

# Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks

Yang Yu, Bhaskar Krishnamachari, and Viktor K. Prasanna

Department of Electrical Engineering

University of Southern California

Los Angeles, CA 90089-2562

{yangyu, bkrishna, prasanna}@usc.edu

**Abstract**— We study the problem of scheduling packet transmissions for data gathering in wireless sensor networks. The focus is to explore the energy-latency tradeoffs in wireless communication 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 where the data gathering must be performed within a specified latency constraint. We present algorithms to minimize the overall energy dissipation of the sensor nodes in the aggregation tree subject to the latency constraint. For the off-line problem, we propose (a) a numerical algorithm for the optimal solution, and (b) a pseudo-polynomial time approximation algorithm based on dynamic programming. We also discuss techniques for handling interference among the sensor nodes. Simulations have been conducted for both long-range communication and short-range communication. The simulation results show that compared with the classic shutdown technique, between 20% to 90% energy savings can be achieved by our techniques, under different settings of several key system parameters. We also develop an on-line distributed protocol that relies only on the local information available at each sensor node within the aggregation tree. Simulation results show that between 15% to 90% energy conservation can be achieved by the on-line protocol. The adaptability of the protocol with respect to variations in the packet size and latency constraint is also demonstrated through several run-time scenarios.

**Index terms** – System design, Mathematical optimization

## I. INTRODUCTION

In many applications of wireless sensor networks (WSNs) [1], data gathering is a critical operation needed for extracting useful information from the operating environment. 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 WSNs. 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 WSNs

is due to data gathering, it is crucial to design energy-efficient communication strategies in implementing such an operation.

One useful approach 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.

In this paper, we explore the above tradeoffs in the context of data gathering in WSNs, subject to application level performance constraints. We consider a real time scenario where the raw data gathered from the source nodes must be aggregated and transmitted to the sink within a specified latency constraint. Our technique is applicable to any given aggregation function. The objective function is to minimize the overall energy dissipation of the sensor nodes in the aggregation tree subject 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 increase 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 the optimal solution, and (b) a pseudo-polynomial time approximation algorithm based on dynamic programming. We also discuss techniques for handling interference. Simulations were conducted for both long-range communication (with radius around 32 m) and short-range communication (with radius around 7 m). The simulation results from the scenarios we studied show that compared with the classic technique that transmits the packets at the highest speed and shut down the radio afterwards, between 20% to 90% energy savings can be achieved by our techniques, under different settings of several key system parameters. We also develop an on-line distributed protocol that needs only local information of the aggregation tree. Simulation results show that between 15% to 90% energy conservation can be achieved by the on-line protocol. The adaptability of the protocol is also demonstrated through several run-time scenarios.

**Related work:** The most relevant works include [4]–[6], [8], [9]. The problem of minimizing the energy dissipation for

This work is supported by NSF under grant IIS-0330445 and by an ITR grant under award number 0325875.

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 [5], an on-line policy for adjusting modulation level is proposed for single-hop communication. In [8], modulation scaling is integrated into the Weighted Fair Queuing (WFQ) scheduling policy. 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 apply the techniques in [4] and [6], as they require explicit latency constraints over each link. Lastly, we consider non-monotonic energy functions, which has not been previously addressed.

**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. Simulation results are shown in Section VI. Finally, concluding remarks are made in Section VII.

## II. BACKGROUND

### A. Data Aggregation Paradigm

We abstract the underlying structure of the network as a data aggregation tree. This is essentially a tree that aggregates and gathers information from multiple sources enroute to the 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, 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  $k = 0$ , the lowest. 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  $k = 0$ .

$$s' = \frac{ds}{dk - k + 1} \quad (1)$$

We consider the scenario that the data gathering must be completed within a specified latency constraint, which is necessary for real-time monitoring or mission-critical applications. To enforce the latency constraint requires the use of time-synchronization schemes such as [11]. The recently proposed epoch-based scheme (refer to [10]) instantiates the latency constraint by the length of each epoch. 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.

### B. Non-Monotonic Energy Model

We model the transmission energy using the example of modulation scaling [5] based on Quadrature Amplitude Modulation (QAM) scheme [12]. Note that the algorithms presented in this paper are extendible to other modulation schemes as well as other techniques that provide energy-latency tradeoffs, such as code scaling [13]. Consider a packet of  $s$  bits to be transmitted between two sensor nodes. Assuming that the symbol rate,  $R$ , is fixed, the transmission time,  $\tau$ , can be calculated as [5]:

$$\tau = \frac{s}{b \cdot R}, \quad (2)$$

where  $b$  is the modulation level of the sender in terms of the constellation size (number of bits per symbol). The corresponding transmission energy can be modeled as the sum of output energy and electronics energy. Though the transmission energy essentially depends on the setting of  $b$ , we would like to characterize it as a function of  $\tau$  [5], denoted as  $w(\tau)$ , to illustrate the key energy-latency tradeoffs in this paper.

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

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 electronics power,  $P_e$ , can be modeled as follows [5]:

$$P_o = C \cdot R \cdot (2^b - 1) \text{ and} \quad (4)$$

$$P_e = F \cdot R. \quad (5)$$

We consider the radio modules from [7], [14]. Typically, for short-range communication with  $R = 1$  Mbaud, the electronics power of the radio is approximately 10 mW, while the output power is approximately 1 mW (at 4-QAM). From equations (4) and (5), it can be derived that  $C \approx 3 \times 10^{-10}$  and  $F = 10^{-8}$ . Further, we consider a  $d^2$  power loss model, where  $d$  is the communication radius. Assuming that it takes 10 pJ/bit/m<sup>2</sup> by the amplifier to transmit one bit at an acceptable quality [15], we infer that the desired communication radius is  $\sqrt{50} \approx 7$  m (from  $\frac{1 \text{ mW}}{2 \times 10^6 \text{ bit/sec}} = 10 \text{ pJ/bit/m}^2 \times d^2$ ). In our study, we consider one more case of communication in WSNs – long-range communication with radius at  $\sqrt{1000} \approx 32$  m 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 in the figure), resulting in discrete values of  $\tau$ . It can be observed that the the transmission energy for the short-range communication eventually increases after the transmission time exceeds 300 nSec. 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.

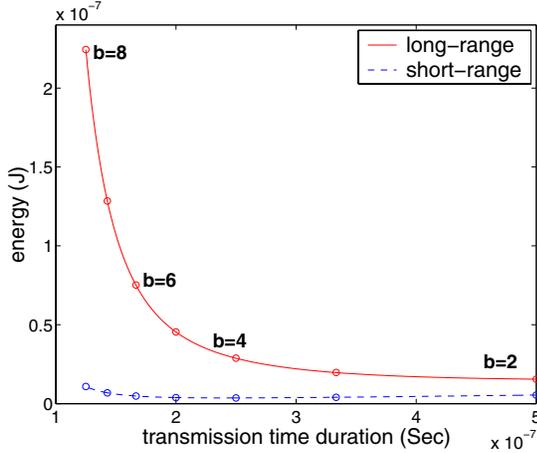


Fig. 1. Energy-latency tradeoffs for transmitting one bit data

### III. PACKET TRANSMISSION PROBLEM OVER DATA AGGREGATION TREES

#### A. Data Aggregation Tree

Let  $T = (V, E)$  denote the data aggregation tree, where  $V$  denotes the set of  $n$  sensor nodes,  $\{V_i : i = 1, \dots, n\}$ , and  $E$  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 indexed in the topological order with  $V_1, \dots, V_M$  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  $P_i$ .

Raw data is generated by a set of source nodes from  $V$  (not necessarily leaf 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,  $V_i, i \in \{1, \dots, M\}$ , 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 are completely 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

Let  $\Gamma$  denote the 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.

A schedule of packet transmission is defined as a vector  $\vec{\tau} = \{\tau_i : i = 1, \dots, n-1\}$ , where  $\tau_i$  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 each  $p_i \in T$ , we have  $L_i \leq \Gamma$ .

Our goal 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.

Let  $w_i(\tau)$  denote the energy function of sensor node  $V_i$ , with  $m_i$  denoting the value of  $\tau \in (0, T]$  when  $w_i(\cdot)$  is minimized. Moreover, by assuming a first order energy model, the reception energy can be modeled by doubling the value of  $F$  in equation (3). Thus, we state the *packet transmission problem* (called PTP) as follows:

**Given:**

- a data aggregation tree  $T$  consisting of  $n$  sensor nodes,
- energy functions for each link  $(i, j) \in E$ ,  $w_i(\tau)$ , and
- the latency constraint,  $\Gamma$ ;

**find** a schedule of packet transmission,  $\vec{\tau}$ , so as to minimize

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

**subject to**

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

The above formulation differs from the problem defined in [6] in two key aspects. (1) We employ a tree structure packet flow where the latency constraint is imposed on each path of the tree. (2) The non-monotonic energy model in Section II-B indicates an upper-bound on the transmission time of each packet, i.e., to optimize PTP, we should have  $\tau_i \leq m_i$ , for each  $i = 1, \dots, n-1$ . The consequences of the above differences are discussed in Section IV-A.

We note that the above model assumes no MAC layer interference, which can be realized by multi-packet reception

---

**Begin**

- |   |  |
|---|--|
| 1. Set $k \leftarrow 0$<br>2. <b>For</b> $(i, n) \in E$ , set $\tau_i^k \leftarrow \min\{\Gamma, m_i\}$<br>3. <b>For</b> $(i, j) \in E$ such that $j \neq n$ , set $\tau_i^k \leftarrow 0$<br>4. Set $flag \leftarrow 0$<br>5. <b>While</b> $flag = 0$<br>6. $k \leftarrow k + 1$<br>7. <b>For</b> each $V_i$ with $i$ from $n - 1$ downto $M+1$<br>8. $(\{\tau_j^k\}_{(j,i) \in E}, \tau_i^k) \leftarrow \text{best}(\{\tau_j^{k-1}\}_{(j,i) \in E}, \tau_i^{k-1})$<br>9. <b>For</b> $(i, n) \in E$<br>10.       Set $\tau_i^k \leftarrow \min\{m_i, \Gamma - (\max_{V_i \in P_j} \{L_j\} - \tau_i^k)\}$<br>11.   if $\bar{\tau}^k = \bar{\tau}^{k-1}$ , $flag \leftarrow 1$ | // initialize iteration counter<br>// initialize transmission time for links to the sink<br>// initialize transmission time for other links<br>// flag to keep track of convergence in the iterations<br><br>// increment the iteration counter by 1<br>// perform local optimization for each internal node<br>// move right the start time of transmission from $V_i$<br><br>// increase the transmission time for links to the sink<br>// check convergence |
|---|--|

**End**


---

Fig. 2. Pseudo code for EMR-Algo

(MPR) techniques [16]. We further elaborate this issue in Section IV-C.

#### 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 an optimal solution for OPTP. A faster dynamic programming based approximation algorithm is then presented. Techniques for handling interference are also discussed.

##### A. A Numerical Optimization Algorithm

Since we must have  $\tau_i \leq m_i$  in an optimal solution to OPTP, the latency of a path does not necessarily equal  $\Gamma$ . We show the following necessary and sufficient condition for the optimality of the OPTP problem.

*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  $V_i$  equals  $\Gamma$ ; and
- 2) for any internal node,  $V_i$ , we have

$$\dot{w}_i(\tau_i^*) = \sum_{(j,i) \in E} \dot{w}_j(\tau_j^*). \quad (8)$$

*Corollary 1:* Consider an optimal schedule,  $\vec{\tau}^*$ , for OPTP; the following hold:

- 1) If  $\tau_i^* = m_i$  for some  $V_i \in V$ , we have  $\tau_j^* = m_j$  for all sensor nodes in  $T_i$ .
- 2) If  $\tau_i^* < m_i$  for some  $V_i \in V$ , we have  $\tau_j^* < m_j$  for all ancestors of  $V_i$ .

Due to space limitations, the proof of Lemma 1 and Corollary 1 is omitted in this paper. Details of the proof can be found in [17].

In this section, we extend the MoveRight algorithm from [6] to solve OPTP in a general-structured aggregation tree with non-monotonic energy functions. The pseudo code for the

extended MoveRight algorithm (EMR-Algo) is shown in Figure 2. In the figure,  $\tau_i^k$  denotes the value of  $\tau_i$  after the  $k$ -th iteration. Initially, we set the starting time for all packet transmission to zero – the transmission time for all the 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 our objective function. Finally, this iterative local optimization leads to a globally optimal solution.

The  $\text{best}(\cdot)$  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  $\tau_i^k$  must lie within  $(0, \tau_i^{k-1}]$ , the  $\text{best}(\cdot)$  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 of all paths is reached.

The proposed EMR-Algo is distinguished from the MoveRight algorithm in two key respects. (recall the differences between our problem and the one defined in [6]). (1) The  $\text{best}(\cdot)$  function respects Lemma 1 regarding the optimality of OPTP in a tree structure. (2) The transmission time for any  $V_i \in V$  is bounded by  $m_i$ , enforced by lines 2, 8 and 10.

The correctness of EMR-Algo can be proved by exploring the convexity property of the energy functions. Let  $\vec{\tau}^* = \{\tau_1^*, \dots, \tau_{n-1}^*\}$  be the optimal schedule. Let  $s_i^* = 0$ , for  $i = 1, \dots, M$ ; and  $s_i^* = \max_{(j,i) \in E} (s_j^* + \tau_j^*)$ , for  $i = M+1, \dots, n-1$ . As previously stated,  $\{\tau_1^k, \dots, \tau_{n-1}^k\}$  indicate the transmission time of nodes  $V_1, \dots, V_{n-1}$  after the  $k$ -th pass of EMR-Algo. Let  $s_i^k = 0$ , for  $i = 1, \dots, M$ , and  $s_i^k = \max_{(j,i) \in E} (s_j^k + \tau_j^k)$ , for  $i = M, \dots, n-1$ . We have:

*Theorem 1:* Let  $s_i^k$  and  $s_i^*$ ,  $i = 1, \dots, n-1$  be as defined above. Then

- 1)  $s_i^k \leq s_i^{k+1}$ ;
- 2)  $s_i^k \leq s_i^*$ ; and
- 3)  $s_i^\infty = s_i^*$ .

The proof of Theorem 1 is presented in Appendix I.

The convergence speed of EMR-Algorithm depends on the structure of the aggregation tree and the exact form of the energy functions. It is therefore difficult to give a theoretical upper bound on the number of iterations. In Section VI, we show the running time of EMR-Algorithm for simulated problems. However, by approximating  $w_i(\tau)$  with a set of interpolated discrete values, we develop a pseudo-polynomial time 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  $\tau$ . Let  $\varepsilon$  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  $T_i$  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  $\varepsilon, 2\varepsilon, \dots, t$ . Therefore, we have the following recursive representation of  $g(V_i, t)$ :

$$g(V_i, t) = \begin{cases} w_i(t), & \text{for } 1 \leq i \leq M \\ \sum_{(k,i) \in E} (\min_{j=1}^{\frac{t}{\varepsilon}} \{w_k(j\varepsilon) + g(V_k, t - j\varepsilon)\}), & (9) \\ \text{otherwise.} \end{cases}$$

The above representation is suitable for a dynamic programming based algorithm (DP-Algorithm for short). DP-Algorithm 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-Algorithm, the cell crossed by the  $j$ -th row and the  $i$ -th column shall contain the value of  $g(V_i, j\varepsilon)$ .

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  $\frac{1}{\varepsilon^2}$ .

**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  $\tau_i$ 's resulted from different modulation levels are not evenly distributed among  $[0, \Gamma]$ . Thus, DP-Algorithm cannot be directly applied. However, one practical method is to, for each  $i$ , set  $\tau_i$  obtained by EMR-Algorithm or DP-Algorithm to the largest time duration below

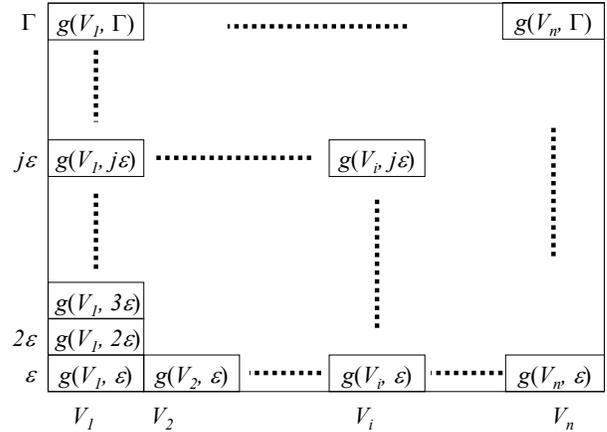


Fig. 3. The  $g(\cdot)$  table computed by DP-Algorithm

$\tau_i$  that can be achieved by an available modulation level. We call the above method the *rounding procedure*.

### 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 [16]. 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.

A more systematic way is to carefully schedule the transmission of sensor nodes that can potentially interfere (or simply interfere) with each other. The goal is to ensure that the corresponding time periods for a group of interfering sensor nodes do not overlap with each other. Intuitively, children of a sensor node are interfering – they cannot send packets to the parent at the same time. In the following, we describe a modified DP-Algorithm under the hypothesis that any group of interfering sensor nodes are children of the same node. Such a hypothesis can be supported by carefully reconstructing the aggregation tree (refer to Appendix II for details).

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 consecutive 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 packets transmitted to  $V_i$  from its children are then scheduled in the second part. Hence, the order of packet transmission in the second part has no effects on the packet scheduling in the first part. The optimal division of the latency constraint over  $T_i$  can be found using dynamic programming with the following recursive representation of  $g(V_i, t)$ :

$$g(V_i, t) = \begin{cases} w_i(t), & \text{for } 1 \leq i \leq M \\ \min_{j=1}^t \{z(i, j\varepsilon) + \sum_{(k,i) \in E} g(V_k, t - j\varepsilon)\}, & (10) \\ \text{otherwise.} \end{cases}$$

The function  $z(i, j\varepsilon)$  returns a schedule for the packets from children of  $V_i$  within time duration  $j\varepsilon$  so that run-time contentions can be avoided. 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  $\varepsilon$ . 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  $\varepsilon$ . Note that the energy gradient can be negative due to the non-monotonic energy functions. We then increase the transmission time of the child with the maximal positive energy gradient by  $\varepsilon$ . 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  $O((\frac{1}{\varepsilon})^2(|V| + |E|))$ .

## V. DISTRIBUTED ON-LINE PROTOCOL

The algorithms presented in Section IV 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  $V_i$  can derive the time cost for data gathering within subtree  $T_i$ .
- 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  $\varepsilon$ .
- 4) Interference among sensor nodes is minimized by using either MPR techniques or MAC layer scheduling.

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 (we shall be assuming that time synchronization schemes, such as [11], are available). In assumption 3, 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. Assumption 4 only applies to each group of interfering nodes, which are children of the same sensor node from the hypothesis in Section IV-C. If the interference is handled by MAC layer scheduling, the incurred time cost

for sequentializing the packet transmissions in each group of interfering nodes can be accommodated by intentionally reducing the latency constraint. When MPR technique is used, however, there is no impact on the latency constraint.

In the following, we first describe the local data structure maintained at each sensor node. A distributed adaptation policy for minimizing the energy dissipation is then proposed. **Local Data Structure:** Each sensor node,  $V_i$ , maintains a simple local data structure  $(r, \tau_i, \tau_d)$ . The flag  $r$  equals one if  $V_i$  is the node with the highest *positive* energy gradient in the subtree rooted at  $V_i$ ,  $T_i$ , and zero otherwise. Field  $\tau_i$  is the time cost for transmitting the packet from  $V_i$  to its parent, while  $\tau_d$  records the time cost for data gathering within subtree  $T_i$  (excluding  $\tau_i$ ).

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 sent to the parent of  $V_i$ . The above procedure continues till all the sensor nodes have the correct value of  $r$ . Fields  $\tau_i$  and  $\tau_d$  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  $\tau_d$  and the difference between  $\Gamma$  and  $\tau_d$ , denoted as  $\delta$ . 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  $\tau_d$  in  $V_i$ 's local data, let  $\tau'_d$  denote the field  $\tau_d$  in the feedback packet.

- 1) If  $\delta < 0$ , the transmission time for packet from  $V_i$  is decreased by a factor of  $\beta$ , where  $\beta$  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  $\varepsilon$ . 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.

**Discussion:** During each dissemination of the feedback packet, the proposed on-line protocol increases the transmission time for at most one sensor node per path. Such an increment is guaranteed not to violate the latency constraint for each path. Therefore, the on-line protocol converges when the transmission latency of all paths reach the latency constraint,

or for each  $V_i \in V$ , we have  $\tau_i = m_i$ . We assume that each sensor node has  $q$  available modulation settings. Before the protocol converges, a feedback packet would reduce the modulation setting for at least one sensor node every time it traverses the aggregation tree. Thus, the protocol converges after the dissemination of at most  $nq$  feedback packets, where  $n$  is the number of sensor nodes in the aggregation tree.

Various tradeoffs can be explored in implementing the above protocol. Ideally, the adaptation should be performed under a stable system state. Thus, the period,  $\alpha$ , 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  $\beta$ . A larger value of  $\beta$  leads to higher transmission speed when the latency constraint is violated. However, extra energy dissipation is caused if the violation is not dramatic. Intuitively,  $\beta$  should be related to the severity of the violation, which is indicated by the value of  $\delta$ .

Another option to handle latency violations is to repeatedly reduce the transmission time of the sensor nodes with the smallest energy gradient till the latency constraint is satisfied. Such an option is more aggressive in reducing incurred energy cost. However, it requires more complex control protocol and more importantly, increases the response time in handling latency violations.

## VI. SIMULATION RESULTS

To conduct the simulations, a simulator was developed using the PARSEC [18] software, which is a discrete-event simulation language. The baseline in our simulations is that all sensor nodes send the packet at the highest speed and shutdown the radio afterward. This policy is proposed, for example, in the PAMAS protocol [19]. 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 simulations 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. Simulation 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) sensor nodes are randomly selected to be the sources. The Greedy Incremental Tree (GIT) algorithm [3] was used for constructing the data aggregation tree. An example data aggregation tree is illustrated in Figure 4 with  $\rho$  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.

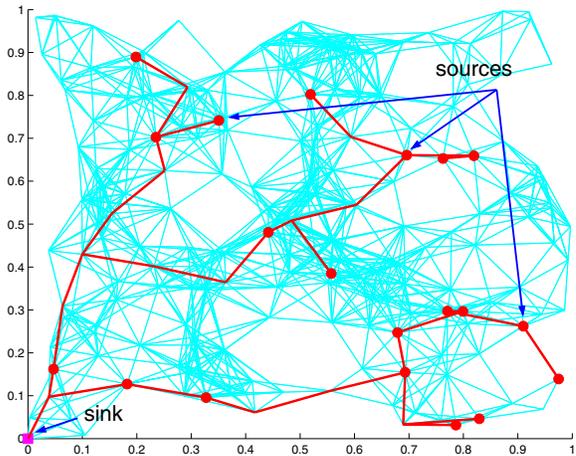


Fig. 4. A example data aggregation tree based on the RS model (connectivity parameter  $\rho = 0.15$ , number of sources  $N = 20$ )

The energy function used in the simulation was in the form of equation (3). Note that  $\tau_i$ 's are assumed to be continuous variables except in the special case of modulation scaling, when  $\tau_i$ '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  $7 \times 10^{-9}$  for the long-range communication and  $3 \times 10^{-10}$  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 adjust the tightness of the latency constraint,  $\Gamma$ . 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. We use equation (2) to model the time cost for packet transmission, with the highest value of  $b_i$  equal to 8. The minimal time cost for data gathering,  $t_{min}$ , was calculated by assuming  $b_i = 8$  for all the sensor nodes. Then, we set  $\Gamma = \frac{t_{min}}{u}$ .

### 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 simulation, the number of sources,  $N$ , was varied from 10 to 30 in increments of 10. The connectivity parameter,  $\rho$ , 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 simulation, more than 100 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 network shown in Figure 4.

**Performance of Our Off-Line Algorithms:** Figure 5 demonstrates the energy saving achieved by our off-line algorithms for both long and short range communication. The investigated algorithms include EMR-Algo, DP-Algo (with  $D = 100$  or  $50$ ), and the special case of DP-Algo for modulation scaling with  $D = 100$  (denoted as MS). 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 in this case, to round the approximated solutions from DP-Algo leads to infeasible solutions.

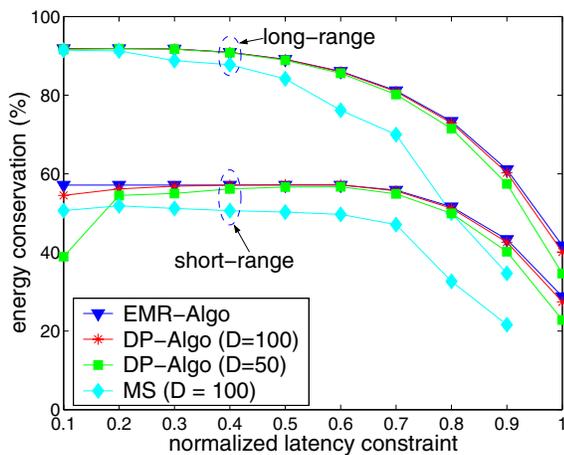


Fig. 5. Performance of our off-line algorithms (connectivity parameter  $\rho = 0.15$ , number of sources  $N = 20$ , aggregation factor  $k = 0.7$ )

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 with  $D = 100$  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, no energy can be saved when  $u = 1$ . 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 even when  $u = 1$ .

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 simulation results show that the DP-Algo with  $D = 100$  achieved at least 92% of the performance of EMR-Algo in all the simulated instances. However, the performance of MS quickly degrades when  $u$  increases. This is because the first derivative of the energy function tends to  $-\infty$  as  $\tau$  tends to 0. Thus, the performance loss due to the rounding procedure

becomes large with solutions having high modulation levels, 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 constraint increases. Simulation results show that such 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, the energy gain of DP-Algo decreases when  $k$  decreases for a fixed  $u$ . 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 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.

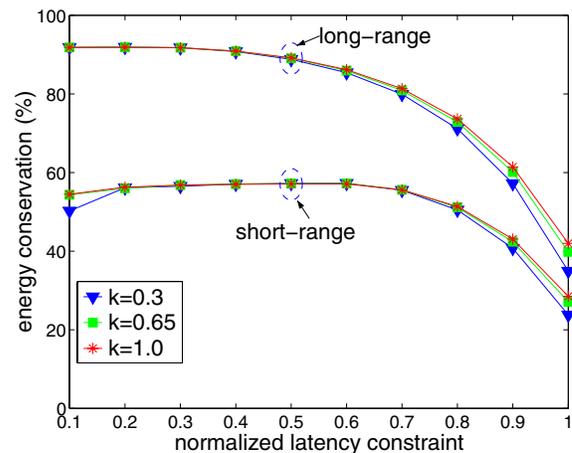


Fig. 6. Energy conservation versus normalized latency constraint and aggregation factor  $k$  (connectivity parameter  $\rho = 0.15$ , approximation accuracy  $D = 100$ , number of sources  $N = 20$ )

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  $\rho$ . It can be observed that the energy saving of DP-Algo increases when  $\rho$  increase. This

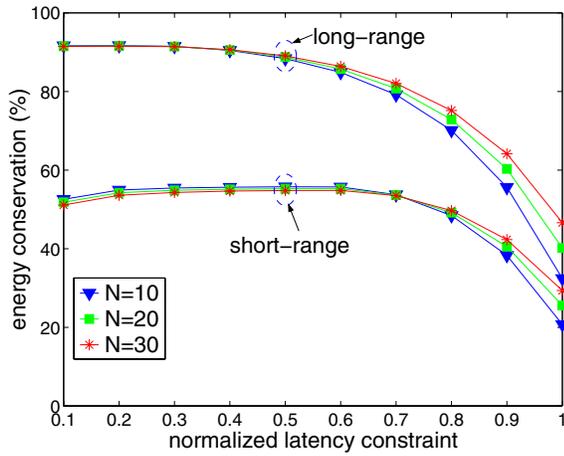


Fig. 7. Energy conservation versus normalized latency constraint and number of sources  $N$  (connectivity parameter  $\rho = 0.15$ , approximation accuracy  $D = 100$ , aggregation factor  $k = 0.7$ )

is understandable since large  $\rho$  reduces the height of the aggregation tree (the extreme case is a star-like aggregation tree formed by setting  $\rho = 1$ ).

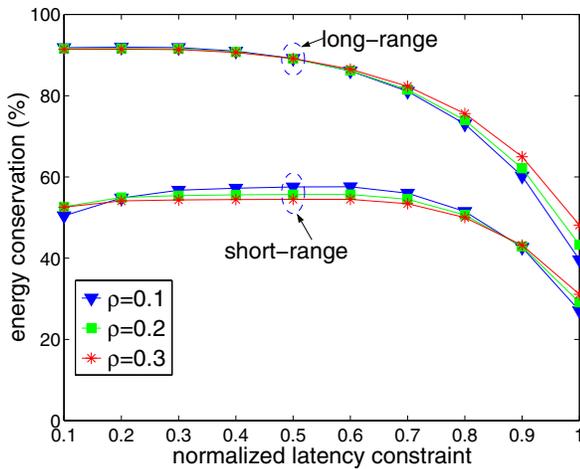


Fig. 8. Energy conservation versus normalized latency constraint and connectivity parameter  $\rho$  (approximation accuracy  $D = 100$ , aggregation factor  $k = 0.7$ )

Together, the above results suggest that when  $u$  is small, DP-Algo is quite robust with respect to variations in different system parameters, including  $k$ ,  $\rho$ , and  $N$ .

**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 gain is observed for the long-range communication when  $u = 1$ . Thus, it is worthwhile to employ mechanisms for multi-packet reception or develop more efficient algorithms for handling interference.

**Impact of the Start-up Energy:** We estimate the energy for starting up the radio as  $1 \mu\text{J}$  [7]. In each epoch, the radio of each sensor node is started at most once. In addition, to emphasize the impact of start-up energy, we change the

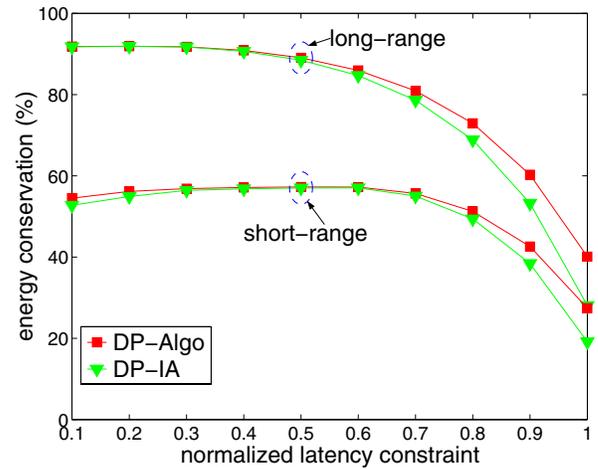


Fig. 9. Performance of the DP-IA algorithm (connectivity parameter  $R = 0.15$ , aggregation factor  $k = 0.7$ , number of sources  $N = 20$ )

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  $1 \mu\text{J}$  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.

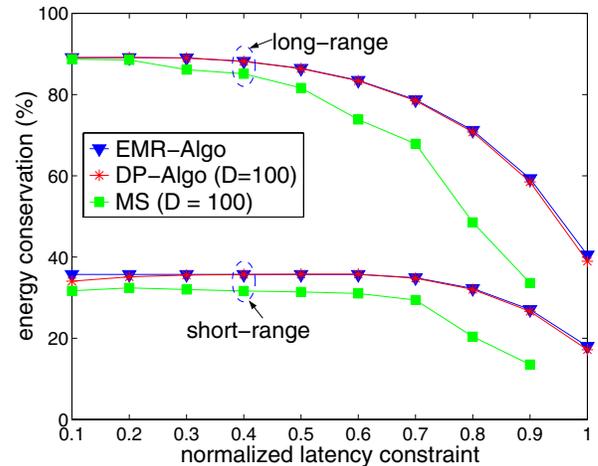


Fig. 10. Performance of off-line algorithms with start-up energy (connectivity parameter  $\rho = 0.15$ , aggregation factor  $k = 0.7$ , number of sources  $N = 20$ )

### C. Performance of the On-Line Protocol

**Energy Conservation:** We show the energy conservation achieved by the on-line protocol in Figure 11. The presented data is averaged over more than 150 problem instances and has a 95% confidence interval with a 5% (or better) precision. In each instance, we generated a sensor network with 200 randomly dispersed sensor nodes. After randomly selecting 20 source nodes, the data aggregation tree was then generated using GIT.

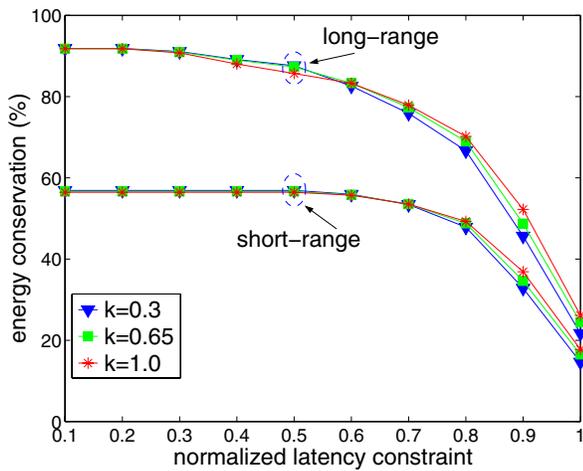


Fig. 11. Performance of the on-line protocol versus normalized latency constraint and aggregation factor  $k$  (connectivity parameter  $\rho = 0.15$ , number of sources  $N = 20$ )

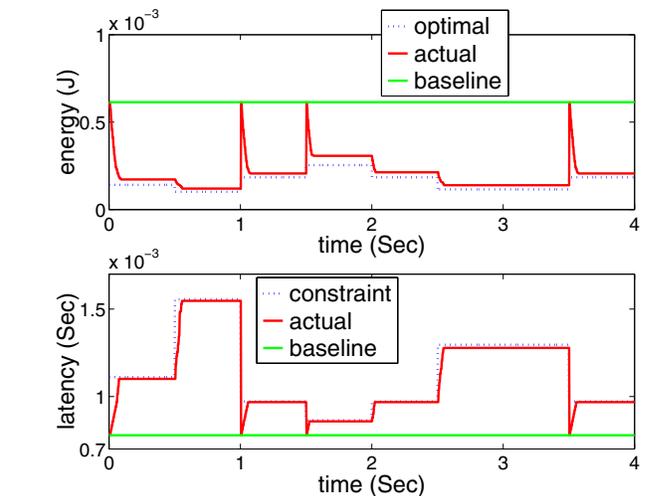
It can be seen that the energy conservation achieved by the on-line protocol is quite close to the performance of the off-line algorithms shown in Figure 5. There is observable performance degradation when the normalized latency constraint is set to one – around 15% less energy conservation for long-range communication and 10% less for short-range communication. Such a performance degradation is reasonable due to the fact that only 4 options are available to set the transmission time for each sensor in the on-line protocol, instead of the fine granularity adjustment of the transmission time in the off-line algorithms. Surprisingly, the on-line protocol actually outperforms the modulation scaling case (MS) shown in Figure 5, implying a large performance degradation of the rounding technique used by MS.

**Adaptability to System Variations:** Our simulations were performed based on the aggregation tree shown in Figures 4 that has 35 sensor nodes, out of which, 20 are source nodes. We assume that modulation scaling is used by all the nodes with the available modulation levels being even numbers between 2 and 8. The data gathering was requested every 2 mSec. For the sake of illustration, we set  $\alpha = 4$  mSec, and  $\beta = 10$ .

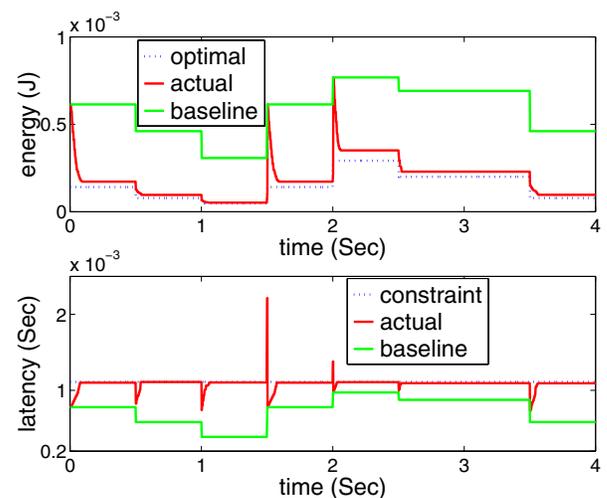
Two run-time scenarios, namely A and B, were investigated to demonstrate the efficiency and adaptability of our protocol. The energy cost and latency for data gathering over 4 seconds are depicted in Figure 12, where the optimal solutions are obtained by using EMR-Algo.

*Scenario A:* We fixed  $s$  at 400 bits, while setting  $u$  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.

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,



(a) Scenario A



(b) Scenario B

Fig. 12. Adaptability of the on-line protocol (connectivity parameter  $\rho = 0.15$ , aggregation factor  $k = 0.7$ )

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 the highest levels when a violation is detected, reflected by the high peaks in the energy curve at time 1, 1.5 and 3 seconds.

*Scenario B:* We set  $\Gamma = 0.6$  mSec, while setting  $s$  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. 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. It is also capable of adapting the packet transmission time with respect to the changing system parameters.

## VII. CONCLUDING REMARKS

In this paper, we have studied the problem of scheduling packet transmissions over a data aggregation tree in wireless sensor networks by exploring the energy-latency tradeoffs. For the off-line version of the problem, we have provided (a) a numerical algorithm for optimal solutions, and (b) a pseudo-polynomial time approximation algorithm based on dynamic programming. Techniques for handling interference has also been discussed. Our simulation results show that between 20% to 90% energy saving can be achieved by the algorithms. We have investigated the performance of our algorithms with different settings of several key system parameters. We have also proposed a distributed on-line protocol that relies only on local information of each sensor node in the aggregation tree. Our simulation results show that the energy saving achieved by the protocol is between 15% to 90%. Also, the ability of the protocol to adapt the packet transmission time upon variations in the system parameters has been demonstrated through several run-time scenarios.

We are interested in integrating the concept of adaptive fidelity computation [2] for aggregation and compression into our work. 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. B. Srivastava, "Instrumenting the world with wireless sensor networks," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2001.
- [2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A scalable and robust communication paradigm for sensor networks," in *ACM/IEEE International Conference on Mobile Computing and Networking (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. B. 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] V. Raghunathan, S. Ganerwal, C. Schurgers, and M. B. Srivastava, "E2WFQ: An energy efficient fair scheduling policy for wireless systems," in *International Symposium on Low Power Electronics and Design (ISLPED'02)*, Aug. 2002, pp. 30–35.
- [9] Y. Yu and V. K. Prasanna, "Energy-balanced multi-hop packet transmission in wireless sensor networks," in *IEEE 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 *Symposium on Operating Systems Design and Implementation (OSDI)*, Dec. 2002.
- [11] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Symposium on Operating Systems Design and Implementation (OSDI)*, 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," *IEEE Trans. on Vehicular Technology*, vol. 47, no. 4, pp. 1134–1147, Nov. 1998.
- [13] E. Armaniou, D. D. Falconer, and H. Yanikomeroglu, "Adaptive modulation, adaptive coding, and power control for fixed cellular broadband wireless systems," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2003.

- [14] A. Wang, S.-H. Cho, C. G. Sodini, and A. P. Chandrakasan, "Energy-efficient modulation and MAC for asymmetric microsensor systems," in *ISLPED*, 2001.
- [15] W. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application specific protocol architecture for wireless microsensor networks," *IEEE Trans. on Wireless Networking*, 2002.
- [16] S. Roy and H. Y. Wang, "Performance of CDMA slotted ALOHA multiple access with multiuser detection," in *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2, 1999, pp. 839–843.
- [17] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Exploring energy-latency tradeoffs for data gathering in wireless sensor networks," Department of Electrical Engineering, University of Southern California, Tech. Rep. CENG-2003-05, 2003.
- [18] PARSEC Project. [Online]. Available: <http://pcl.cs.ucla.edu/projects/parsec>
- [19] C. S. Raghavendra and S. Singh, "PAMAS – power aware multi-access protocol with signaling for ad hoc networks," *Computer Communication Review*, July 1998.

## APPENDIX I PROOF OF THEOREM 1

We define the level of a tree as the greatest number of edges contained by any path in the tree. We consider an OPTP problem with a two-level aggregation tree that has exactly one internal node with  $p$  children (see Figure 13). We call such a problem 2-Lev-OPTP. Let  $V_{p+1}$  denote the internal node, with  $V_{p+2}$  denoting its parent and  $\mathcal{C} = \{V_1, \dots, V_p\}$  denoting the set of children. We assume that for any  $V_i \in \mathcal{C}$ , a packet is ready to transmission at time  $s_i$  and  $V_{p+2}$  must received aggregated information from  $V_{p+1}$  by time  $t$ .

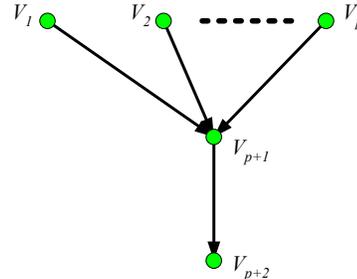


Fig. 13. A problem instance of 2-Lev-OPTP

We first show the following lemma:

*Lemma 2:* Let  $\vec{\tau}^* = \{\tau_1^*, \dots, \tau_{p+1}^*\}$  denote an optimal schedule to the 2-Lev-OPTP problem as defined above, then the following hold:

- 1) The schedule  $\vec{\tau}^*$  is unique.
  - 2) Let  $s_{p+1}$  denote the start time of packet transmission from  $V_{p+1}$  to  $V_{p+2}$  in the optimal schedule, i.e.,  $s_{p+1} = \max_{V_i \in \mathcal{C}} (s_i + \tau_i^*)$ . Then  $s_{p+1}$  never decreases when (a) some  $s_i$ 's,  $V_i \in \mathcal{C}$ , increase, holding  $t$  fixed; or (b)  $t$  increases, holding  $s_i$ 's fixed, for all  $V_i \in \mathcal{C}$ ; or (c) both some  $s_i$ 's and  $t$  increase.
- Let  $\mathcal{D}$  denote the set of sensor nodes that increase their transmission start time in cases (a) and (c). Then, in particular,  $s_{p+1}$  increases in case (a) if for any  $V_i \in \mathcal{D}$ , we have  $s_{p+1} - s_i \leq m_i$ ; or in case (b) we have  $t - s_{p+1} < m_{p+1}$ ; or in case (c) we have either of the previous two conditions hold.
- 3)  $s_{p+1}$  never increases when (a) some  $s_i$ 's,  $V_i \in \mathcal{C}$ , decrease, holding  $t$  fixed; or (b)  $t$  decreases, holding  $s_i$ 's fixed, for all  $V_i \in \mathcal{C}$ ; or (c) both some  $s_i$ 's and  $t$  decrease.

Let  $\mathcal{D}'$  denote the set of sensor nodes that decrease their transmission start time in cases (a) and (c). Then, in particular,  $s_{p+1}$  decreases in case (a) if for any  $V_i \in \mathcal{D}'$ , we have  $s_{p+1} - s_i < m_i$ ; or in case (b) we have  $t - s_{p+1} \leq m_{p+1}$ ; or in case (c) we have either of the previous two conditions hold.

Due to space limitations, the proof of Lemma 2 is omitted in this paper. Details of the proof can be found in [17].

Now we prove Theorem 1.

*Proof:*

- 1) Recall that EMR-Algo works in iterations: for each iteration  $k$ , the algorithm determines  $s_i^k$  by decreasing  $i$  from  $n - 1$  to  $M + 1$ . Since the EMR-Algo initializes  $s_i = 0$  for  $i = 1, \dots, n - 1$ , it follows that  $s_i^0 \leq s_i^1$  for each  $i = 1, \dots, n - 1$ . Suppose that  $i' > 1$  and  $k' > 1$  are the first time that there is a violation; that is,  $s_{i'}^{k'} > s_{i'}^{k'+1}$ . We consider the 2-level aggregation tree formed by  $V_{i'}$  together with its parent, denoted as  $V_p$ , and its children, denoted as set  $\mathcal{C}$ . We have  $s_p^{k'} \leq s_p^{k'+1}$ , and  $s_i^{k'-1} \leq s_i^{k'}$ , for each  $V_i \in \mathcal{C}$ .
- 2) It is obvious that  $s_i^0 \leq s_i^*$ , for each  $i = 1, \dots, n - 1$ . Similar to the proof for property (1), suppose that  $i' \geq 1$  and  $k' \geq 1$  are the first time that there is a violation; that is,  $s_{i'}^{k'} > s_{i'}^*$ .

Again, we consider the 2-level aggregation tree formed by  $V_{i'}$  together with its parent, denoted as  $V_p$ , and its children, denoted as set  $\mathcal{C}$ . We have  $s_p^{k'} \leq s_p^*$ , and  $s_i^{k'-1} \leq s_i^*$ , for each  $V_i \in \mathcal{C}$ . We know that  $s_p^{k'}$  and  $s_i^{k'-1}$ 's actually give the boundaries within which EMR-Algo determines  $s_{i'}^{k'}$ . Similarly,  $s_p^*$  and  $s_i^*$ 's give the boundaries within which EMR-Algo determines  $s_{i'}^*$ . Part (2) of Lemma 2 again leads to the contradiction that  $s_{i'}^{k'} \leq s_{i'}^*$  and proves property (2).

- 3) We prove by contradiction and hence assume that  $j = \max\{i : s_i^\infty < s_i^*\}$ . Let  $V_p$  denote the parent of  $V_j$  and  $V_g$  denote the parent of  $V_p$ . We have  $s_p^\infty = s_p^*$  and  $s_g^\infty = s_g^*$ . Since  $\tau_j^*$  is optimal, we have  $\tau_j^* \leq m_j$ . We consider two cases:

*Case (i):* We suppose that  $\tau_j^* < m_j$ . Considering the 2-level tree formed by  $V_p$ ,  $V_g$  and the children of  $V_p$ , denoted as  $\mathcal{C}$ , we have  $s_j^\infty < s_j^*$  and  $s_i^\infty \leq s_i^*$ , for each  $V_i \in \mathcal{C} \wedge i \neq j$ . Suppose that we run EMR-Algo for one more pass and let  $\hat{s}_p$  denote the resulting start time for the transmission from  $V_p$  to  $V_g$ . From part (3) of Lemma 2, we have  $s_p^\infty - (s_j^\infty - s_j^*) < \hat{s}_p < s_p^\infty = s_p^*$ , contradicting both property (1) for  $V_p$  and the definition of  $j$ .

*Case (ii):* We assume that  $\tau_j^* = m_j$ . From part (1) of Corollary 1, we have  $\tau_i^* = m_i$  for any  $V_i \in T_j$ . Moreover, we have  $s_p^\infty - s_j^\infty > s_p^* - s_j^* = \tau_j^* = m_j$ . Since EMR-Algo maintains the invariant that  $\tau_i^k \leq m_i$  for all  $V_i \in V$ , we have  $\tau_j^\infty = m_j$ . Again from part (1) of Corollary 1,

we have  $\tau_i^\infty = m_i$  for any  $V_i \in T_j$ . Based on the definition of  $s_j^\infty$  and  $s_j^*$ , we obtain the contradiction that  $s_j^\infty = s_j^*$ . ■

## APPENDIX II

### JUSTIFICATION OF THE HYPOTHESIS FOR DP-IA

The DP-IA algorithm is designed based on the hypothesis that any group of interfering sensor nodes are children of the same node. Such a hypothesis can be satisfied by carefully constructing the aggregation tree as follows.

We consider possible scenarios for interference between sensor nodes that are not children of the same node. First we consider packet transmission along each path – the packets must be transmitted by following the topological order of sensor nodes along the path. Therefore, no interference could happen between sensor nodes on the same path.

Now we examine the case when sensor nodes from two different paths interfere with each other. Let  $p_1$  and  $p_2$  denote the two paths and let  $V_a \in p_1$  and  $V_b \in p_2$  denote the two interfering sensor nodes. More precisely, let  $V_c$  denote the parent of  $V_a$  and  $V_d$  denote the parent of  $V_b$  (shown in Figure 14 (a)). Let  $d(V_i, V_j)$  denote the physical distance between sensor nodes  $V_i$  and  $V_j$ . Without loss of generality, we say that  $V_b$  interferes  $V_a$  if  $d(V_b, V_c) \leq d(V_b, V_d)$ . Intuitively, it implies that  $V_c$  is within the communication range of both  $V_a$  and  $V_b$ . Hence, packets simultaneously transmitted from  $V_a$  and  $V_b$  may collide at  $V_c$ .

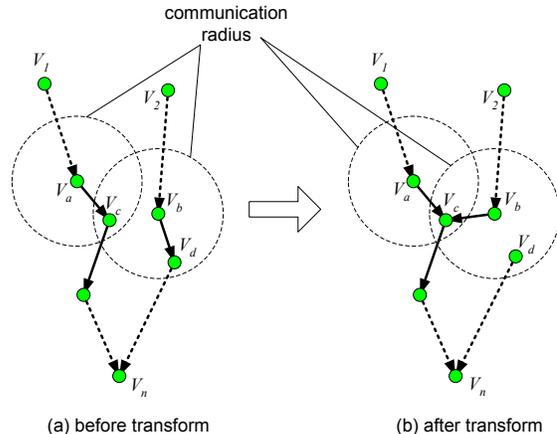


Fig. 14. A example for transforming the data aggregation tree

We use a simple transform procedure to re-construct the aggregation tree. The key idea is to group interfering sensor nodes to be the children of the same sensor node. Specifically, we make  $V_b$  also a child of  $V_c$  and break the links from  $V_b$  to its ancestors until an ancestor that performs local sensing is reached. The aggregation tree after the transformation is illustrated in Figure 14 (b).

It can be shown that by applying the above transform procedure for each pair of interfering sensor nodes from two different paths as previously defined, we obtain an aggregation tree such that any group of interfering sensor nodes are children of the same sensor node.