Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks

Yang Yu, Bhaskar Krishnamachari and Viktor K. Prasanna
Department of EE-Systems
University of Southern California
Los Angeles, CA 90089-2562
{yangyu, bkrishna, prasanna}@usc.edu

Abstract— **This paper provides algorithmic solutions to the challenging problem of scheduling packet transmissions for data gathering in wireless sensor networks. The focus of our work is to explore the energy-latency tradeoffs in packet transmission using techniques such as modulation scaling. The data aggregation tree – a multiple-source single-sink communication paradigm – is employed for abstracting the packet flow. We consider a real-time scenario in which the data gathering must be performed within a specified latency constraint. We present algorithms to minimize the overall energy dissipation of sensor nodes in the data aggregation tree without violating the latency constraint. For the off-line version of the problem, we propose (a) a numerical algorithm for optimal solutions, and (b) a faster approximation algorithm based on dynamic programming. We then discuss techniques for handling interference among the sensor nodes. Experimental results show that compared with the classic shut-down technique, between 20% up to 90% energy savings can be achieved by our techniques, under different settings of several key system parameters. Moreover, we develop an on-line distributed protocol that needs only local information of the aggregation tree. The significant energy saving achieved by the protocol is demonstrated through several run-time scenarios.**

Index terms – System design, Mathematical optimization

I. Introduction

In many applications for wireless sensor networks (WSNs) [1], data gathering is important for extracting useful information from the coupled environment to users. Recent studies [2], [3] show that data aggregation is particularly useful in eliminating the data redundancy and reducing the communication load.Typical communication patterns in data aggregation involve multiple data sources and one data sink (or recipient). Thus, the corresponding packet flow resembles a reverse-multicast structure, which is called the *data aggregation tree*.

Energy-efficiency is a key concern in WSN. The large number of sensor nodes involved in such networks and the need to operate over a long period of time require careful management of the energy resources. In addition, wireless communication is a major source of power consumption. Since a significant portion of the communication in WSN is due to data gathering, it is crucial to design energy-efficient communication strategies in implementing such an operation.

One of the useful approaches for energy-efficient communication is to explore the energy-latency tradeoffs. An important observation in [4] is that in many channel coding schemes, the transmission energy can be significantly reduced by lowering transmission power and increasing the duration of transmission. Techniques such as modulation scaling [5] have been proposed for implementing such tradeoffs. Foreseeing the integration of these techniques into commercial radio modules, we believe that it is important to analyze these techniques in the context of data gathering in WSNs.

This paper provides algorithmic solutions to the problem of scheduling packet transmissions over a data aggregation tree. The main purpose is to explore the energy-latency tradeoffs at the MAC layer with respect to the system performance at the application layer. We consider a real time scenario where the raw data gathered from source nodes must be aggregated and transmitted to the sink within a specified latency constraint. Our technique is applicable to any given aggregation functions. The objective function is to minimize the overall energy dissipation of the sensor nodes in the aggregation tree with respect to the latency constraint. Compared with [4], [6], we use a more general and accurate energy model for abstracting the energy characteristics for packet transmission in WSNs. Specifically, the transmission energy does not monotonically decrease as the transmission time increases – the transmission energy may increases when the transmission time exceeds some threshold value [7]. We refer to the above general model as the *non-monotonic energy model*.

For the off-line version of the problem, we present (a) a numerical algorithm for optimal solutions, and (b) a faster approximation algorithm based on dynamic programming. Both algorithms are capable of handling the non-monotonic energy functions. We also discuss techniques for handling interference. Experimental results show that compared with the classic shut-down technique proposed in PAMAS [8], between 20% up to 90% energy saving can be achieved by our algorithms, under different settings of several key system parameters. Next, we develop a distributed on-line protocol that needs only local information of the tree. Simulation results show that significant energy savings can be achieved by the protocol in several run-time scenarios.

Related work: The most relevant works include [4]–[6], [9]. In [5], an on-line policy for adjusting modulation level is proposed for single-hop communication. The problem of minimizing the energy dissipation for transmitting a set of packets over a single-hop link subject to a specified latency constraint is discussed in [4]. An extension of the problem that considers a single transmitter and multiple receivers is investigated in [6]. In both [4] and [6], optimal off-line algorithms and near-optimal on-line solutions are provided. In [9], the problem of balancing the energy dissipation along a multi-hop communication path is studied.

To the best of our knowledge, this is the first paper that addresses packet scheduling in a general tree structure. The challenges of our problem are multi-fold. Firstly, the energy functions can vary for different links. It is therefore required to develop general optimization techniques instead of explicit solutions. Secondly, the latency constraint for data gathering in real applications is typically given by considering the aggregation tree as a whole. It is difficult to directly applying the techniques in [6] and [4], as they require explicit latency constraints over each link. Lastly, we consider the non-monotonic energy functions, which has not been addressed in previous work. However, the tree structure leads us to an extension of the numerical optimization algorithm proposed in [6] as well as a recursive representation of the problem for applying dynamic programming.

Paper Organization: We discuss the background of our work in Section II. The packet transmission problem is defined in Section III. Off-line algorithms for the problem are presented in Section IV. In Section V, a distributed on-line protocol is described. Experimental results are shown in Section VI. Finally, the concluding remarks are given in Section VII.

II. Background

Data Aggregation Paradigm: We abstract the underlying structure of the network as a data aggregation tree. This is essentially a tree that gathers and aggregates information from multiple sources enroute to a single sink. Such a topological structure is common to data-centric routing schemes for sensor networks such as Directed Diffusion [2], [3]. While there may be transients during the route creation phase (for example, when the gradients are being determined in Directed Diffusion), we assume that this tree, once formed, lasts for a reasonable period of time and provides the routing substrate over which aggregation can take place during data gathering.

Specific techniques have been previously proposed for computing aggregates on such trees [10]. For our analysis we make the following abstraction: each sensor node in the tree aggregates the information from all its children or by local sensing so that it results in a reduced size packet that is dependent on the subtree rooted at the sensor node and an aggregation factor, $k \in [0, 1]$. For simplicity, we assume that each source node generates a data packet with the same size, s. Let d denote the number of source nodes in the subtree rooted at a sensor node, and s' denote the amount of output data after aggregation. Intuitively, low correlations among data make s' close to ds, while high correlations make s' close to s. We use the above aggregation factor, k, to indicate the degree of correlations among data with $k = 1$ meaning the highest correlations and $k = 0$ implying the lowest correlations. Based on the above intuition, we abstract the relationship between S' and s using equation 1. It can be verified that $s' = s$ when $k = 1$ and $s' = ds$ when $k = 0$.

$$s' = \frac{ds}{dk - k + 1} \tag{1}$$

We consider the following scenario. There is an epoch, during the beginning of which each source node generates a packet based on its sensing. It is desired to send the information contained in these source-generated packets to the sink before a deadline that occurs at or before the end of the epoch, when the next set of data samples are generated. Such an epoch-based scheme has been considered in [10], and requires the use of time-synchronization schemes such as [11]. Prior work, however, has not considered the possibility of using packet-scheduling techniques that trade latency for energy in such a scenario. This is the focus of our work.

Non-Monotonic Energy Model: We model the transmission energy using the example of modulation scaling [5]. Note that the algorithms presented in this paper are extendible to other techniques for exploring the energy-latency tradeoffs as well. For ease of analysis, we focus on the Quadrature Ampitude Modulation (QAM) scheme [12]. Consider a packet of s bits to be transmitted between two sensor nodes. Assuming that the symbol rate, R, is fixed, the transmission time, τ, can be calculated as [5]:

$$\tau = \frac{s}{b \cdot R} \tag{2}$$

where b is the modulation level of the sender in terms of the constellation size (number of bits per symbol). The corresponding energy function, w, is as follows [5]:

$$w(\tau) = [C \cdot (2^{\frac{s}{\tau \cdot R}} - 1) + F] \cdot \tau \cdot R \tag{3}$$

where C is determined by the quality of transmission (in terms of Bit Error Rate) and the noise power, and F is a device-dependent parameter that determines the power consumption of the electronic circuitry of the sender. Further, the output power, P_o, and the electronic power, P_e, can be modeled as follows [5]:

$$P_o = C \cdot R \cdot (2^b - 1) \tag{4}$$
$$P_e = F \cdot R \tag{5}$$

It is known that, for short-range communication with $R = 1$ Mbaud, the electronic power of the state-of-the-art radios is approximately 10 mW, while the output power is approximately 1 mW (at 4-QAM) [7], [13]. Assuming that it takes 100 pJ/bit/m^2 by the amplifier to transmit one bit at an acceptable quality [14], we infer that the desired communication radius is $\sqrt{5} \approx 2.2$ m. Moreover, from equations 4 and 5, it can be derived that $C \approx 3 \times 10^{-10}$ and $F = 10^{-8}$. In our study, we consider one more case of communication in WSNs – long-range communication with radius at 10m and the output power at 20 mW for 4-QAM ($C \approx 7 \times 10^{-9}$).

Figure 1 plots the energy functions with $b \in [2, 8]$ for the long and short range communication based on the above analysis. In practice, b is typically set to positive integers (indicated by circles), resulting in discrete values of τ. It can be observed that the transmission energy for the short-range communication eventually increases as the transmission time increases, while the energy function for the long-range communication is monotonically decreasing. Intuitively, it is more beneficial to explore the energy-latency tradeoffs for the long-range communication. However, we demonstrate in Section VI that up to 60% energy savings can still be achieved by our algorithms for the short-range communication.

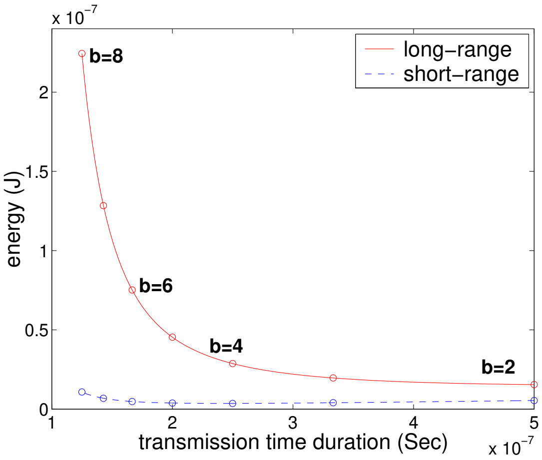


Fig. 1. Energy-latency tradeoffs for transmitting one bit data

III. Packet Transmission Problem over Data Aggregation Trees

A. Data Aggregation Tree

The data aggregation tree is represented by $T = (V, E)$, where node set V denotes the set of n sensor nodes, $\{V_i : i = 1, \ldots, n\}$, and edge set E denotes the set of directed communication links between the sensor nodes. Let M denote the number of leaf nodes in the tree. Without loss of generality, we assume that the sensor nodes are numbered in the topological order with $V_1, \ldots, V_M$ denoting the M leaf nodes and V_n denoting the sink node. Every link in E is represented as a tuple (i, j), meaning that a packet, denoted as P_i, needs to be transmitted from V_i to V_j. Let s_i denote the size of P_i.

While raw data is generated by the source nodes, data aggregation is performed by any non-sink and non-leaf node (called an *internal node* hereafter). We assume that aggregation is performed only after all input information is available – either received from children, or generated by local sensing. The aggregated data is then transmitted to the parent node. Although we use the expression in equation 1 as a typical aggregation function, please note that our technique is not limited to this function alone. The only requirement is that we can derive the value of s_i's based on the functions. Thus, even different functions can be used to specify the aggregation at different sensor nodes. The time and energy costs for generating raw data at source nodes or aggregating data at internal nodes are considered to be negligible.

Let T_i denote the subtree rooted at any node, V_i, with $T_n = T$. A path in T is defined as a series of alternate nodes and edges from any leaf node, $V_i, i \in \{1, \ldots, M\}$, to V_n, denoted as p_i. We use the notation $V_j \in p_i$ to signify that node V_j is an intermediate node of path p_i.

We assume that sensor nodes can be shutdown when there is no packet to transmit or receive. Mechanisms such as signaling channel [1] can be used for synchronization between sensor nodes before any packet transmission. However, the modeling of power assumption of such mechanisms is beyond the scope of this paper.

B. Problem Definition

As we consider an epoch-based scenario, let Γ denote the length of each epoch (or called latency constraint). For ease of analysis, it is assumed that raw data is available at source nodes at time 0. Further, we assume that the energy functions for all links in the target aggregation tree follow the model described in Section II (not necessarily using modulation scaling though).

A schedule of packet transmission is defined as a vector $\vec{\tau} = \{\tau_i : i = 1, \ldots, n-1\}$, where τ_i is the time duration for packet transmission over link (i, j). Since a sensor node can transmit its packet only after receiving all input packets from its children, the start time of each transmission is implicitly determined by $\vec{\tau}$. The transmission latency of a path, p_i, is denoted as L_i and calculated as $L_i = \sum_{j:V_j \in p_i} \tau_j$. A schedule is feasible if for any $p_i \in T$, we have $L_i \leq \Gamma$.

The goal of our problem is to improve the energy-efficiency of the system. Various objective functions can be developed for interpreting energy-efficiency. For ease of analysis, the objective function defined in this paper is the overall energy dissipation of the sensor nodes in the aggregation tree. Note that other objective functions, such as minimizing the maximal energy dissipation over all sensor nodes, can also be incorporated into the proposed techniques. Moreover, by assuming a first order energy model, the reception energy can be modeled by double the value of F in equation 3. Thus, we state the *packet transmission problem* (called PTP) as follows:

Given:
a. a data aggregation tree T of n sensor nodes, and
b. energy functions for each link $(i, j) \in E$, $w_i(\tau)$;
find *a schedule of packet transmission, $\vec{\tau}$, so as to minimize*

$$f(\vec{\tau}) = \sum_{i=1}^{n-1} w_i(\tau_i) \tag{6}$$

subject to

$$\forall p_i \text{ in } T, L_i = \sum_{j:V_j \in p_i} \tau_j \leq \Gamma \tag{7}$$

IV. Off-line Algorithms for PTP

In this section, we consider an off-line version of PTP (called OPTP) by assuming that the structure of the aggregation tree and the energy functions for all sensor nodes are known a priori. We first describe an extension of the MoveRight algorithm [6] to get optimal solutions for OPTP. A faster dynamic programming based approximation algorithm is then presented. Techniques for handling interference are also discussed.

A. A Numerical Optimization Algorithm

For a given energy function of any link $(i, j) \in E$ defined on interval $(0, \Gamma]$, let m_i denote the minima of the function. It is clear that in any optimal solution to OPTP, we have $\tau_i \leq m_i$. Consequently, the latency of a path does not necessarily equal Γ. Moreover, let V_i denote an internal node. For any optimal solution to OPTP, we show that the first derivative of the energy function of V_i equals the sum of the first derivatives of the energy functions of all children of V_i.

Lemma 1: A schedule, $\vec{\tau^*}$, is optimal for OPTP iff
(1) for any node V_i with $\tau_i^* < m_i$, the length of at least one path that contains V_i is equal to Γ; and
(2) for any internal node, V_i, we have

$$w_i'(\tau_i^*) = \sum_{(j,i) \in E} w_j'(\tau_j^*) \tag{8}$$

Proof: $\Rightarrow$ Condition (1) trivially holds for an optimal solution. Let A denote a matrix of size $M \times n$, where $A[i][j] = 1$ if and only if $V_j \in p_i$. Intuitively, the 1's in the i-th row of A indicate the set of nodes on path p_i. Let $\vec{\Gamma}$ be a $M \times 1$ vector with all elements equal to Γ. Then OPTP can be expressed as to minimize $f(\vec{\tau})$, subject to $A\vec{\tau} = \vec{\Gamma}$. Based on the first-order necessary condition for linearly constrained problems [15], there exists a vector of values $\vec{\lambda} = (\lambda_1, \ldots, \lambda_M)^T$, such that

$$\nabla f(\vec{\tau^*}) + A^T \vec{\lambda} = 0 \tag{9}$$

where $\nabla f(\vec{\tau^*})$ is a column vector with the partial derivative $\frac{\partial f(\vec{\tau^*})}{\partial \tau_i^*} = w_i'(\tau_i^*)$ as the i-th element. By solving equation 9, we have for any internal node, V_i,

$$w_i'(\tau_i^*) + \sum_{j:V_i \in p_j} \lambda_j = 0 \tag{10}$$

The sum in equation 10 sums up λ_j's such that path p_j passes through V_i. Since any path containing the children of V_i must also pass through V_i, we have

$$\sum_{j:V_i \in p_j} \lambda_j = \sum_{(k,i) \in E} \{ \sum_{j:V_k \in p_j} \lambda_j \} \tag{11}$$

Thus, the necessary condition of optimality can be obtained by summing up equations 10 for all the children of V_i and subtracting equation 10 for V_i.

$\Leftarrow$ The proof follows the fact that the feasible space of OPTP is convex. ■

The MoveRight algorithm developed in [6] solves the problem of scheduling packet transmission over a single transmitter multiple receiver connection, which is essentially a special case of OPTP with a pipeline of sensor nodes. Based on Lemma 1, we extend the MoveRight algorithm to solve OPTP in a general-structured aggregation tree with non-monotonic energy

Begin

1\. $k = 0$ // initialize iteration counter

2\. **For** $(i, n) \in E$, set $\tau_i^k = \min\{\Gamma, m_i\}$ // initialize transmission time for links to the sink

3\. **For** $(i, j) \in E$ such that $j \neq n$, set $\tau_i^k = 0$ // initialize transmission time for other links

4\. $flag = 0$ // flag to keep track of convergence in the iterations

5\. **While** $flag = 0$

6\. $\quad k = k + 1$ // increment the iteration counter by 1

7\. $\quad$ **For** each V_i with i from $n - 1$ downto M+1 // perform local optimization for each internal node in T

8\. $\qquad (\{\tau_j^k\}_{(j,i)\in E}, \tau_i^k) =$ best$(\{\tau_j^{k-1}\}, \tau_i^{k-1})$ // move right the start time of transmission from V_i

9\. $\quad$ **For** $(i, n) \in E$

10\. $\qquad$ set $\tau_i^k = \min\{m_i, \Gamma - (\max_{V_i \in p_j}\{L_j\} - \tau_i^k)\}$ //increase the transmission time for links to the sink

11\. if $\vec{\tau}^k = \vec{\tau}^{k-1}$ $flag = 1$ // check convergence

End

Fig. 2. Pseudo code for EMR-Algo

functions. The pseudo code for the extended MoveRight algorithm (EMR-Algo) is shown in Figure 2. Initially, we set the starting time for all packet transmission to zero – the transmission time for links to the sink is set to $\min\{\Gamma, m_i\}$, while the transmission time for the rest links is set to 0 (Steps 2 and 3). The main idea is to iteratively increase (move right) the starting times of packet transmissions, so that each move locally optimizes the overall energy function. Finally, this iterative local optimization leads to a globally optimal solution.

In Figure 2, τ_i^k denotes the value of τ_i in the k-th iteration. The best$(\cdot)$ function returns the transmission durations for node V_i and its children, such that Lemma 1 holds for V_i with respect to the invariant that $\tau_i^k \leq m_i$. Since the value of τ_i^k must lie within $(0, \tau_i^{k-1}]$, the best$(\cdot)$ function can be easily implemented using binary search. Step 10 is important as it moves right the complete time of transmissions on links to the sink. This movement stops when the latency constraint is reached.

The correctness of EMR-Algo can be proved by exploring the convexity property of the energy functions. Intuitively, suppose an internal node, V_i, satisfies Lemma 1 in iteration k. Then in iteration $k+1$, the start time of the transmission from V_i is increased to a local optima by best$(\cdot)$ in either of the two cases: (a) if the start time of the transmission from any of V_i's children increases, or (b) if the complete time of the transmission from V_i increases. The complete proof is presented in [16].

The convergence speed of EMR-Algo depends on the structure of the aggregation tree and the explicit form of the energy functions. It is therefore difficult to give a theoretical bound on the number of iterations. However, by approximating $w_i(\tau)$ with a set of interpolated discrete values, we develop a much faster approximation algorithm based on dynamic programming. We present the approximation algorithm in Section IV-B.

B. A Dynamic Programming Based Approximation Algorithm

For ease of analysis, we assume that for each sensor node, D discrete values are evenly distributed over $[0, \Gamma]$ in the domain of τ. Let ε be the difference between two adjacent values. That is $\varepsilon = \frac{\Gamma}{D}$. Hereafter, D is called the approximation accuracy. Higher value of D leads to a more accurate approximation of the energy function. By changing D, we can explore the tradeoffs between the quality of the solution and the time cost of the algorithm.

Let $g(V_i, t)$ denote the minimal overall energy dissipation of a subtree rooted at V_i within latency constraint t. The original OPTP problem can be expressed as $g(V_n, \Gamma)$. It is clear that for any sensor node V_i, $g(V_i, t)$ can be computed as the sum of (a) the energy dissipation for packet transmission by the children of V_i, and (b) the energy dissipated by transmitting packets within the subtrees rooted at each child of V_i. Additionally, the packet transmission time from any child of V_i can take $\frac{t}{\varepsilon}$ values, namely $\varepsilon, 2\varepsilon, \ldots, t$. Therefore, we have the following recursive representation of $g(V_i, t)$:

$$g(V_i, t) = \begin{cases} w_i(t), & \text{for } 1 \leq i \leq M \\ \sum\limits_{(k,i)\in E} \left(\min\limits_{j=1}^{\frac{t}{\varepsilon}}\{w_k(j\varepsilon) + g(V_k, t - j\varepsilon)\}\right), & \text{otherwise} \end{cases} \tag{12}$$

The above representation is suitable for a dynamic programming based algorithm (DP-Algo for short). DP-Algo can be viewed as a procedure to build a table of size $D \times n$ (Figure 3). The i-th column from the left side corresponds to sensor node V_i, while the j-th row from bottom-up corresponds to $j\varepsilon$. After the execution of DP-Algo, the cell crossed by the j-th row and the i-th column shall contain the value of $g(V_i, j\varepsilon)$.

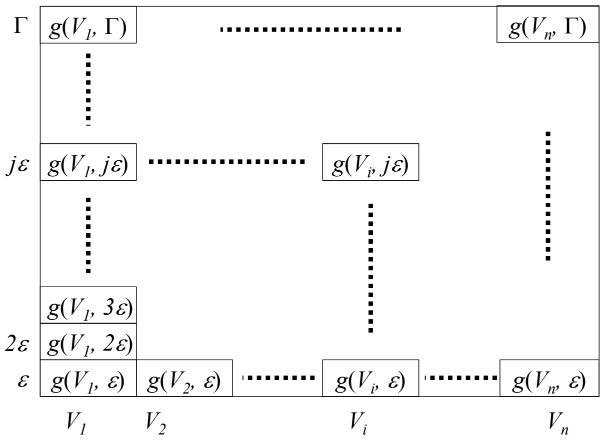


Fig. 3. The $g(\cdot)$ table computed by DP-Algo

To build the table, we start from the bottom left cell that contains $g(V_1, \varepsilon) = w_1(\varepsilon)$. The table is then completed column by column, from left to right. To calculate the value of $g(V_i, j\varepsilon)$ for $i > M$, we need to compare, for each child of V_i, j different values by varying the packet transmission time of the child. Therefore, the time cost for building up the table is $O((\frac{\Gamma}{\varepsilon})^2(|V| + |E|))$, which is pseudo-polynomial due to the factor Γ^2.

A Special Case for Modulation Scaling: In practice, the modulation levels are typically set to positive even integers. Based on equation 2, it can be verified that the τ_i's resulted from different modulation levels are not evenly distributed among $[0, \Gamma]$. Thus, DP-Algo cannot be directly applied. However, one practical method is to, for each i, set τ_i obtained by EMR-Algo or DP-Algo to the largest time duration below τ_i that can be achieved by an available modulation level. We call the above method the *rounding procedure*.

Another issue with modulation scaling is that the maximal or minimal transmission time can be bounded, due to the lowest and highest settings for the modulation level. Such boundaries can be captured by filling infinite to the corresponding cells in the table.

C. Handling Interference

The definition of OPTP implicitly assumes that there is no interference among the sensor nodes. Such an assumption can be realized by using MAC layer scheduling or multi-packet reception (MPR) through spatial, time, frequency, or code diversity [17]. However, use of such techniques may increase the hardware cost of the sensor nodes.

In case the above techniques are not available, one possible way for handling interference is to intentionally set the latency constraint imposed on OPTP to be less than the actual constraint. The preserved laxity can then be used for accommodating the back-off time of the sensor nodes when collision occurs. However, energy dissipation of the system will be increased due to the preserved laxity. By adjusting the amount of laxity, tradeoffs can be explored between the extra energy dissipation and the system ability for satisfying the latency constraint.

A more systematic way is to carefully schedule the transmission of sensor nodes that can potentially interfere with each other. The goal is to ensure that the corresponding time periods for a group of potentially interfering sensor nodes do not overlap with each other. Intuitively, children of a sensor node are potentially interfering – they cannot send packets to the parent at the same time. In the following, we describe a modified DP-Algo under the assumption that any group of potentially interfering sensor nodes are children of the same node.

To solve OPTP with the above interference restriction is actually non-trivial, as for any sensor node, the order of packet transmission from its children matters – the child that transmits earlier has a larger latency constraint over the subtree rooted at the child. Our basic idea is to divide the latency constraint over any subtree T_i (rooted at V_i) into two parts. We schedule the packet transmission in the subtrees rooted at each child of V_i with respect to the first part of the latency constraint. The packet transmission to V_i from its children are then carefully scheduled for avoiding the interference among the children with respect to the second part of the latency constraint The optimal division of the latency constraint over T_i can be found using dynamic programming with the following recursive representation of $g(V_i, t)$:

$$g(V_i, t) = \begin{cases} w_i(t), \text{ for } 1 \le i \le M \\ \min_{j=1}^{\frac{t}{\varepsilon}} \{z(i, j\varepsilon) + \sum_{(k,i) \in E} g(V_k, t - j\varepsilon)\}, \text{ otherwise} \end{cases} \tag{13}$$

The function $z(i, j\varepsilon)$ returns a schedule for the packets from children of V_i within time duration $j\varepsilon$. Obviously, if j is less than the number of children of V_i, no feasible solution exists. Otherwise, we use the following greedy heuristic. Initially, the transmission time from all children of V_i to V_i are set to ε. Let the energy gradient of a sensor node be the energy gain that can be obtained by increasing the current transmission time of the node by ε. Note that the energy gradient can be negative if the transmission time exceeds the minima in the energy function. We then increase the transmission time of the child with the maximal positive energy gradient by ε. The above operation is repeated until the sum of the transmission time of all children reaches $j\varepsilon$, or no more energy savings can be achieved by increase the transmission time (i.e., the gradients of all children are negative). We call the modified DP-Algo as the DP-IA algorithm. It can be verified that the time complexity of DP-IA is also $O((\frac{\Gamma}{\varepsilon})^2(|V|+|E|))$.

V. Distributed On-line Protocol

The algorithms presented in Section III all assume a complete knowledge of the aggregation tree. However, the discrete approximation of the energy function motivates an on-line distributed protocol that relies on local information of the aggregation tree only. To facilitate the on-line scheduling, we make the following assumptions:

1) Some local unique neighbor identification mechanisms are available at each sensor node for identifying the parent and children.
2) Every sensor node can derive the time cost for data gathering within the subtree rooted at the node.
3) Every sensor node is able to measure its contemporary power consumption, and hence its energy gradient – the energy gain that can be obtained by increasing the transmission time of the node by ε.
4) Interference between sensor nodes is minimized either by MAC layer scheduling or multi-packet reception techniques.

The local identifier in assumption 1 is commonly implemented in protocols such as Directed Diffusion [2]. The parent and children information is set up after constructing the aggregation tree. Assumption 2 can be fulfilled by attaching a time stamp to each packet from the leaf nodes, provided that time synchronization schemes, such as [11], are available. The power consumption and energy gradient of a sensor node can be determined using the system parameters provided by the hardware vendors and the operating configuration of the system, such as the modulation level and packet size.

In the following, we first describe the local data structure maintained at each sensor node. A distributed adaptation policy for heuristically minimizing the energy dissipation is then proposed.

Local Data Structure: Each sensor node, V_i, maintains a simple local data structure (r, τ_i, τ_d). The flag r equals one if V_i is the node with the highest *positive* energy gradient in subtree T_i, and zero otherwise. Field τ_i is the time cost for transmitting the packet from V_i to its parent, while τ_d records the time cost for data gathering within subtree T_i (excluding τ_i).

The local data structure is maintained as follows. Every leaf node attaches its energy gradient to the outgoing packet. Once a sensor node, V_i, receives packets from all its children, the node compares the energy gradients attached to each packet and the energy gradient of its own. The value of r at V_i is then set accordingly. If V_i is not the sink, the largest energy gradient from the above comparison is attached to the packet to the parent of V_i. The above procedure continues till all the sensor nodes have the correct value of r. Fields τ_i and τ_d can be easily maintained based on the above assumptions.

Adaptation Policy: The sink node periodically disseminates a feedback packet to its children that contains the value of its local τ_d and the difference between Γ and τ_d, denoted as δ. Note that if Directed Diffusion [2] is used for maintaining the aggregation tree, the feedback packet can be easily embedded into the *interest* packet sent by the sink.

Once a sensor node, V_i, receives the feedback packet, it checks its local data and performs one of the following actions. To distinguish from the field τ_d in V_i's local data, let τ_d' denote the field τ_d in the feedback packet.

1) If $\delta < 0$, the transmission time for packet from V_i is decreased by a factor of β, where β is a user-specified parameter. The feedback packet is then forwarded to all of V_i's children.
2) If $r = 1$ and $\delta \geq \varepsilon$, the transmission time of V_i's outgoing packet is increased by ε. Moreover, the local data structure at V_i is updated accordingly; and the feedback packet is suppressed.
3) Otherwise, the feedback packet is updated by setting $\delta = \delta + (\tau_d' - \tau_i - \tau_d)$ and $\tau_d' = \tau_d$. The updated packet is then forwarded to all children of V_i.

The rationale behind the above adaptation policy is that when the latency constraint is violated, all the sensor nodes send out packets in an increased speed. If V_i is the node with the largest positive energy gradient in T_i and the latency laxity allows, the second action is performed to reduce the energy dissipation of V_i. Otherwise, the latency laxity is accumulated and the sensor nodes in T_i are recursively examined. Since we assume interference is handled by MAC layer scheduling or MPR techniques, the optimization of different branches from V_i is independent of each other.

Various tradeoffs can be explored in implementing the above protocol. Ideally, the adaptation should be performed under a stable system state. Thus, the period, α, for disseminating the feedback packet should be large enough to accommodate oscillation in system performance. However, a larger period means a longer convergence process with greater energy dissipation. There is also a tradeoff involved in selecting the value of β. A larger value of β leads to higher transmission speed when the latency constraint is violated. However, extra energy dissipation is caused if the violation is not dramatic. Intuitively, β should be related to the severity of the violation, which is indicated by the value of δ.

VI. Experimental Results

To conduct the experiments, a simulator was developed using the PARSEC [18] software, which is a discrete-event simulation language. The baseline in our experiments is that all sensor nodes send the packet at the highest speed and shutdown the radio afterward. This policy is proposed in the PAMAS protocol [8]. For a fair comparison, we first show data that does not consider the energy for starting up the radio in both the baseline and our techniques. We also show data when the start-up energy is considered.

The purposes of the experiments are:

1) to demonstrate the energy gain achieved by EMR-Algo and DP-Algo compared with the baseline;
2) to evaluate the impact of several key system parameters to the performance of our algorithms; and
3) to show the energy saving and the adaptation capability of our on-line protocol in various run-time scenarios.

A. Experimental Setup

A sensor network was generated by randomly scattering 200 sensors in a unit square. The sink node was put at the left-bottom corner of the square. The number of sensor nodes that can communicate directly to a specific node is determined by a connectivity parameter, $\rho \in (0, 1]$, such that the average number of neighbors of a sensor node is $200\pi\rho^2$. We used the so-called random sources (RS) model [3] for generating the location of the data sources. Specifically, N (the number of sources) of the sensor nodes are randomly selected to be the sources. The Greedy Incremental Tree (GIT) algorithm [3] was used for constructing the data aggregation tree. The data aggregation tree generated by using the RS model is illustrated in Figure 4 with ρ set to 0.15. For ease of analysis, we assume that the size of raw data generated by any source node equals 400 bits. In addition, the aggregation factor, k, is assumed to be the same for all the sensor nodes.

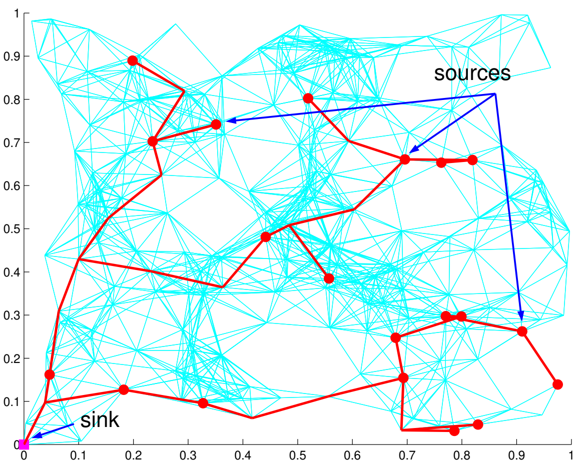


Fig. 4. A example data aggregation tree based on the RS model (connectivity parameter $\rho = 0.15$, number of sources $N = 20$)

The energy function used in the experiments was in the form of equation 3. Note that τ_i's are assumed to be continuous variables except in the special case of modulation scaling, when τ_i's have discrete values determined by the modulation level of the sensor nodes. We set $R_i = 10^6$ and $F_i = 10^{-8}$ for all the sensor nodes, while the value of C_i was determined by the distance from node V_i to its parent in the aggregation tree. More specifically, we assume a d^2 power loss model, where d is the distance between V_i and its parent. That is $C_i = C_{base} \cdot (\frac{d}{\rho})^2$. Based on our analysis in Section II, C_{base} was set to 7×10^{-9} for the long-range communication and 3×10^{-10} for the short-range communication.

To investigate the performance of our algorithms under various latency laxity, we use a user-specified parameter – *normalized latency constraint* to indicate the tightness of the latency constraint. Specifically, let $u \in (0, 1]$ denote the normalized latency constraint, with a higher value of u meaning a tighter latency constraint, and consequently, less laxity for exploring the energy-latency tradeoffs. The latency constraint was set in the following way. Equation 2 was used for modeling the time cost for packet transmission. We assume that the highest value of b_i is 8. The minimal time cost for data gathering, t_{min}, was calculated by assuming all packets are transmitted at the highest speed – $b_i = 8$ for all the sensor nodes. Then, we set the latency constraint as $\frac{t_{min}}{u}$.

B. Performance of the Off-Line Algorithms

The performance metric is defined as the percentage of energy savings achieved by using our techniques, compared with the baseline. In the experiments, the number of sources, N, was varied from 10 to 30 in increments of 10. The connectivity parameter, ρ, was varied from 0.1 to 0.3 in increments of 0.1. The aggregation factor, k, was varied from 0.3 to 1.0 in increments of 0.35. The normalized latency constraint, u, was varied from 0.1 to 1.0 in increments of 0.1. In addition, the approximation accuracy, D, was set to be 50 or 100.

For each case of the experiments, more than hundred of instances were conducted. The presented data has a 95% confidence interval with a 5% (or better) precision. In each instance, the source nodes were randomly selected from the sensor field shown in Figure 4. The data aggregation tree was then generated using GIT.

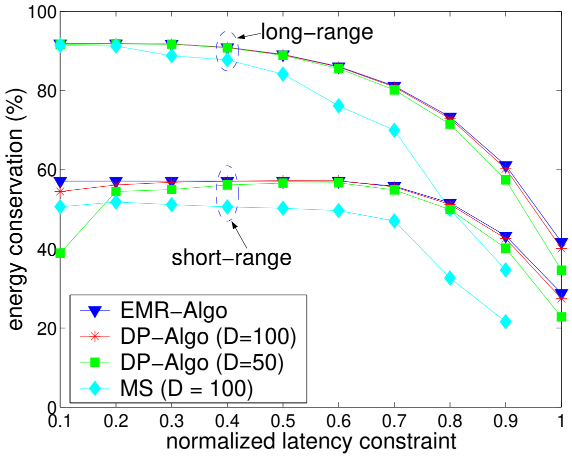


Fig. 5. Performance of our off-line algorithms (connectivity parameter $\rho = 0.15$, number of sources $N = 20$, aggregation factor $k = 0.7$)

Performance of Our Off-Line Algorithms: Figure 5 demonstrates the energy saving achieved by our off-line algorithms in both long and short range communication scenarios. The investigated algorithms include EMR-Algo, DP-Algo (with D set to 100 and 50), and the special case of DP-Algo for modulation scaling (denoted as MS in the figure) with $D = 100$. For MS, the available modulation levels are even numbers between 2 and 8. Note that the performance for MS at $u = 1.0$ is not shown in the figure. This is because, when $u = 1.0$, the latency constraint equals the transmission latency by setting all modulation levels to 8. However, due to the approximation nature of our technique, we cannot find a feasible solution by the rounding procedure in that case.

The first thing to note is that up to 90% of energy saving is achieved by all algorithms in the long-range communication when u is small. Moreover, more than 20% energy saving is achieved by all algorithms when $u \leq 0.9$. Even when $u = 1.0$, EMR-Algo and DP-Algo can still save more than 30% of the energy in both communication scenarios.

The reason for successful energy saving even when $u = 1$ is as follows. Intuitively, the latency constraint is determined by the longest path in the aggregation tree. Thus, energy can be reduced for links on paths other than the longest one by trading the latency laxity of the paths. On one hand, when there exists only one path in the aggregation tree, our problem becomes similar to the one studied in [9]. No energy can be saved when $u = 1$ in this case. On the other hand, when the aggregation tree forms a star-like structure, all links, except the longest ones, can be optimized for saving energy.

The plot shows that the performance of DP-Algo improves when D increases. When $D = 100$, the performance of DP-Algo is quite close to the performance of EMR-Algo. Our experimental results show that in the worst-case instance, the DP-Algo ($D = 100$) achieved 92% of the performance of EMR-Algo. However, the performance of MS quickly degrades when u increases. This is because the first derivative of the energy function tends to $-\infty$ as τ tends to 0. Thus, the performance loss due to the rounding procedure becomes large when the feasible modulation levels are high, which is caused by a large value of u.

It can be observed that in short-range communication, the performance of EMR-Algo and DP-Algo becomes saturated when u decreases below 0.5. This is because the non-monotonic energy function limits the amount of latency laxity that can be traded for energy conservation. The little performance degradation of DP-Algo and MS when u approaches 0 is because with a fixed D, the approximation accuracy actually decreases when the latency requirement increases due to decrement of u. Our experimental results show that the performance degradation can be overcome by increasing D.

The simulation was performed on a SUN Blade1000 with a 750 MHz SUN UltraSPARC III processor. The running time of EMR-Algo is between 0.1 to 2 second. The running time of DP-Algo is around 0.003 second when $D = 50$ and 0.008 second when $D = 100$. Thus, the value of D can be used to trade the performance of DP-Algo for the running time.

Energy Conservation vs. Various System Parameters: We show the results of DP-Algo with $D = 100$ in the following study. Figure 6 shows the energy conservation achieved by DP-Algo with respect to variations in k and u. It was observed that while u mainly determines the energy gain, k does affect the performance of DP-Algo when u is large. In particular, when $u = 1.0$, the energy saved by DP-Algo decreases when k decreases. This is because smaller value of k causes larger size of data packet after aggregation. Thus, the energy dissipated by links close to the sink node dominates the overall energy

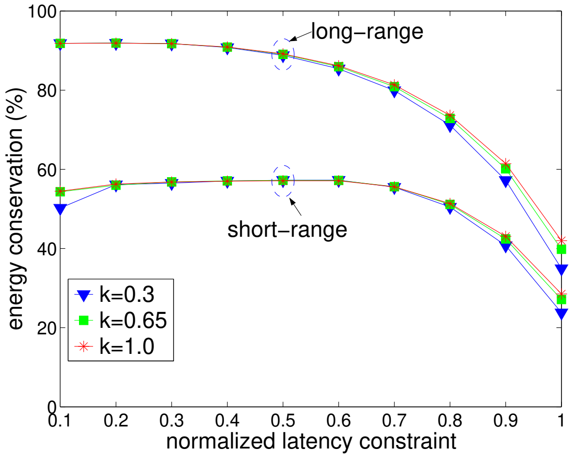


Fig. 6. Energy conservation versus normalized latency constraint and aggregation factor k (connectivity parameter $\rho = 0.15$, approximation accuracy $D = 100$, number of sources $N = 20$)

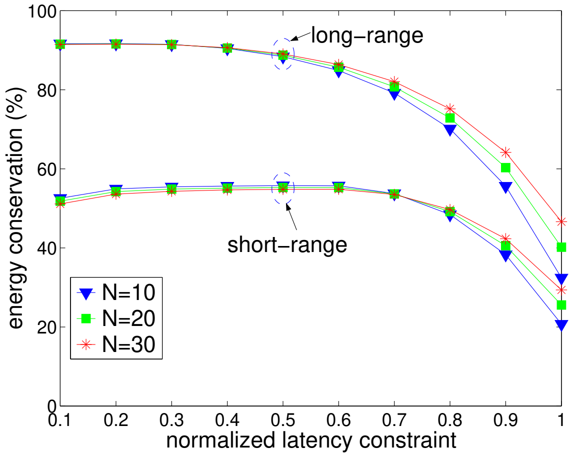


Fig. 7. Energy conservation versus normalized latency constraint and number of sources N (connectivity parameter $\rho = 0.15$, approximation accuracy $D = 100$, aggregation factor $k = 0.7$)

dissipation of the tree. It is however difficult to reduce the energy cost of these links since they have high likelihood to lie on the longest path of the aggregation tree.

Figure 7 plots the performance of DP-Algo with respect to variations in u and N. It can be seen that when u is large, the energy gain of DP-Algo increases as the number of sources increases. This is because larger number of sources facilitates the optimization of links on paths other than the longest one.

Figure 8 demonstrates the performance of DP-Algo with respect to variations in u and ρ. It can be observed that the energy saving of DP-Algo increases when ρ increase. This is understandable since large ρ reduces the height of the aggregation tree (the extreme case is a star-like aggregation tree formed by setting $\rho = 1$).

Together, the above results suggest that when u is small, DP-Algo is quite robust with respect to variations

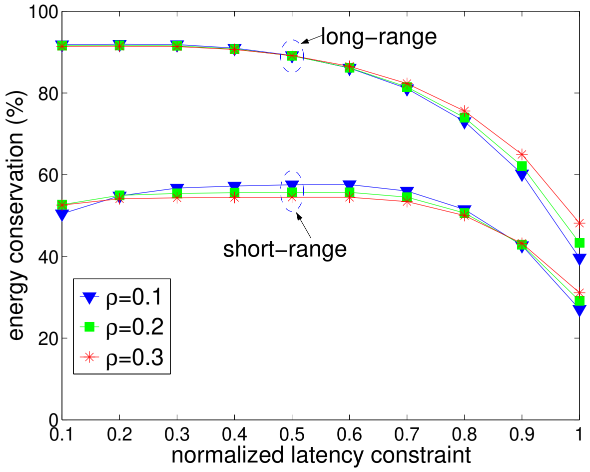


Fig. 8. Energy conservation versus normalized latency constraint and connectivity parameter ρ (approximation accuracy $D = 100$, aggregation factor $k = 0.7$)

in different system parameters, including k, ρ, and N.

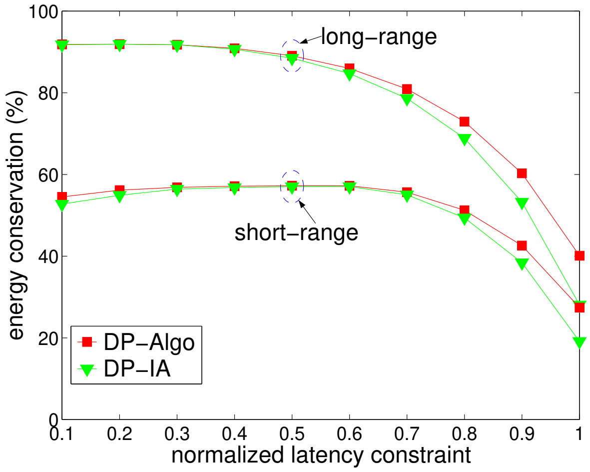


Fig. 9. Performance of the DP-IA algorithm (connectivity parameter $R = 0.15$, aggregation factor $k = 0.7$, number of sources $N = 20$)

Impact of Interference Avoidance: We also examined the performance of DP-IA for the two communication scenarios with respect to variations in u (Figure 9). It can be seen that the performance degradation due to interference avoidance is severe when u is large. In particular, a decrease of more than 10% in energy saving is observed for the long-range communication when $u = 1$. Thus, it is important to employ mechanisms for multi-packet reception or develop more efficient algorithms for handling interference. It can be observed that there is almost no effect of the interference avoidance when the performance of DP-Algo is saturated (e.g, when $u \leq 0.5$ in the short-range communication).

Impact of the Start-up Energy: We estimate the energy for starting up the radio as 1 μJ [7]. In each epoch,

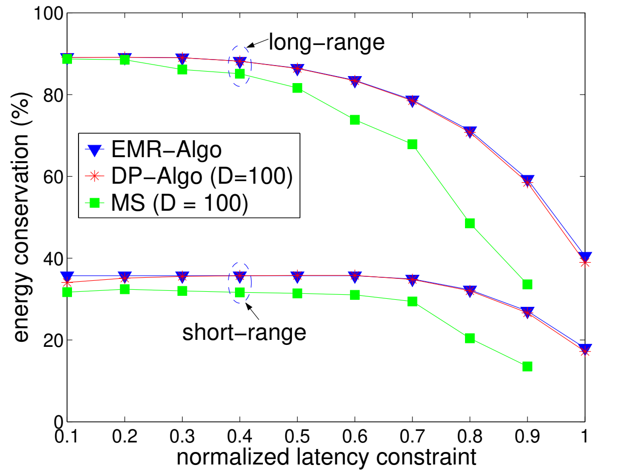
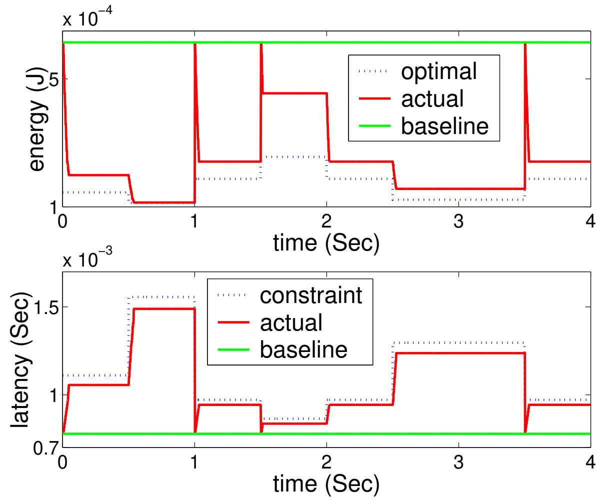
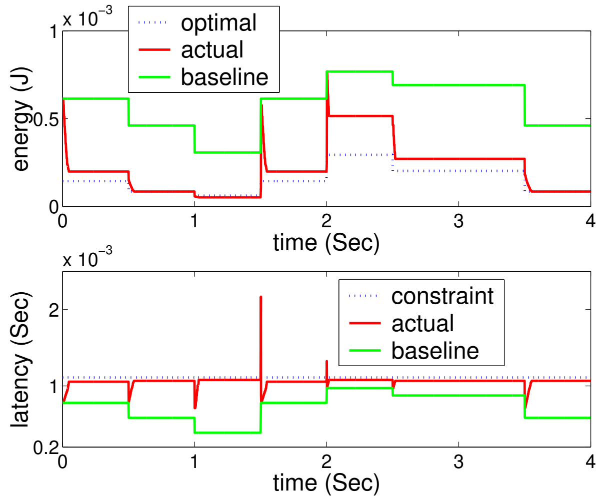


Fig. 10. Performance of our algorithms with start-up energy (connectivity parameter $\rho = 0.15$, aggregation factor $k = 0.7$, number of sources $N = 20$)



(a) Scenario A



(b) Scenario B

Fig. 11. Performance of our distributed protocol (connectivity parameter $\rho = 0.15$, aggregation factor $k = 0.7$)

the radio of each sensor node is started at most once. In addition, to emphasize the impact of start-up energy, we change the packet size to 200 bits. Figure 10 shows the performance of EMR-Algo, DP-Algo ($D = 100$), and MS with start-up energy. It can be observed that the impact of the start-up energy to the long-range communication is almost negligible. This is understandable since the 1 μJ start-up energy is negligible compared with the transmitting energy of the radio. However, the start-up energy is comparable to the transmitting energy in the short-range communication. Thus, a decrease around 20% in energy saving is observed for the short-range communication. However, the energy conservation of our algorithms is still considerable in the studied scenario.

C. Performance of the On-Line Protocol

Two run-time scenarios, namely A and B, were investigated to demonstrate the efficiency of the on-line protocol. In scenario A, we fixed the size of raw data packets while varying normalized latency constraint. In scenario B, we fixed the latency constraint while varying the size of raw data packets.

In both scenarios, an aggregation tree with 35 sensor nodes was used for the RS model. The aggregation tree is illustrated in Figures 4. We assume that modulation scaling is used by all the sensor nodes with the available modulation levels being even numbers between 2 and 8. The data gathering was requested every 2 mSec. For illustration, the period for sending the feedback packet, α, was set to 4 mSec. In Figures 11, the optimal energy cost is obtained by using EMR-Algo, while the baseline is the energy/time cost when all the sensor nodes send packets using the highest modulation level.

Scenario A: The size of raw data packets was fixed to 400 bits, while the normalized latency constraint was set to 0.7, 0.5, 0.8, 0.9, 0.8, 0.6, and 0.8 at time 0, 0.5, 1, 1.5, 2, 2.5, and 3.5 seconds, respectively. In real life, such variations can be caused by for example, changed user requests. By setting $\beta = 10$, the energy cost and latency for data gathering over 4 seconds are depicted in Figure 11.

It can be observed that when u is fixed, the actual energy cost gradually decreases till it is close to the optimal, while the latency approaches the constraint. At time 1 second, u is varied from 0.5 to 0.8, which causes a violation of the latency constraint. Due to the feedback mechanism, the transmission latency dramatically decreases as the modulation settings of all the sensor nodes are restored to higher levels. Consequently, the energy cost is also increased. After that, the energy cost drops again as time advances.

Note that by setting $\beta = 10$, the modulation levels of the sensor nodes were restored to almost the highest levels when a violation is detected. This is reflected by the high peaks in the energy curve.

Scenario B: The latency constraint was fixed to 0.6 mSec, while the size of raw data packets was set to 400, 300, 200, 400, 500, 450, and 300 at time 0, 0.5, 1, 1.5, 2, 2.5, and 3.5 seconds, respectively. In real life, the change of packet size may be caused by variations in gathered information, or the aggregation factor of individual sensor node. In this scenario, we abstract such variations by changing the raw data size as a whole. The energy cost and latency for data gathering over 4 seconds are depicted in Figure 11 with $\beta = 10$. An analysis similar to the one in scenario A can be performed.

In short, our on-line protocol is capable of saving significant energy in the studied scenarios. The ability of the protocol to adapt the packet transmission time with respect to the changing system parameters is also demonstrated.

VII. Concluding Remarks

This paper studied the problem of scheduling packet transmissions over a data aggregation tree by exploring the energy-latency tradeoffs. For the off-line version of the problem, we provided (a) a numerical algorithm for optimal solutions, and (b) a faster approximation algorithm based on dynamic programming. Techniques for handling interference were also discussed. Experimental results show that between 20% up to 90% energy saving can be achieved by the algorithms. We investigated the performance of our algorithms with different settings of several key system parameters. Furthermore, we proposed a distributed on-line protocol that relies on local information only. The significant energy saving achieved by the protocol and its ability to adapt the packet transmission time upon variations in the system parameters were demonstrated through several run-time scenarios.

We conclude the paper with a brief discussion of the future work. This paper considers an epoch-based application scenario. It would be interesting to model pipelined packet transmission, where the transmission speed is determined by the bottleneck link. Also, adaptive fidelity computation [2] for aggregation and compression provides more opportunities for exploring the energy-latency tradeoffs. The fidelity of the computation can be characterized by the size of the output data, which affects the consequent transmission time and energy costs. Thus, a broader tradeoff space could be explored.

References

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *The International Conference on Acoustics, Speech and Signal Processing*, May 2001.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *ACM MobiCom*, 2000.

[3] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *International Workshop on Distributed Event-Based Systems*, 2002.

[4] B. Prabhakar, E. Uysal-Biyikoglu, and A. E. Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *IEEE InfoCom*, 2001.

[5] C. Schurgers, O. Aberhorne, and M. Srivastava, "Modulation scaling for energy-aware communication systems," in *ISLPED*, 2001, pp. 96–99.

[6] A. E. Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi, "Energy-efficient scheduling of packet transmissions over wireless networks," in *IEEE InfoCom*, 2002.

[7] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, March 2002.

[8] C. S. Raghavendra and S. Singh, "PAMAS – power aware multi-access protocol with singaling for ad hoc networks," *Computer Communication Review*, July 1998.

[9] Y. Yu and V. K. Prasanna, "Energy-efficient multi-hop packet transmission using modulation scaling in wireless sensor networks," in *to appear on GlobeCom*, Dec. 2003.

[10] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation service for Ad-Hoc sensor networks,," in *5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Dec. 2002.

[11] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Dec. 2002.

[12] T. Ue, S. Sampei, N. Morinaga, and K. Hamaguchi, "Symbol rate and modulation level-controlled adaptive modulation/TDMA/TDD system for high-bit rate wireless data transmission," *Trans. on Vehicular Technology*, vol. 47, no. 4, pp. 1134–1147, Nov. 1998.

[13] A. Wang, S.-H. Cho, C. G. Sodini, and A. P. Chandrakasan, "Energy-efficient modulation and MAC for asymmetric microsensor systems," in *ISLPED*, 2001.

[14] W. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application specific protocol architecture for wireless microsensor networks," *IEEE Transaction on Wireless Networking*, 2002.

[15] G. P. McCormick, *Nonlinear Programming: Theory, Algorithms, and Applications*. John Wiley & Sons, 1982.

[16] Y. Yu, V. K. Prasanna, and B. Krishnamachari, "Energy-efficient packet scheduling for data gathering in wireless sensor networks," Technical Report, in preparation.

[17] S. Roy and H. Y. Wang, "Performance of CDMA slotted ALOHA multiple access with multiuser detection," in *Wireless Communication and Networking Conference*, vol. 2, 1999, pp. 839–843.

[18] PARSEC Project. [Online]. Available: http://pcl.cs.ucla.edu/projects/parsec