proceedings of IEEE INFOCOM*, 2002.

[8] D. Ganesan, R. Govindan, S. Shenker and D. Estrin.”Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks”. In *Mobile Computing and Communications Review (MC2R)*, vol 1., No. 2., 2002.

[9] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin and D. Ganesan . ”Building efficient wireless sensor networks with low-level naming”. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)*, 2001.

[10] W. Heinzelman, A. Chandrakasan and H. Balakrishnan. ”Energy-Efficient communication protocols for wireless microsensor networks”. In *Proceedings of Hawaii International Conference On System Sciences*, 2000.

[11] C. Intanagonwiwat, R. Govindan and D. Estrin. ”Directed Diffusion : A scalable and robust communication paradigm for sensor networks”. In *Proceedings of ACM International Conference on Mobile Computing and Networking, (MOBICOM)*, 2000.

[12] J. Kaza and C. Chakrabarti. ”Design and Implementation of Low-Energy Turbo Decoders”. In *IEEE Transactions on VLSI Systems*, vol. 12, no.9, pp. 968-977, Sep. 2004.

[13] A. Mainwaing, R. Szewczyk, D. Culler, J. Anderson. ”Wireless Sensor Networks for Habitat Monitoring”. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[14] P. Youssef-Massaad, M. Medard and L. Zheng, "Impact of Processing Energy on the Capacity of Wireless Channels". In *Proceedings of International Symposium on Information Theory and its Applications (ISITA)*, 2004.
[15] T. Richardson and R. Urbanke. "The capacity of low-density parity check code under message-passing decoding". In *IEEE Transactions on Information Theory*, vol. 47, pp. 599-618, Feb. 2001.
[16] A. Sankar and Z. Liu. "Maximum Lifetime Routing in Wireless Ad Hoc Networks". In *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.
[17] L. Schwiebert, S. Gupta, J. Weinmann. "Research Challenges in Wireless Networks of Biomedical Sensors". In *Proceedings of ACM International Conference on Mobile Computing and Networking, (MOBICOM)*, 2001.
[18] E. Shih, S. Cho, F. Lee, B. Calhoun and A. Chandrakasan. "Design Considerations for Energy-Efficient Radios in Wireless Microsensor Networks". *Journal of VLSI Signal Processing*, vol.37, pp 77-94, 2004.
[19] S. Vasudevan, D. Goeckel and D. Towsley. "Optimal Power Allocation in Channel-Coded Wireless Networks". In *Proceedings of Forty Second Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, October 2004.
[20] S. Vasudevan, C. Zhang, D. Goeckel and D. Towsley. "Optimal Power Allocation in Wireless Networks with Transmitter-Receiver Power Tradeoffs". *UMASS Computer Science Technical Report*, 05-31, May 2005. Available at http://www-net.cs.umass.edu/networks/publications.html
[21] S. Verdu. "Spectral efficiency in wideband regime". In *IEEE Transactions on Information Theory*, vol.48, pp 1319-1343, June 2002.
[22] W. Ye, J. Heidemann and D. Estrin. "An Energy-Efficient MAC Protocol for Wireless Sensor Networks". In *Proceedings of IEEE INFOCOM*, 2002.
[23] F. Zhao, J. Shin and J. Reich. "Information-Driven Dynamic Sensor Collaboration for Tracking Applications". In *IEEE Signal Processing Magazine*, March 2002.

APPENDIX

A. Result

Theorem 4: Let $\mathbf{P_t}$ denote the optimal solution to SPLMP and let τ denote the corresponding system lifetime. Furthermore, suppose that the power settings $\mathbf{P_t}$ are such that some subset of nodes have non-zero remaining energy after time τ. We now show that the lifetime τ can be achieved using a solution $\mathbf{Q_t}$ such that none of the nodes have any remaining energy after τ.

Proof: Consider a node i with non-zero remaining energy after time τ. From the *Energy Constraints* of SPLMP, we have:

$$R'_i P^i_t \tau + \sum_{j \in C_i} F(P^j_t) R'_j \tau < E_i$$

This means that there exists a power setting $Q^i_t > P^i_t$ such that i can achieve τ using Q^i_t. This Q^i_t can be obtained by solving the equation:

$$R'_i Q^i_t \tau + \sum_{j \in C_i} F(P^j_t) R'_j \tau = E_i$$

Since F is a monotonic non-increasing function of P^i_t, by choosing a larger transmit power setting, i cannot increase the receive energy consumption of its parent. Based on the new transmit power setting of i, parent of i in turn, computes its new total energy consumption and increases its transmit power setting, if it has energy remaining after τ. In this manner, we can iteratively re-adjust the power settings of nodes in a bottom-up fashion, starting with the leaf nodes in the tree. At the end of the iteration, the transmit power settings of the nodes achieved are such that each node dies out exactly after time τ. This means that the lifetime τ is achieved by a solution in which all nodes die out at the same time. ■

B. Analysis of the Binary Search Algorithm

Let m denote the number of iterations of the **while** loop in lines 3-6 of the function COMPUTE-LIFETIME specified in Figure 1. Then, we observe that m is the smallest integer that satisfies the inequality:

$$2^m C \geq \tau^*$$

$$\therefore m \leq \max(0, \log_2(\frac{\tau^*}{C}))$$

Upon termination of this **while** loop, we have:

$$T_u = 2^m C, T_l = 2^{m-1} C$$

The number of iterations of **while** loop from lines 7-14 of the function COMPUTE-LIFETIME depends on T_u, T_l and Δ. Smaller the Δ, greater the desired accuracy, thereby needing more iterations. Since the interval size $T_u - T_l$ is halved in every iteration, it requires $O(\log_2(\frac{T_u - T_l}{\Delta}))$ iterations for the condition $T_u - T_l < \Delta$ (or equivalently, $\frac{T_u - T_l}{\Delta} < 1$) to be satisfied.

Hence, the worst-case running time of the *Binary Search Algorithm* is $O(\max(0, \log_2(\frac{\tau^*}{C})) + \log_2(\frac{\max(\tau^*, C)}{\Delta}))$.