Exploring Energy-Latency Tradeoffs for Real-Time Data Gathering in Wireless Sensor Networks

Technical Report CENG-2003-05

Yang Yu, Bhaskar Krishnamachari, and Viktor K. Prasanna
Department of Electrical Engineering
3740, McClintock Ave., EEB
University of Southern California
Los Angeles, CA 90089-2562
(213) 821-2930, (213) 821-2528, (213) 740-4483,
{yangyu, bkrishna, prasanna}@usc.edu

Sep., 1. 2004

Exploring Energy-Latency Tradeoffs for Real-Time 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 studies 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 dynamic programming-based polynormial approximation algorithm by discretizing the transmission time. While interference can be minimized by medium access control (MAC) layer mechanisms, we also illustrate explicit packet scheduling for interference avoidance when multi-packet reception techniques (such as CDMA, FDMA) are used. Moreover, the discretized transmission time naturally leads to a simple, distributed on-line protocol that relies only on the local information available at each sensor node. Extensive simulations were conducted

This work is supported in part by NSF grants 0330445, 0325875, and 0347621.

A preliminary version of this paper appears in IEEE InfoCom 2004 [1]. This version contains additional content and improvements, including complete proofs, improved on-line protocol, and more comprehensive simulation results.

for both long-range communication (with radius around 32 m) and short-range communication (with radius around 7 m). We used the baseline of transmitting all packets at the highest speed and shutting down the radios afterward. Our simulation results from the scenarios that we studied show that compared with the baseline, up to 90% energy savings can be achieved by our techniques, under different settings of several key system parameters. The adaptability of the protocol is also demonstrated through several run-time scenarios.

Index Terms

real-time data gathering, wireless sensor networks, rate adaptation, energy efficiency, dynamic programming

I. Introduction

Wireless sensor networks (WSNs) are being developed for a wide range of civil and military applications, such as target tracking, infrastructure monitoring, habitat sensing, and battlefield surveillance [2]–[4]. In many applications for wireless sensor networks (WSNs) [3], data gathering is important for extracting useful information from the coupled environment to users. Recent studies [5], [6] 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 of the useful approaches for energy-efficient communication is to explore the energy-latency tradeoffs. An important observation in [7] 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 [8] 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 studies 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 subject to the latency constraint. Compared with [7], [9], 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 [10]. We refer to the above general model as the *non-monotonic energy model.* **Technical Approach Overview**: We solve the considered problem using two different but related approaches. In the first approach, we assume a continuously tunable transmission time. A numerical optimization algorithm is developed for solving the off-line version of our problem, where the structure of the data gathering tree and the energy characteristics of all sensor nodes are known *a priori*. While this numerical optimization provides the optimal off-line solution to our problem, it is theoretically difficult to give an upper bound on its running time.

In the second approach, we approximate the continuous transmission time using a set of discrete values. We then derive a recursive presentation of the considered problem, which naturally leads to a dynamic programming based algorithm (DP-Algo) for solving the off-line problem. For a given number of discrete values for the transmission time, DP-Algo solves the off-line problem in polynomial time.

While the communication interference in the aforementioned two solutions can be minimized by medium access layer (MAC) scheduling, we also study the application of multi-packet reception (MPR) techniques for explicit interference avoidance. To do so, we consider the so-called "scheduling constraint", which essentially enforces that at any time, any sensor node can receive packets from at most one of its children. We propose a heuristic approach to schedule the transmission from multiple children such that they do not overlap with each other.

Furthermore, a simple, localized on-line protocol is developed based on discretized transmission time. The key idea is to iteratively identify the sensor node with the highest energy cost in the gathering tree by piggybacking existing data messages and reduce its energy cost when allowed by the latency constraint. In this protocol, each sensor node only needs to perform simple operation based on its local information and the piggybacked information from data messages. The protocol is designed with the aim of self-adaptation to various dynamics in the system, including changes of packet size and latency constraint.

Finally, we evaluate the performance of our algorithms and protocol through extensive simulations. The simulations were conducted for both long and short-range communications. We considered two models of source placement, namely the random source and the event radius source placements [6]. We used the baseline where all sensor nodes transmit the packets at the highest speed (8 bits/symbol) and shut down the radio afterwards. Our simulation results from the scenarios we studied show that compared with this baseline, up to 90% energy savings can be achieved by our off-line algorithms and the on-line protocol. We also investigate the impact of several key network and radio parameters. The adaptability of the protocol is also demonstrated through two run-time scenarios.

Paper Organization: We briefly discuss the related work in Section II. We describe our underlying network model and the energy-latency tradeoffs for wireless communication in Section III. The packet transmission problem is then defined in Section IV. Off-line algorithms for the problem are presented in Section V. In Section VI, a distributed on-line protocol is described. Simulation results are shown in Section VII. Finally, concluding remarks are made in Section VIII.

II. Related work

Data gathering tree is common to data-centric routing in WSNs [5], [6]. The construction of the data gathering tree has been studied under various circumstances. For example, several localized tree topology generation mechanisms are compared by Zhou *et. al.* using metrics including node degree, robustness, and latency [11]. When the joint entropy of multiple information sources is modeled as a concave function of the number of sources, a randomized logarithmic approximation algorithm is developed by Goel *et. al.* [12]. By considering a simplified compression model, where the entropy conditioning at nodes only depends on the availability of side information, a hybrid scheme of Shortest Path Tree and Traveling Salesman Path is proved to provide 2-approximation performance for minimizing the overall cost of the data gathering tree [13]. A nice analysis of the impact of spatial correlation on several practical schemes for tree construction [14] indicates that a simple cluster-based routing scheme performs well regardless the correlation among sources. All these works provide the underlying communication substrate above which our algorithms and protocols can be applied for energy minimization.

From wireless communication perspective, rate adaptation have been widely studied to optimize spectral efficiency (e.g., network throughput) subject to the channel conditions in cellular networks [15]–[17] or local-area wireless networks [18]–[20]. Several recent works [7]–[9], [21]–[23] have studied the application of rate adaptation for energy conservation, which is closely related to our work.

For a single-hop link, the problem of minimizing the energy cost of transmitting a set of packets subject to a specified latency constraint is studied by Prabhakar *et. al.* [7]. An extension of the problem [9] investigates the packet transmission from one single transmitter to multiple receivers. In both [7] and [9], optimal off-line algorithms and near-optimal on-line solutions are provided. The concept of modulation scaling was first proposed by Schurgers *et. al.* [8]. For a single-hop link, policies for adjusting the modulation level are developed for cases where no real-time requirements are imposed [8] or each packet delivery has a deadline to meet [21], Also, modulation adaptation is integrated into multi-packet scheduling with deadline for each packet [21] and the Weighted Fair Queuing (WFQ) scheduling policy [22]. For a multi-hop communication path, modulation scaling is used for balancing the energy cost for all nodes along the path [23].

The real-time latency constraint considered in this paper requires the use of global time-synchronization schemes [24]. Our scenario is similar to the epoch-based data gathering scheme [25], where the length of each epoch actually plays the role of latency constraint. However, prior work has not considered the possibility of using packet-scheduling techniques that trade latency for energy in such a scenario.

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 [9] and [7], 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 [9] as well as a recursive representation of the problem for applying dynamic programming.

III. Models and Assumptions

In this section, we first describe the underlying network model, the data gathering tree. We then explain our scheme of computing data aggregation along the tree. Finally, we discuss the energy model for packet transmission and reveal the inherent energy-latency tradeoffs. For the sake of clarity, we list a summary of notations used in this paper in Table I.

TABLE I

Table of notations

$T =< V, E >$	the data gathering tree composed by the set of n sensor nodes V and the set of communication links E
$\{V_1, \ldots, V_M\}$	the set of the M leaf nodes in T
V_n	the sink node of T
T_i	the subtree rooted at V_i
Γ	the latency constraint for data gathering over T
s_i	the size of packet transmitted by V_i to its parent
τ_i	the transmission time of s_i
p_i	the path from a leaf node V_i to the sink
L_i	the length of p_i, in the metric of transmission time
C, F	the radio parameters that determine the output power and the electronic power, respectively
R	the symbol rate of the radios
$w_i(\tau_i)$	the energy function for packet transmission by V_i, with C, F, and R as parameters
m_i	the value of τ_i over $(0, \Gamma]$ that minimizes $w_i(\tau_i)$
$\vec{\tau}$	a schedule of packet transmission, $\vec{\tau} = \{\tau_1, \tau_2, \ldots, \tau_{n-1}\}$
D	the approximation accuracy of DP-Algo
x_i	the latency laxity of V_i
ρ, c	the connectivity and correlation parameters used by our simulation, respectively
N	the number of source nodes used by the random source model
S	the sensing range used by the event radius model

A. Data Gathering Tree

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 [5], [6]. 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.

Let $T = (V, E)$ denote the data gathering tree, where V denotes the set of n sensor nodes, $\{V_i : i = 1, \ldots, n\}$, 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 $V_1, \ldots, V_M$ denoting the M leaf nodes and V_n denoting the sink node. Every link in E is represented as a (source, destination) pair (i, j), implying that V_j is the parent of V_i.

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.

Raw data is generated by a set of source nodes from V (not necessarily leaf nodes). Data aggregation is performed by all non-sink and non-leaf nodes (referred to as *internal nodes* hereafter). We assume that aggregation at an internal node is performed only after all input information is available at the node – either received from its children, or generated by local sensing if the node is a source node. The aggregated data is then transmitted to the parent node. Let s_i denote the size of the packet transmitted by V_i to its parent. We discuss the computation of data aggregation to determine s_i in the next section.

For ease of analysis, it is assumed that raw data is available at all source nodes at time 0. Let Γ denote the latency constraint, within which data from all source nodes needs to be aggregated and transmitted to the sink node.

We assume that sensor nodes are completely shut down when there is no packet to transmit or receive[1]. Also, the time and energy costs for generating raw data at source nodes or aggregating data at internal nodes are considered to be negligible.

[1] Mechanisms such as signaling channel [3] can be used for synchronization between sensor nodes before any packet transmission. However, the modeling of energy cost of such mechanisms is beyond the scope of this paper

B. Data Aggregation Paradigm

While there may be transients during the creation phase of a data gathering tree, 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. Various techniques have been previously proposed for computing aggregates, or joint information entropy, from multiple source nodes. In our study, we adopt the model proposed by Pattern *et. al.* [14] where the joint entropy (or total compressed information) from multiple information sources is modeled as a function of the inter-source distance d, and a pre-specified correlation parameter c, that characterizes the extent of spatial correlation between data. Specifically, let H_1 denote the data size generated from any single source. The compressed information of two sources is calculated as [14]:

$$H_2 = H_1 + \frac{d}{d+c} H_1 \tag{1}$$

We assume that the correlation parameter c is the same for any set of sources. Based on (1), a recursive calculation of the total compressed information of multiple sources can be developed [14]. Due to space limitation, we omit the details here.

Although we use the expression in (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.

C. Energy-Latency Tradeoffs in Wireless Communication

We model the transmission energy using the example of modulation scaling [8] based on the Quadrature Ampitude Modulation (QAM) scheme [17]. 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 [26]. 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 [8]:

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

where b is the modulation level of the sender in terms of the constellation size (number of bits per symbol).

The corresponding transmission energy can be modeled as the sum of output energy and electronics energy. To illustrate the key energy-latency tradeoffs, we abstract the energy cost as a function of τ [8], denoted as $w(\tau)$:

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

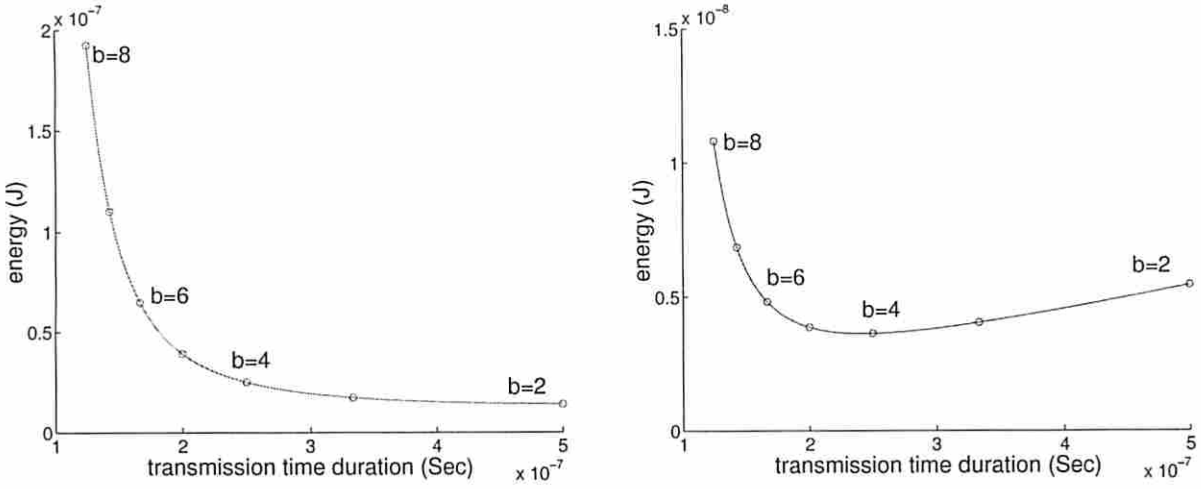
where C is determined by the quality of transmission (in terms of Bit Error Rate) and the noise power, and F is a device-dependent parameter that determines the power consumption of the electronic circuitry of the sender. Further, the output power, P_o, and the electronic power, P_e, can be modeled as follows [8]:

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

$$P_e = F \cdot R \,. \tag{5}$$

Note that different assumptions about the radio characteristics, including power consumption and data rate, may significantly affect the analysis of various energy-saving mechanisms. In this work, we consider the radio modules described in [10], [27]. Typically, for *short-range* communication with $R = 1$ Mbaud, the electronic power of the radio is approximately 10 mW, while the output power is approximately 1 mW at 4-QAM (which is translated into 2 Mbps). Note that the above data rate and power consumption are better than currently available radios for commercial sensor nodes that typically support data rate up to 100 Kbps with slightly higher power characteristics, such as Berkeley motes. However, radio devices with the above specifications are anticipated in the near future.

From the calculation of P_o and P_e, it can be derived that $C \approx 3 \times 10^{-10}$ and $F = 10^{-8}$. Further, we consider a d^2 path loss model for signal propagation, where d is the communication radius. Assuming that it takes 10 pJ/bit/m^2 by the amplifier to transmit one bit at an acceptable quality [28], we infer that the corresponding communication radius for 1 mW output power is $\sqrt{50} \approx 7$ m (from $\frac{1 \text{ mW}}{2 \times 10^6 \text{ bit/sec}} = 10$ pJ/bit/m$^2 \times d^2$). In our study, we also consider one more case of communication in WSNs with longer radius. Specifically, we set the communication range to 30 m, implying an output power of 10 pJ/bit/m$^2 \times 30^2$ m$^2 \times 2 \times 10^6$ bit/sec $= 18$ mW at 4-QAM, and consequently $C = 6 \times 10^{-9}$. We refer to this communication scenario as *long-range* communication. Note that these numbers for communication radii are for illustrative purpose



(a) Long-range communication

(b) Short-range communication

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

only – to show the different weights of C against F with respect to variations in communication radius. They may vary according to different radio devices and operating environments.

Figure 1 plots the energy functions with $b \in [2, 8]$ for the long and short range communication scenarios. In practice, b is typically set to positive even integers, as indicated by circles in the figure. We can observe a 10-time energy reduction for long-range communication by varying b from 8 to 2 and a 3-time energy reduction for short-range communication by varying b from 8 to 4. Intuitively, it is more beneficial to explore the energy-latency tradeoffs for the long-range communication. However, we demonstrate in Section VII that more than 50% energy savings can still be achieved by our algorithms for the short-range communication.

Though QAM is used as an example for abstracting the energy model, the algorithms presented in this paper are extendible to other modulation schemes and techniques that can be used to trade latency against energy, such as code scaling [26]. The applicable energy functions are assumed to satisfy the following conditions:

1) $w(\tau) > 0$;
2) there exists m such that $w(\tau)$ is monotonically decreasing in interval $(0, m]$ and monotonically increasing in interval $[m, \infty)$;
3) $w(\tau)$ is strictly convex;

4) $w(\tau)$ is continuously differentiable and its derivative $\dot{w}(\tau)$ tends to $-\infty$ when τ tends to 0.

Note that condition 3) prohibits the existence of multiple m's for any given energy function. Apparently, we only need to consider the energy-latency tradeoffs between $(0, m]$.

IV. Problem Definition

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 $L_i \leq \Gamma$.

While 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 to minimize the overall energy cost for packet transmission of all the sensor nodes in the data gathering tree.

Let $w_i(\tau_i)$ denote the energy function of V_i, with m_i denoting the value of $\tau_i \in (0, \Gamma]$ when $w_i(\cdot)$ is minimized. Note that $w_i(\cdot)$ may vary for different nodes due to variations in packet size and transmission radius (in other words, such information is implicitly embedded into $w_i(\cdot)$). We now formally state the *packet transmission problem* (PTP) as follows:

Given:

a. a data gathering tree T consisting of n sensor nodes,

b. energy functions for each link $(i, j) \in E$, $w_i(\tau_i)$, and

c. the latency constraint, Γ;

find *a schedule of packet transmission, $\vec{\tau} = \{\tau_i : i = 1, \ldots, n-1\}$, so as to minimize*

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

subject to

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

The above formulation differs from the problem defined in [9] 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 III-C indicates the presence of 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 $i = 1, \ldots, n-1$. The consequences of such differences are discussed in Section V-A.

We consider two versions of PTP, an off-line version and an on-line version. In the off-line version, the structure of the data gathering tree and the energy functions for all sensor nodes are known a priori. Centralized algorithms can be developed for solving the off-line version. In the on-line version, each sensor node only has local knowledge about its own radio status and can communication with its parent and children. Hence, distributed on-line protocols are needed to locally adapt the transmission time of each sensor node to achieve global energy minimization.

As for now, we assume a simplified communication model with a medium access control (MAC) layer that ensures no collision or interference at a node. Such an assumption can be realized by multi-packet reception (MPR) techniques through frequency or code diversity [29]. We further investigate the case when such interference must be explicitly handled by the packet scheduling algorithms in Section V-C.

V. 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 [9] 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

Since we must have $\tau_i \leq m_i$ in an optimal solution to OPTP, 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

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

The proof of the lemma is presented in Appendix I.

The problem proposed in [9] is to schedule multiple packet transmission over a single transmitter multiple receiver connection, where the ready time of packets can differ from each other. A special case of the problem is to assume all packets are of equal size and ready at time 0. This special case can also be regarded as a special case of the proposed OPTP problem where (1) the aggregation tree degenerates into a pipeline of sensor nodes – the latency constraint is imposed over exactly one path; and (2) all energy functions are monotonically decreasing. The MoveRight algorithm proposed in [9] can be directly applied to solve such a special case.

```
Begin
Set $k \leftarrow 0$                                                  // initialize iteration counter
For $(i,n) \in E$, set $\tau_i^k \leftarrow \min\{\Gamma, m_i\}$      // initialize transmission time for links to the sink
For $(i,j) \in E$ such that $j \neq n$, set $\tau_i^k \leftarrow 0$   // initialize transmission time for other links
Set $flag \leftarrow 0$                                               // flag to keep track of convergence in the iterations
While $flag = 0$
    $k \leftarrow k + 1$                                              // increment the iteration counter by 1
    For each $V_i$ with $i$ from $n-1$ downto M+1                     // perform local optimization for each internal node
        $(\{\tau_j^k\}_{(j,i)\in E}, \tau_i^k) \leftarrow$ best$(\{\tau_j^{k-1}\}, \tau_i^{k-1})$   // move right the start time of transmission from $V_i$
    For $(i,n) \in E$
        Set $\tau_i^k \leftarrow \min\{m_i, \Gamma - (\max_{V_i \in p_j}\{L_j\} - \tau_i^k)\}$   //increase the transmission time for links to the sink
    if $\vec{\tau}^k = \vec{\tau}^{k-1}$, $flag \leftarrow 1$          // check convergence
End
```

Fig. 2. Pseudo code for EMR-Algo

In this section, we are interested in extending the MoveRight algorithm 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, τ_i^k denotes the value of τ_i in 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 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 proposed EMR-Algo is distinguished from the MoveRight algorithm in two key respects (recall the difference discussed in Section IV between our problem and the one defined in [9]). (1) The best($\cdot$) function respects Lemma 1 regarding the optimality 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 $\vec{\tau}^* = \{\tau_1^*, \ldots, \tau_{n-1}^*\}$ be the optimal schedule. Let $s_i^* = 0$, for $i = 1, \ldots, M$; and $s_i^* = \max_{(j,i) \in E}(s_j^* + \tau_j^*)$, for $i = M, \ldots, n-1$. As previously stated, $\{\tau_i^k : k = 1, \ldots, n-1\}$ indicate the transmission time of nodes $V_1, \ldots, V_{n-1}$ after the k-th pass of EMR-Algo. Let $s_i^k = 0$, for $i = 1, \ldots, M$, and $s_i^k = \max_{(j,i) \in E}(s_j^k + \tau_j^k)$, for $i = M, \ldots, n-1$.

Theorem 1: Let s_i^k and s_i^*, $i = 1, \ldots, n-1$ be as defined before. 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 similar to that of Theorem 1 in [9] and is detailed in Appendix I.

The convergence speed of EMR-Algo depends on the structure of the aggregation tree and the exact form of the energy functions. It is therefore difficult to give a theoretical bound on the number of iterations. In Section VII, we show running time of EMR-Algo for simulated problems. 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 V-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 $g(V_i, t)$:

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

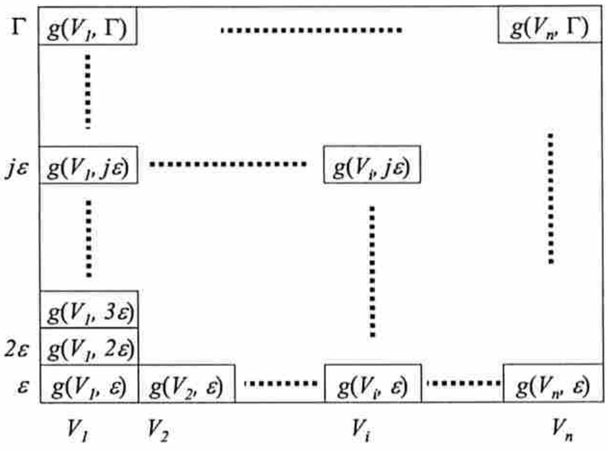


Fig. 3. The $g(\cdot)$ table computed by DP-Algo

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 $g(V_i, j\varepsilon)$.

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*. Such a rounding procedure may affect the performance of DP-Algo. As shown in Section VII, the performance degradation is around 10% for loose latency constraints and 50% for stringent latency constraints.

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 the corresponding cells in the table with infinity.

C. Multi-Packet Reception Techniques for Interference Avoidance

The definition of PTP assumes that there is no interference among the sensor nodes. Such an assumption can be realized by using MAC layer scheduling that minimizes interference using, for example, CTS/RTS mechanisms. To use such mechanisms, we need to intentionally set the latency constraint imposed on PTP to be less than the actual constraint. The preserved time slot can be used to accommodate the back-off time of the sensor nodes when collision occurs.

A more efficient technique is to use multi-packet reception (MPR) through frequency or code division multiple access (i.e., FDMA and CDMA) [29]. The challenges to apply the MPR technique is to satisfy the so-called "scheduling constraint", which enforces that at any time, any sensor node can either transmit (broadcast) to a set of nodes or receive from at most one node. For the data gathering tree topology, since every sensor node transmits its packet only after receiving packets from all of its children, the scheduling constraint essentially requires that at any time, at most one child of any sensor node can transmit packets to the parent. In other words, the packet transmission time from children of the same sensor node should not overlap with each other. We refer to such an approach as *interference avoidance* (IA).

To achieve interference avoidance in the context of PTP 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 stricter 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 in 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 $g(V_i, t)$:

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

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 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 due to the non-monotonic energy functions. 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 above modified DP-Algo as the DP-IA algorithm. It can be verified that the time complexity of DP-IA is also $O(D^2(|V|+|E|))$. We further examine the performance degradation resulting from interference avoidance in Section VII.

VI. Distributed On-line Protocol

The algorithms presented in Section V all assume a complete knowledge of the data gathering tree. However, the discrete approximation of the energy function motivates a simple on-line distributed protocol that relies on local information of sensor nodes only. The key idea of the protocol is to identify the sensor nodes with the largest energy cost on each path of the tree. We then increase their transmission time if the latency constraint will not be violated. We repeat the above procedure until either the latency constraint is reached for all paths or the energy cost of the gathering tree is minimized.

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 node V_i can derive the time cost for data gathering within the subtree rooted V_i.
3) Every sensor node can measure its current power consumption, and hence its energy gradient – the energy gain by increasing the transmission time of the node by ε.
4) Interference among sensor nodes is minimized by using MAC layer scheduling.

The local identifier in assumption 1 is commonly implemented in protocols such as Directed Diffusion [5]. 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 [24], 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 can be satisfied by intentionally setting the latency constraint to be tighter than the actual constraint for accommodating the incurred time cost for resolving collisions.

Moreover, we define the *latency laxity* of a node as the maximal amount of time that can be used to increase the transmission time of the node without violating the latency constraint. Let x_i denote the latency laxity of V_i. The latency laxity of each node is dynamically maintained during the protocol to verify if the transmission time of the node can be safely increased.

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

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 of the longest path, *excluding* τ_i, in T_i.

The local data structure is maintained as follows. Every leaf node piggybacks 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 piggybacked 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 piggybacked 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 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. Basically, δ is the latency laxity of nodes on the longest path of the data gathering tree.

Once a sensor node V_i receives the feedback packet from its parent, it performs the following adaptation. 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. First, the latency laxity of V_i can be calculated as $x_i = \delta + \tau'_d - (\tau_i + \tau_d)$. This is because $\tau_i + \tau_d$ is the time cost of T_i; τ'_d is the time cost of the longest path in the subtree rooted at V_i's parent (excluding the transmission time of V_i's parent); and δ is actually the latency laxity of nodes on this longest path. Then, V_i takes one of the following actions.

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 $x_i \geq \varepsilon$, the transmission time of V_i is increased by ε. 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 $\delta = x_i$ 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 with an increased rate (action 1). If V_i is the node with the largest positive energy gradient in T_i and the latency laxity allows, the transmission time of V_i is increased (action 2). Otherwise, the latency laxity of V_i is recorded in the feedback packet and the sensor nodes in T_i are recursively examined (action 3).

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. Therefore, the on-line protocol converges after the latency constraint is reached by all paths, or $\tau_i = m_i$, for each $V_i \in V$. We assume that each sensor node has q discretized transmission time. Before the protocol converges, a feedback packet would increase the transmission time for at least one sensor node when it traverses the data gathering tree. Thus, the protocol converges after the dissemination of at most nq feedback packets, where n is the number of sensor nodes in the 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 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 cost. 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 cost 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.

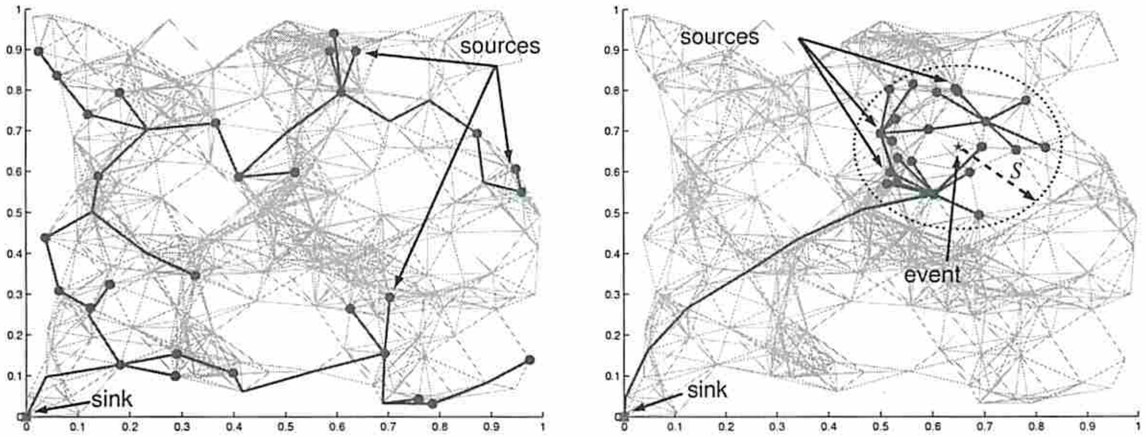
Another option to handle latency violations is to repeatedly reduce the transmission time of the sensor nodes with the smallest energy gradient till the latency constraint is satisfied. Compared with the proposed technique that simultaneously reduces the transmission time of all sensor nodes, such an option is more aggressive in the sense of reducing the incurred increment in energy cost. However, it requires more sophisticate control protocol and more importantly, increases the response time in handling latency violations.

VII. Simulation Results

To conduct the simulations, a simulator was developed using the PARSEC [30] software, which is a discrete-event simulation language. The purposes of the simulations are (1) to demonstrate the energy gain achieved by our algorithms 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

The transmission speed of sensor nodes is continuously tunable by setting modulation level within $[2, 8]$, except for the special case of modulation scaling where modulation level can only be integer even values within $[2, 8]$. Hence, for all sensor nodes, the highest data rate is 8 Mbps and the lowest data rate is 2 Mbps. The baseline in our simulations is to transmit all packets at the highest speed (i.e., 8 Mbps), and shutdown the radios afterward. This policy is use, for example, in the PAMAS protocol [31] and the DMAC protocol [32].



(a) Random sources model (number of sources $N = 30$)

(b) Event radius model (sensing range $S = 0.2$)

Fig. 4. Two example data gathering trees generated by the random sources and event radius models, respectively (connectivity parameter $\rho = 0.15$)

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 neighbors that a sensor node can directly communicate is determined by a connectivity parameter, $\rho \in (0, 1]$. Specifically, two sensor nodes can communicate with each other only if the distance between them is within ρ. We used two models for generating the location of the data sources, namely the random sources (RS) model and the event radius (ER) model. In the RS model, N (the number of sources) out of 200 sensor nodes are randomly selected to be the sources, whereas in the ER model, all sources are located within a distance S (essentially the sensing range) of a randomly chosen "event" location. For both models, the Greedy Incremental Tree (GIT) algorithm [6] was used for constructing the data gathering tree. In Figure 4, we illustrate two example data gathering trees generated based on the RS and ER models, respectively.

The energy function used in the simulation was in the form of (3). Unless otherwise stated, we set $R = 10^6$ and $F = 10^{-8}$ for all the sensor nodes, while the value of C of a sensor node was determined by the distance from the node to its parent in the tree. Specifically, we assume a d^2 power loss model, where d is the distance between a node and its parent. Then, for node V_i, we have $C_i = C_{base} \cdot (\frac{d}{\rho})^2$. Based on our analysis in Section III-C, C_{base} was set to 6×10^{-9} for the long-range communication and 3×10^{-10} for the short-range communication.

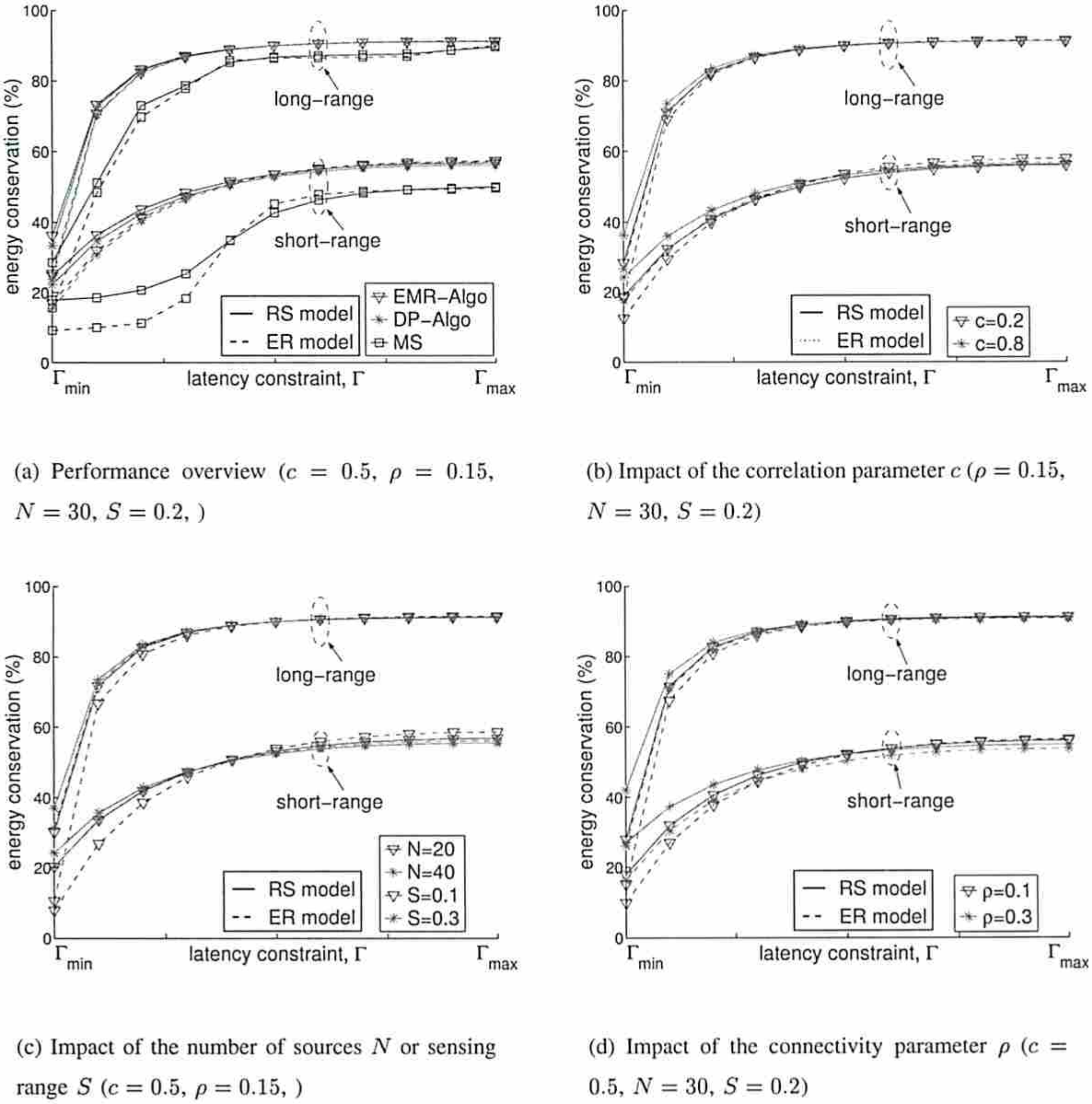
During our simulation, the latency constraints Γ was determined as follows. We define the shortest time cost, Γ_{min} of a gathering tree as the transmission latency of the longest path in the tree when all sensor nodes transmit at the highest speed (8 Mbps). On the other hand, the longest time cost, Γ_{max} of the gathering tree is defined as the transmission latency of the longest path in the tree when every sensor node V_i sends its packet using time $\min\{m_i, \frac{s_i}{2}\}$. In the above definition, the term m_i comes from the fact that it is not energy beneficial for V_i to transmit its packet using time beyond m_i; the term $\frac{s_i}{2}$ is due to the lower bound of modulation level in our simulation. Therefore, Γ was adjusted between Γ_{min} and Γ_{max}.

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 approximation accuracy for DP-Algo, D was set to 100. The size of raw data generated by source nodes was set to 200 bits.

In Figure 5, we show the energy saving achieved by our off-line algorithms for both the RS and ER models. For both models, we show the results for long and short range communication. Each data point in the figures is averaged over more than 100 instances such that it has a 95% confidence interval with a 5% (or better) precision. In the following, we focus on analyzing the results for the RS model; similar analysis can be made for results of the ER model.

Performance Overview: In Figure 5(a), we investigate the performance of algorithms including EMR-Algo, DP-Algo and the special case of DP-Algo for modulation scaling (denoted as MS in the figure) when varying Γ from Γ_{min} to Γ_{max}. The first thing to notice is that when Γ approaches Γ_{max}, our algorithms achieve more than 90% energy saving for the long-range communication, and around 50% for the short-range communication. Even when $\Gamma = \Gamma_{min}$, EMR-Algo and



(a) Performance overview ($c = 0.5$, $\rho = 0.15$, $N = 30$, $S = 0.2$,)

(b) Impact of the correlation parameter c ($\rho = 0.15$, $N = 30$, $S = 0.2$)

(c) Impact of the number of sources N or sensing range S ($c = 0.5$, $\rho = 0.15$,)

(d) Impact of the connectivity parameter ρ ($c = 0.5$, $N = 30$, $S = 0.2$)

Fig. 5. Performance of our off-line algorithms (c: correlation parameter, ρ: connectivity parameter, N: number of sources, S: sensing range)

DP-Algo can still save more than 30% of the energy for long-range communication and 20% for short-range communication.

The reason for successful energy saving even when $\Gamma = \Gamma_{min}$ is because Γ equals the transmission time of the longest path in the data gathering tree. Thus, energy can still be reduced for nodes not on the longest path. On one hand, when there exists only one path in the tree, no energy can be saved when $\Gamma = \Gamma_{min}$. On the other hand, when the tree forms a star-like

TABLE II

THE MISS RATE OF MS (BASED ON SIMULATED INSTANCES FOR FIGURE 5(A))

source model	communication scenario	# of successful instances	# of failed instances	total number of instances	miss rate (%)
RS	long-distance	102	28	130	21
	short-distance	104	16	120	13
ER	long-distance	324	66	390	17
	short-distance	352	38	390	10

structure, all links, except the longest ones, can be optimized for energy savings when $\Gamma = \Gamma_{min}$. This also explains the performance degradation of our algorithms in the ER model, compared with the performance in the RS model. Specifically, as illustrated by Figure 4, the gathering tree for the ER model forms a small cluster connected to the sink by a linear array of sensor nodes, while the tree for the RS model is more close to a star-like structure.

The plot shows that the performance of DP-Algo is quite close to the performance of EMR-Algo. However, the performance of MS quickly degrades when Γ decreases. From Figure 1, the first derivative of the energy function decreases fast as τ tends to 0. Thus, when Γ is small, the rounding procedure for solutions with high modulation levels leads to large performance loss.

During our simulation, we also observed that when $\Gamma = \Gamma_{min}$, MS fails to find feasible solutions for some problem instances, due to the rounding procedure of the approximated solutions from DP-Algo. The ratio of instances that MS fails over the total number of instances is between 10 - 20% in our simulations. We define the miss rate of MS as the ratio of number of instances that MS fails to find a feasible solution over the number of total problem instances. The miss rate for the performed simulations is given in Table II.

The simulation was performed on a SUN Blade1000 with a 750 MHz SUN UltraSPARC III processor. The running time of EMR-Algo is between 0.5 to 3 second, whereas the running time of DP-Algo is around 0.01 second.

Impact of Network Parameters: Figure 5(b) shows the energy conservation achieved by DP-Algo with respect to variations in c and Γ. It was observed that for a fixed Γ, the energy gain of DP-Algo slightly increases when c increases. This is because a smaller value of c causes a larger size of data packet after aggregation. Thus, the energy cost by links close to the sink node dominates the overall energy cost of the tree. It is however difficult to reduce the energy cost of these links since they have high a likelihood of lying on the longest path of the tree.

Figure 5(c) plots the performance of DP-Algo with respect to variations in N and Γ. It can be seen that when Γ is close to Γ_{min}, the energy gain of DP-Algo increases as the number of sources increases. This is because a larger number of sources offers more opportunities for the optimization of links on paths other than the longest one.

Figure 5(d) demonstrates the performance of DP-Algo with respect to variations in ρ and Γ. It can be observed that the energy saving of DP-Algo increases when ρ increase. This is understandable since a large ρ reduces the height of the data gathering tree (the extreme case is a star-like tree formed by setting $\rho = 1$).

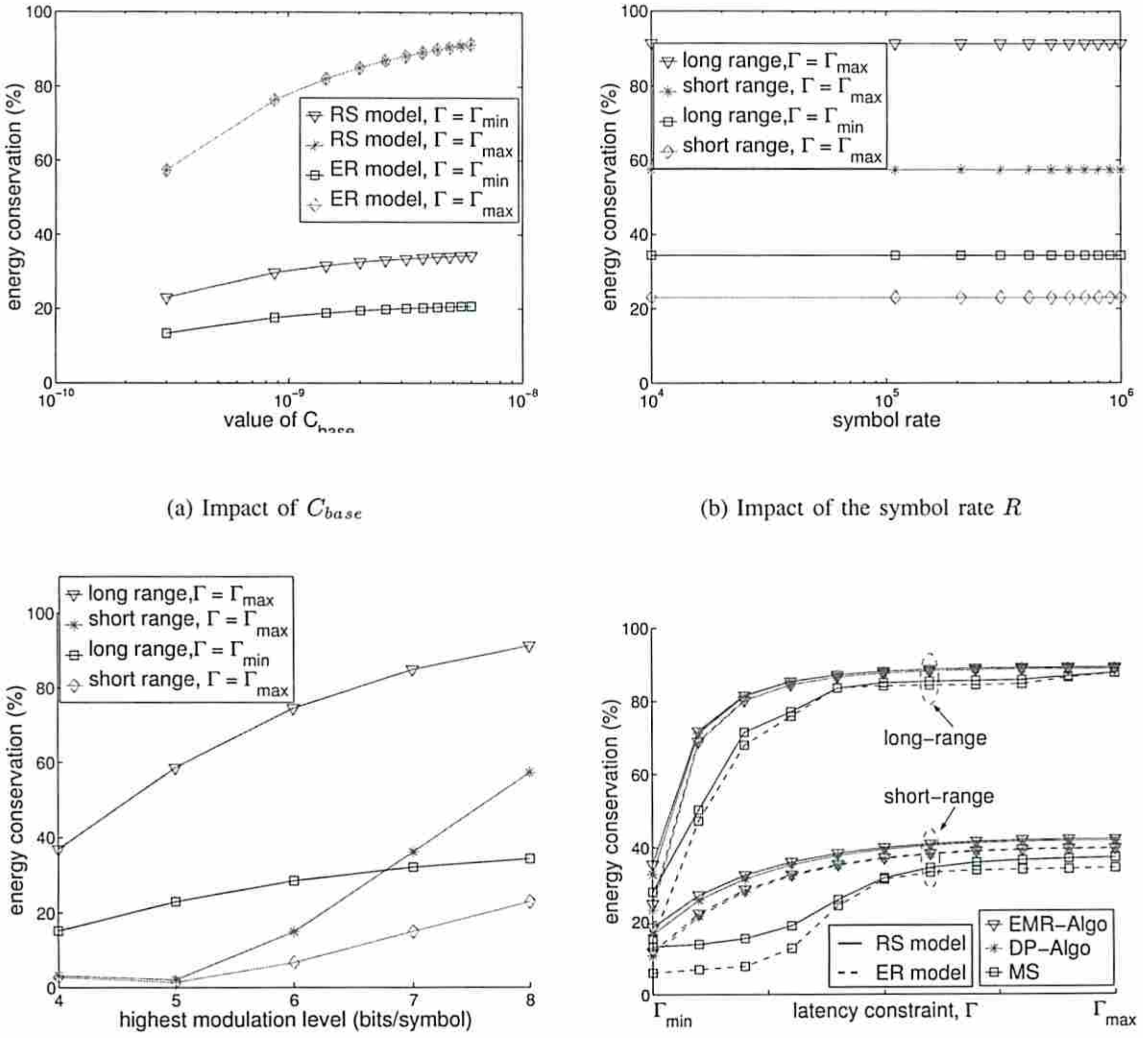
Together, the above results suggest that DP-Algo is quite robust with respect to variations in different system parameters, including c, ρ, N, and S.

Impact of the radio parameters: In Figure 6(a), we show the impact of radio parameter C_{base} on the performance of DP-Algo. In the figure, the x-axis represents the value of C_{base} from 3×10^{-10} to 6×10^{-9} in logarithmic scale. As expected, the energy conservation achieved by DP-Algo increases with C. Also, when $\Gamma = \Gamma_{max}$, there is almost no difference in the performance of DP-Algo under either RS or ER models; whereas when $\Gamma = \Gamma_{min}$, a performance degradation of 9-14% is observed for the ER model compared with the RS model.

To evaluate the impact of symbol rate R, we varied R from 10 KBaud to 1 MBaud. Consider the modulation level within $[2, 8]$, the above range of R reflects a bit rate of 20 to 80 Kbps when $R = 10$ KBaud and 2 to 8 Mbps when $R = 1$ MBaud. We show the performance of DP-Algo for RS model in Figure 6(b). It can be seen that the energy saving is almost the same throughout the variation of R. This is quite understandable, since from (3), the performance ratio of DP-Algo to the baseline is determined by b, but not R.

We further investigate the energy saving of DP-Algo under different settings of the highest modulation level of the radio. In Figure 6(c), we show the energy saving achieved by DP-Algo in RS model when the highest modulation level is varied from 4 to 8. As expected, lower highest modulation level results in less energy saving. When the modulation level is restricted within [2, 4], less than 20% energy saving is achieved for both long and short range communication.

We now show the impact of start-up energy of radios, which was estimated as 1 μJ [10]. In each epoch, the radio of each sensor node is started exactly once. Figure 6(d) shows the



(a) Impact of C_{base}

(b) Impact of the symbol rate R

(c) Impact of the highest modulation level

(d) Impact of the start-up energy (1 μJ)

Fig. 6. Impact of radio parameters (correlation parameter $c = 0.5$, connectivity parameter $\rho = 0.15$, number of sources $N = 30$, sensing range $S = 0.2$)

performance of our algorithms with start-up energy. Though the impact of the start-up energy to the long-range communication is almost negligible, we observe a decrease of 6-15% in energy conservation for the short-range communication (compared with Figure 5(a)). This is because the start-up energy is comparable to the transmission energy for short-range communication.

Impact of Interference Avoidance: We also examined the performance of DP-IA with respect to variations in Γ (shown in Figure 7). It can be seen that the performance degradation due to interference avoidance is severe when Γ is small. When $\Gamma = \Gamma_{min}$, DP-IA achieves around half of the energy conservation by DP-Algo for both long and short range communication. However, when Γ is large, DP-Algo achieves almost the same energy conservation with or without interference avoidance.

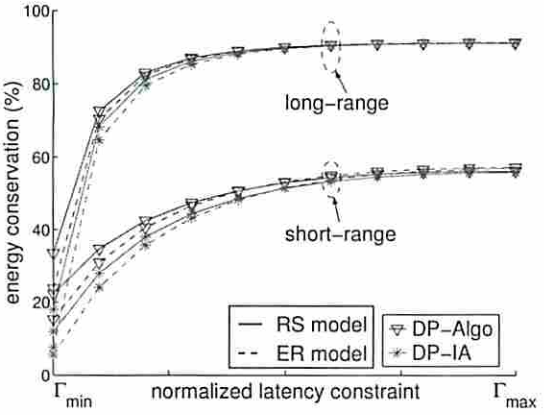


Fig. 7. Impact of interference avoidance (correlation parameter $c = 0.5$, connectivity parameter $\rho = 0.15$, number of sources $N = 30$, sensing range $S = 0.2$)

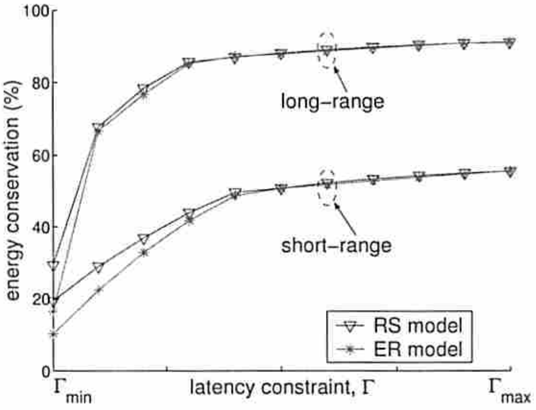
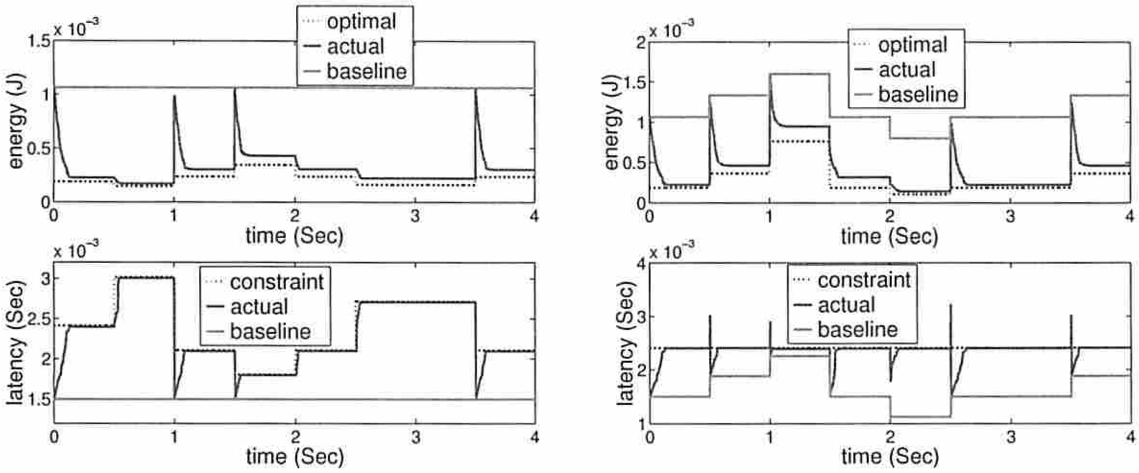


Fig. 8. Performance of the on-line protocol (correlation parameter $c = 0.5$, connectivity parameter $\rho = 0.15$, number of sources $N = 30$, sensing range $S = 0.2$)

C. Performance of the On-Line Protocol

Energy Conservation: We show the energy conservation achieved by the on-line protocol in Figure 8. The simulated on-online protocol is based on modulation scaling where available modulation levels are even integers within $[2, 8]$. The presented data is averaged over more than 150 problem instances and has a 95% confidence interval with a 10% (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 gathering tree was then generated using GIT.

When the latency constraint approaches Γ_{min}, there is slight performance degradation compared with DP-Algo from Figure 5(a). Specifically, for the RS model, we observe around 4% less energy conservation for long-range communication and 3% for short-range communication. This is reasonable considering 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 DP-Algo. Moreover, the on-line protocol actually outperforms the modulation scaling case (MS) shown in Figure 5(a), implying a large performance degradation of the rounding technique used by MS.



(a) Scenario A

(b) Scenario B

Fig. 9. Adaptability of the on-line protocol (correlation parameter $c = 0.5$, connectivity parameter $\rho = 0.15$)

Adaptability to System Variations: Our simulations were performed based on the tree shown in Figures 4(a) that has 44 sensor nodes out of which 30 are source nodes. Again, we assume that modulation scaling is used by all the nodes with the available modulation levels being even numbers within $[2, 8]$. The data gathering was requested every 2 milliseconds. For the sake of illustration, we set $\alpha = 4$ milliseconds, and $\beta = 10$. Two run-time scenarios, namely A and B, were investigated to demonstrate the efficiency and adaptability of our protocol.

Scenario A: We fixed s at 200 bits. It can be calculated that $\Gamma_{min} \approx 1.5$ milliseconds and $\Gamma_{max} \approx 4.5$ milliseconds. We set Γ to 2.4, 3, 2.1, 1.8, 2.1, 2.7, 2.1 milliseconds at 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, change of user requests.

We depict the energy cost and latency for data gathering over 4 seconds in Figure 9(a), where the optimal solutions are obtained by using EMR-Algo. It can be observed that when Γ 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, Γ is varied from 3 milliseconds to 2.1 milliseconds, 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. Moreover, 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.

Scenario B: We set $\Gamma = 2.1$ milliseconds, while setting s to 200, 250, 300, 200, 150, 200, and 250 at 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 correlation parameter at sensor nodes. The results is illustrated in Figure 9(b), where the optimal solutions are also obtained by using EMR-Algo. 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.

VIII. Concluding Remarks

In this paper, we have 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 have 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. Our simulation results show that between 20% up 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. Furthermore, we have proposed a distributed on-line protocol that relies on local information only. Our simulation results show that the energy saving achieved by the protocol is between 15% up to 90%. Also, the ability of the protocol 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 [5] 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] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *IEEE InfoCom*, Mar. 2004.

[2] S. Conner, L. Krishnamurthy, and R. Want, "Making everyday life easier using dense sensor networks," in *ACM UbiComp*, 2001.

[3] 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.

[4] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Processing Magazine*, Mar. 2002.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, March 2002.

[11] C. Z. Zhou and B. Krishnamachari, "Localized topology generation mechanisms for self-configuring sensor networks," 2003, iEEE GlobeCom.

[12] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk," in *ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[13] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *IEEE InfoCom*, 2004.

[14] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *ACM/IEEE International Symposium on Information Processing in Sensor Networks*, 2004.

[15] M. H. Ahmed, H. Yanikomeroglu, D. D. Falconer, and S. Mahmoud, "Performance enhancement of joint adaptive modulation, coding and power control using cochannel-interferer assistance and channel reallocation," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2003.

[16] K. Balachandran, S. R. Kadaba, and S. Nanda, "Channel quality estimation and rate adaption for cellular mobile radio," *IEEE J. of Selected Areas in Communication (JSAC)*, vol. 17, pp. 1244–1256, 1999.

[17] 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.

[18] O. B. Akan and I. F. Akyildiz, "ARC: The analytical rate control scheme for real-time traffic in wireless networks," *IEEE/ACM Trans. on Networking*, June 2004, to appear.

[19] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2001.

[20] W. H. Yuen, H.-N. Lee, and T. D. Andersen, "A simple and effective cross layer networking system for mobile ad hoc networks," in *PIMRC*, 2002.

[21] C. Schurgers, V. Raghunathan, and M. B. Srivastava, "Modulation scaling for real-time energy aware packet scheduling," in *IEEE GlobeCom*, Nov. 2001.

[22] V. Raghunathan, S. Ganeriwal, 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.

[23] Y. Yu and V. K. Prasanna, "Energy-balanced multi-hop packet transmission in wireless sensor networks," in *IEEE GlobeCom*, Dec. 2003.

[24] 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.

[25] 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.

[26] E. Armanious, 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.

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

[28] W. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application specific protocol architecture for wireless microsensor networks," *IEEE Trans. on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[29] 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.

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

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

[32] B. K. G. Lu and C. Raghavendra, "An adaptive energy-efficient and low-latency mac for data gathering in sensor networks," in *4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN 04)*, Apr. 2004.

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

APPENDIX I
CORRECTNESS OF EMR-ALGO

Proof of Lemma 1: $\Rightarrow$ Condition (1) trivially holds for an optimal solution. Otherwise, we can always increase τ_i^* without violating the latency constraint and thus decrease the energy dissipation of V_i.

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 [33], there exists a vector of values $\vec{\lambda} = (\lambda_1, \ldots, \lambda_M)^T$, such that

$$\nabla f(\vec{\tau^*}) + A^T \vec{\lambda} = 0 \tag{11}$$

where $\nabla f(\vec{\tau^*})$ is a column vector with the partial derivative $\frac{\partial f(\vec{\tau^*})}{\partial \tau_i^*} = \dot{w}_i(\tau_i^*)$ as the i-th element. By solving equation 11, we have for any internal node, V_i,

$$\dot{w}_i(\tau_i^*) + \sum_{j:V_i \in p_j} \lambda_j = 0 \tag{12}$$

The sum in equation 12 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

$$\sum_{j:V_i \in p_j} \lambda_j = \sum_{(k,i) \in E} \{ \sum_{j:V_k \in p_j} \lambda_j \} \tag{13}$$

Thus, the necessary condition of optimality can be obtained by summing up equations 12 for all the children of V_i and subtracting equation 12 for V_i.

⇐ The proof follows the fact that the feasible space of OPTP is convex and compact. ■

Comments: We examine Lemma 1 in two interesting examples. In the first example, consider an aggregation tree $T = (V, E)$ where $V = \{V_1, V_2, V_3, V_4\}$ and $E = \{(1,3), (2,3), (3,4)\}$. Let $\vec{\tau^*}$ denote the optimal schedule of such a problem. Assume that $m_2 + m_3 \geq \Gamma$ and m_1 is fairly small compared with m_2 and m_3 such that we have $\tau_1^* = m_1$ in the optimal schedule. In this case, we shall have $\dot{w}_1(\tau_1^*) = 0$ and $\dot{w}_2(\tau_2^*) = \dot{w}_3(\tau_3^*)$, which respects Lemma 1.

In the second example, consider an OPTP problem with an extremely large latency constraint (e.g., $\Gamma = \infty$) such that the optimal schedule is obtained by setting $\tau_i^* = m_i$ for each $V_i \in V$. In such a case, we have $\dot{w}_i(\tau_i^*) = 0$, for each $V_i \in V$, which again respects Lemma 1.

More importantly, we shall note the following fact in the optimal solution for the second example. Let s_i denote the start time for packet transmission from V_i. Ideally, s_i can be determined as $\max_{(j,i) \in E}(s_j + \tau^* j)$. However, due to the laxity in the latency constraint, we can safely increase the start time of packet transmissions to some extent without compromising the optimality of the schedule. To prevent such situations, we adopt a strict definition of s_i in the proof of Lemma 2 and Theorem 1 such that s_i must equal $\max_{(j,i) \in E}(s_j + \tau^* j)$.

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

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

Proof of Corollary 1:

1) Since $\tau_i^* = m_i$ implies $\sum_{(j,i)\in E} \dot{w}_j(\tau_j^*) = \dot{w}_i(\tau_i^*) = 0$, we have $\tau_j^* = m_j$ for all children of V_i. So on so forth, we have $\tau_j^* = m_j$ for all sensor nodes in the subtree rooted at V_i.
2) Let V_j denote the parent of V_i. $\tau_i^* = m_i$ implies $\dot{w}_j(\tau_j^*) \leq \dot{w}_i(\tau_i^*) < 0$; hence $\tau_j^* < m_j$. So on and so forth. ■

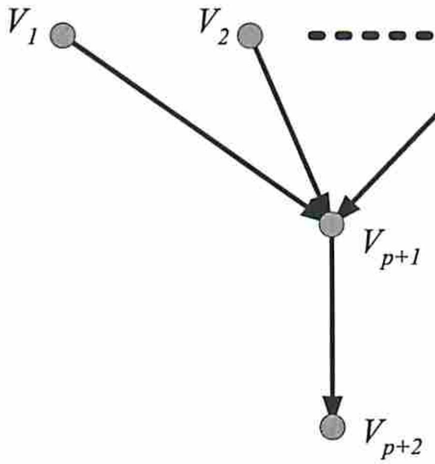


Fig. 10. A problem instance of 2-Lev-OPTP

We now 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 with exactly one internal node that has p children (see Figure 10). We call the above problem 2-Lev-OPTP. Let V_{p+1} denote the internal node, with V_{p+2} denoting its parent and $\mathcal{C} = \{V_1, \ldots, V_p\}$ denote its children. 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. We first prove the following lemma:

Lemma 2: Let $\vec{\tau^*} = \{\tau_1^*, \ldots, \tau_{p+1}^*\}$ 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 (≥ 1) 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.

Proof of Lemma 2:

1) The uniqueness of the optimal solution follows the strict convexity of the energy functions.
2) From Lemma 1, the optimal schedule $\vec{\tau^*}$ must satisfy $\sum_{i=1}^{p} \dot{w}_i(\tau_i^*) = \dot{w}_{p+1}(\tau_{p+1}^*)$. Moreover, we have $\tau_i^* \leq m_i$ for $i = 1, \ldots, p+1$. In the following, we prove property (a).

We first assume that the transmission start time of exactly one child of V_{p+1} increases. That is, for some $V_a \in \mathcal{C}$, s_a increases to s'_a. Let $\hat{s}_{p+1}$ denote the start time of packet transmission from V_{p+1} in the resulting optimal schedule. We consider the following two cases.

Case (i): Suppose that in schedule $\vec{\tau^*}$, $\tau_a^* \leq m_a$. This implies that $s_{p+1} = s_a + \tau_a^*$. Otherwise, a schedule with less energy dissipation than $\vec{\tau^*}$ can be constructed by increasing τ_a^* by $\delta \leq \min\{s_{p+1} - (s_a + \tau_a^*), m_a - \tau_a^*\}$ without affecting τ_{p+1}^* or violating the latency constraint. Similarly, we have $\tau_{p+1}^* \leq m_{p+1}$ and $s_{p+1} + \tau_{p+1}^* = t$.

Now we consider the problem resulted from increasing s_a to s'_a. Suppose that in the new packet schedule, we still enforce the transmission by V_{p+1} to start at s_{p+1}, we have:

$$\dot{w}_a(\tau_a^* - (s'_a - s_a)) + \sum_{V_i \in \mathcal{C} \wedge i \neq a} \dot{w}_i(\tau_i^*) < \sum_{V_i \in \mathcal{C}} \dot{w}_i(\tau_i^*) = \dot{w}_{p+1}(\tau_{p+1}^*) . \tag{14}$$

The above inequality comes from the fact that the first derivative of an energy function is strictly increasing due to its strict convexity.

Or, we may start the transmission by V_{p+1} at $s_{p+1} + (s'_a - s_a)$ in the new schedule. Since

$\tau^*_{p+1} \leq m_{p+1}$, we have:

$$\begin{aligned} \dot{w}_a(\tau^*_a) + \sum_{V_i \in C \wedge i \neq a} \dot{w}_i(\min\{\tau^*_i + (s'_a - s_a), m_i\} &\geq \sum_{V_i \in C} \dot{w}_i(\tau^*_i) \\ &= \dot{w}_{p+1}(\tau^*_{p+1}) \\ &> \dot{w}_{p+1}(\tau^*_{p+1} - (s'_a - s_a)) \ . \qquad (15) \end{aligned}$$

Equations 14 and 15 and the uniqueness of the optimal schedule imply that $s_{p+1} < \hat{s}_{p+1} < s_{p+1} + (s'_a - s_a)$.

Case (ii): Suppose that in schedule $\vec{\tau^*}$, $\tau^*_a = m_a$; hence, we have $s_{p+1} - s_a \geq m_a$. If $s_{p+1} - s'_a < m_a$, a similar analysis as in Case (i) can be carried out to show that $(s_{p+1} < \hat{s}_{p+1} < s_{p+1} + (s'_a - s_a)$. Otherwise, $\vec{\tau^*}$ remains an optimal schedule, implying that $\hat{s}_{p+1} = s_{p+1}$.

In case when multiple s_i's ($V_i \in C$) increase, it can handled by increasing these s_i's one after another and the lemma still holds. Property (b) can be analyzed in a similar fashion. Also, Property (c) can be handled by first increasing s_i's and then increasing t.

3) This part of the lemma is actually an inverse case of part (2) and can be easily proved by contradiction. ■

Now we present the proof of Theorem 1.

Proof of Theorem 1:

1) Recall that EMR-Algo works in iterations: for each iteration k, the algorithm determines s^k_i by decreasing i from $n-1$ to $M+1$. Since the EMR-Algo initializes $s_i = 0$ for $i = 1, \ldots, n-1$, it follows that $s^0_i \leq s^1_i$ for each $i = 1, \ldots, n-1$. Suppose that $i' > 1$ and $k' > 1$ are the first time that there is a violation; that is, $s^{k'}_{i'} > s^{k'+1}_{i'}$.

   Consider the 2-level aggregation tree formed by $V_{i'}$ together with its parent, denoted as V_p, and its children, denoted as set C. We have $s^{k'}_p \leq s^{k'+1}_p$, and $s^{k'-1}_i \leq s^{k'}_i$, for each $V_i \in C$. From line 8 in EMR-Algo, the time stamps $s^{k'}_p$ and $s^{k'-1}_i$'s actually give the boundaries within which EMR-Algo determines $s^{k'}_{i'}$. Similarly, the time stamps $s^{k'+1}_p$ and $s^{k'}_i$'s give the boundaries within which EMR-Algo determines $s^{k'+1}_{i'}$. From part (2) of Lemma 2, we have $s^{k'}_{i'} \leq s^{k'+1}_{i'}$. This contradicts the assumption $s^{k'}_{i'} > s^{k'+1}_{i'}$ and hence property (1) holds.

2) It is obvious that $s^0_i \leq s^*_i$, for each $i = 1, \ldots, n-1$. 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, $s^{k'}_{i'} > s^*_{i'}$.

Again, 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 $s_p^{k'} \le s_p^*$, and $s_i^{k'-1} \le s_i^*$, for each $V_i \in \mathcal{C}$. The time stamps $s_p^{k'}$ and $s_i^{k'-1}$'s actually give the boundaries within which EMR-Algo determines $s_{i'}^{k'}$. Similarly, the time stamps 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 $s_{i'}^{k'} \le s_{i'}^*$ 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 τ_j^* is optimal, we have $\tau_j^* \le m_j$. 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 $s_i^\infty \le s_i^*$, for each $V_i \in \mathcal{C} \wedge i \ne j$. 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 $s_p^\infty - (s_j^* - s_j^\infty) < \hat{s}_p < s_p^\infty = s_p^*$, 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 $s_p^\infty - s_j^\infty > s_p^* - s_j^* = \tau_j^* = m_j$. Obviously, we shall 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^∞ and s_j^*, we obtain the contradiction that $s_j^\infty = s_j^*$. ■