Power Management in Cluster-Based Energy-Harvesting Sensor Networks through Dynamic Modulation Scaling

Maryam Bandari, Robert Simon, and Hakan Aydin
Department of Computer Science
George Mason University, Fairfax, VA 22030
{mbandari, simon, aydin}@gmu.edu

***Abstract*—This paper considers real-time cluster-based wireless sensor networks where the nodes harvest energy from the environment. We target performance sensitive applications that have to collectively send their information to cluster head by a predefined deadline, such as in distributed real-time monitoring and detection. The nodes are equipped with Dynamic Modulation Scaling (DMS) capable wireless radios. The problem is to determine the time slots and modulation levels that will be used by each node while communicating with the cluster-head in order to achieve energy-neutral (perpetual) operation and maximize energy reserves. We propose a solution that adjusts underlying TDMA slots to enable high-energy nodes to transmit faster and thus produce larger slack for low-energy nodes, while meeting the performance constraint. We present an optimal mixed integer programming based solution. We also develop fast heuristics that are shown to provide approximate solutions through comprehensive experiments with actual solar energy harvesting profiles.**

I. Introduction

Power management in wireless sensor networks (WSNs) remains as a critical challenge, considering the long-term deployment requirements of these systems and the scarce availability of the battery power. Recently, there has been a growing interest in using *energy harvesting* solutions in WSNs. Using environmental energy harvesting technology to charge the storage units that power the wireless sensor nodes offers multiple benefits [1]. Energy harvesting reduces or eliminates the need for a direct connection to a power main or the requirement to change potentially inaccessible or expensive batteries. It is also a sustainable and environmentally friendly approach to energy production. For nodes installed in harsh and inaccessible environments, energy harvesting provides long term system life that reduces the need for maintenance.

Employing energy harvesters requires careful selection of harvesting source, converter and consumption circuits, and energy storage unit. Heat, vibration, and radiation are among common harvested energy types varying by availability and conversion efficiency factors. The choice of storage unit depends on the desirable output voltage, energy density, charge-discharge efficiency, memory effect, and weight [2]. Typically, energy harvesting solutions need to be integrated within the global energy management frameworks implemented in actual systems. Approaches to this problem include cross-layer duty cycling strategies [3] and relay node selection for data transfer [4]. It is nevertheless still an open question as to whether energy harvesting methods based upon predictable but *non-controllable* environmental energy sources can be used to support low-power wireless systems that demand performance-sensitive network performance. Due to the variability in the availability of the harvested energy, energy depletion and potential system shutdown might pose serious problems for safety-critical and industrial systems.

This paper considers deadline-driven cluster-based wireless sensor networks whose energy storage devices are powered by harvesting non-controllable, but predictable energy sources (such as solar or wind energy). Cluster-based WSNs offer multiple advantages such as increasing scalability, hierarchical routing, energy saving through data aggregation, and minimizing topology maintenance [5]. As shown in Fig. 1, a cluster-based system has a coordinator node that directly communicates with all of the other nodes under its control [6]. Each sensor node transmits its data to the cluster-head periodically. The cluster-head has to finish the data collection from all the nodes within a specified deadline. Once it has collected each node's data, the cluster head in turn communicates with the base station directly or through the network to complete data delivery. In this paper, we focus on intra-cluster communication. For example, consider a target tracking application in which all nodes in a site periodically send their recording to the cluster head. For timely tracking of the target, data delivery in every period must be guaranteed. Delayed data will be overridden by new contents and lose their importance.. Another example is utility companies and municipalities that require the constant monitoring of water flow, for the purposes of correctly billing customers, as well as for quickly detecting the main breaks. A WSN could harvest energy from sources such as solar, or miniature water dynamos, and provide constant monitoring of the lines. The objective is to ensure enough time and energy for all nodes along a time horizon to be able to complete their data transmission in a timely manner. This requires an energy aware time slot assignment for each epoch in the time horizon given the energy profile of the nodes.

Our focus in this paper is the use of *Dynamic Modulation Scaling* (DMS) as an energy management technique to support real-time energy harvesting WSNs [7]. Similar to the Dynamic Voltage Scaling technique that uses the processor voltage and

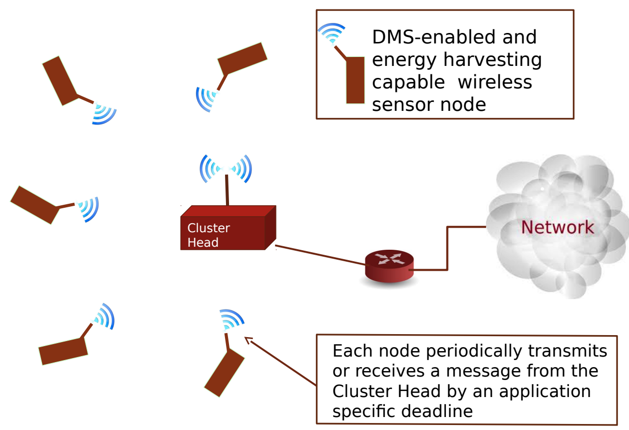

Fig. 1: Wireless energy harvesting nodes equipped with DMS radios communicating with a cluster-head

frequency as a control knob [8], [9], DMS adjusts the radio modulation levels and constellation sizes, and hence trades off the energy expenditure with communication latency [7]. In our target application, the nodes in the cluster communicate with the coordinator using DMS to save energy. By assigning time slots and modulation levels to individual nodes, the system's energy and timing constraints can be simultaneously satisfied. We are motivated to use DMS for several reasons. Since the radio is the dominant energy consumer within wireless sensor nodes, it is more beneficial to manage this trade-off than other energy management techniques, such as DVS [10]. Finally, DMS-enabled radios are supported by embedded wireless standards such as 802.15.4 [11].

A seminal work in theorizing the fundamentals of DMS is [7]. In order to schedule packets in time-invariant channels, the authors suggest deploying DMS in data link layer accompanied with TDMA. Packets are scheduled based on EDF and are assigned a static scaling factor based on their maximum size. This requires sending a control packet to communicate the factor with the receiver. For a single packet in a time-variant channel the authors propose a DMS approach based on sampling the channel quality. [12] targets maximizing system resilience to network-wide workload bursts, or avoiding shortage in harvested energy by maximizing the minimum battery level among the nodes in the network while meeting the system constraints. The authors use a joint DVS-DMS power management scheme. Nodes produce a list of all possible speed assignment combinations they can be assigned to, along with their respective remaining energy supplies. A binary search is then performed over all possible values of remaining battery level reported by nodes at the first level to check the feasibility. The above algorithm is implemented in both centralized and distributed versions. Our work is different since we guarantee energy neutrality at the end of period, and so we consider a larger scheduling horizon. The model in [13] addresses precedence, interference, and timing constraints to minimize joint communication and computation energy. A heuristic method is proposed that converts the problem into a graph model and assigns slots with maximum parallelism in order to achieve the maximum static slack. [14] studies dynamic slack reclamation in applications that discard redundant data by simply sending the header. The authors suggest nodes to probe the channel after transmission time of the header of their predecessors in the schedule. If channel was found free, they apply DMS to reclaim the dynamic slack due to redundancy. Our work is different from the above two references as the energy is restricted in our model by harvesting source.

In our setting, the problem is to determine the time slots and modulation levels that will be used by individual nodes to maximize the summation of node energy levels, while meeting the time constraints and energy neutrality requirements [15] given the predicted harvested energy. The central node periodically transmits a beacon packet that assigns time slots and modulation levels to all of the nodes in the system. Each node transmits using its time slot and modulation level. To find the optimum time slot and modulation level assignments, we have developed an optimal algorithm using mixed binary integer programming technique. We have also developed two fast heuristic algorithms. The mixed binary integer programming solution is optimal in the sense that it will find a *feasible* solution that avoids energy depletion while meeting communication deadlines, if one exists. It will also maximize the aggregate energy reserves of the nodes. The two fast algorithms reduce computational complexity at the risk of occasionally failing to find some feasible solutions, or generating a sub-optimal solution in terms of maximizing energy reserves. In our solution, energy-rich nodes are allowed to increase their communication speeds, thereby providing the low-energy nodes with large time slack and ability to slow down without violating the timing constraints.

We experimentally evaluate the tradeoffs between our optimal but more computationally intensive algorithm and the simpler heuristic approaches. Our results indicate that while the optimal algorithm enables the use of energy storage units with smaller capacities and tighter deadlines, the two fast algorithms are competitive in terms of maximizing energy reserves when they generate feasible solutions.

II. System Model

In a cluster-based WSN, each sensor node consists of an energy harvesting element such as a solar panel or a wind turbine, an energy storage unit such as rechargeable battery or super capacitor, a CPU, and a DMS-capable radio. Nodes communicate directly with a gateway or cluster-head, which serves as the system coordinator. The gateway will often, but not always, be connected to a back-end network. Compared to the nodes the gateway has ample computational capabilities and is powered either by easily replaceable batteries or a main power supply. Each node periodically senses its environment and after a processing step sends these readings in the form of

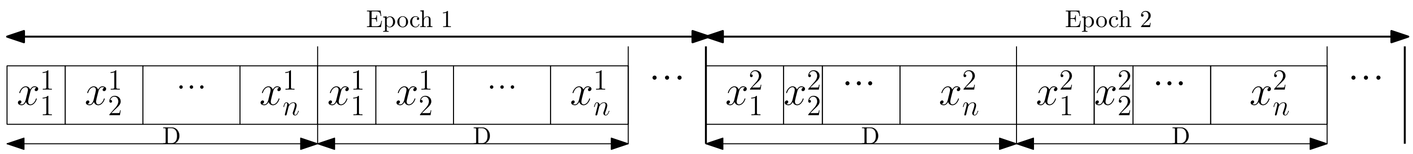

Fig. 2: Transmission slots in super-frames of length D in consecutive epochs

wireless data packets to the gateway. We assume the requirement for real-time communication between the gateway and each node, which is specified by a fixed deadline. To ensure real-time performance and to avoid channel contention, the cluster uses frame-based transmission scheduling techniques such as Time Division Multiple Access (TDMA) or Guaranteed Time Slot (GTS) assignments during the contention free period of the ZigBee/802.15.4 superframe [6].

The clustered topology and the presence of the computationally powerful gateway enable a coordinated set of DMS node level settings that guarantee, when possible, both real-time communication and energy neutrality constraints. The *energy neutrality* condition [15] requires that each node be able to sustain itself by harvesting energy and regulating its energy consumption according to the initial energy reserves as well as predicted harvesting profiles. Time is divided into a set of energy harvesting *epochs*. For concreteness and without loss of generalityy, we consider harvesting solar power and consider a 24-hour scheduling horizon. Epoch is a time length during which the rate of harvested energy is approximated to be constant. For example, in a period of 24 hours, 48 epochs of length 30 minutes exist. Using techniques such as those discussed in [16], the gateway possesses a function or table that predicts for each epoch the amount of energy that will be harvested by the nodes in the cluster. The gateway uses its knowledge of the currently available stored energy and the predicted amount of the newly harvested energy to assign modulation levels at each node for the entire epoch. The solution must make sure that all deadlines are met and the energy neutrality condition is satisfied, whenever possible.

The gateway is responsible for running the algorithms and then letting each node know what its modulation level will be for the duration of the epoch. As shown in Fig. 2, each epoch consists of a repeating set of super-frames, each of length D. The super-frame itself is divided into a set of variable-length transmission slots assigned to different nodes. x_i^j represents the transmit time the scheduler assigns to node i in each superframe during epoch j, which is adjustable through DMS. The transmission time for a node is identical for different super-frames within an epoch but may vary for different epochs. Hence, with γ super-frames within each epoch, node i receives a total of $t_i^j = x_i^j \cdot \gamma$ transmission time during epoch j. Although not shown in the figure, in practice each super-frame could also include space for a gateway generated beacon used for management purposes.

The energy consumption of a node during an epoch is the sum of sensing, processing, and radio transmission energy figures. Our previous work [17] shows that managing transmit energy yields more energy saving than CPU energy management. We assume sensing and processing energy components are constant and only the radio energy consumption is manageable. Energy consumption due to radio usage has two components: electronic circuitry power and transmit (receive) power [7]. Electronic circuitry power consumption P_e is due to activities such as filtering, modulation and upconverting. It is linearly proportional to the symbol rate R_s, and can be expressed as $P_e = C_e \cdot R_s$, where C_e is a radio specific constant. The transmit power P_s is a function of the modulation level b and is given by $P_s = (C_s \cdot \phi(b)) \cdot R_s$., For DMS the modulation level is taken from a set of size B, $\{b_1, b_2, ..., b_B\}$ where $b_i < b_{i+1}, i = 1, \ldots, B-1$. The function $\phi(b)$ depends on the modulation scheme. For example, for QAM, $\phi(b) = 2^b - 1$. The coefficient C_s is constant with respect to modulation level and depends on receiver implementation, operating temperature, distance, and propagation environment. The time required to send one bit is calculated as:

$$T_{bit}(b) = \frac{1}{R_s \cdot b} \tag{1}$$

Assuming QAM and combining the expressions above for electronic circuitry power, transmit power, and transmit time, the energy to send one bit can be derived as [7]:

$$E_{bit}(b) = (P_s + P_e) \cdot T_{bit}(b) = \frac{C_s \cdot (2^b - 1) + C_e}{b} \tag{2}$$

We observe that the energy consumption is independent of the symbol rate, and decreasing the modulation level substantially reduces energy consumption, at the expense of higher transmission time. This observation demonstrates the potential importance of DMS. The amount of DMS effectiveness is reflected in the ratio of $\frac{C_s}{C_e}$. A high ratio shows when dynamic power is the dominant factor and therefore applying DMS is most effective. In other words, there is an *effective modulation level* below which DMS is no longer beneficial, as shown in [7]. Its value can analytically be determined by setting the first derivative of the radio energy to zero. We are assuming that the minimum modulation level b_1 is determined by considering the maximum of the effective modulation level, and the lowest level supported by the modulation scheme.

M packets each of length ρ bits are sent by each node in every super-frame. Each node is assigned a fixed modulation level in a given epoch, but the assignment may vary from epoch to epoch, and from node to node. Since there are γ super-frames within each epoch, during the j^{th} epoch, for the i^{th} node with modulation level of b_i^j, the total communication time and energy consumption are given by $t_i^j = (\gamma \cdot M \cdot \rho) \cdot T_{bit}(b_i^j)$ and $e_i^j = (\gamma \cdot M \cdot \rho) \cdot E_{bit}(b_i^j)$, respectively. Our algorithms assume that if all the nodes use the maximum modulation levels, in the absence of the energy constraints, it is possible to meet the deadline in each super-frame; i.e., $n \cdot \frac{M \cdot \rho}{R_s \cdot b_B} \leq D$. Obviously no solution can exist if that condition is not satisfied.

Due to the short duration of an epoch, (e.g., 15-30 minutes), we assume that the harvesting power is constant in an epoch [18]. There are several techniques for forecasting how much energy will be harvested during the epoch from uncontrollable but predictable environmental sources, including using an autoregressive filter [15], a round-based approach [19], or an exponentially weighted moving average method [20]. In order to consider environmental temporary conditions, [21] introduces a short term factor as the ratio of observed and predicted energy in the last epoch. This, combined with a degrading weight factor is applied to the predicted model to improve the accuracy. Our approach is independent of the prediction technique but assumes its availability.

P_i^j denotes the harvested power prediction during epoch j, for node i. The process of energy conversion from the energy harvesting panel to electrical energy consumable by the node components is subject to several harvesting, conversion, storing, and consumption inefficiencies. We assume that these factors are already analyzed and factored in the reported value of P_i^j. On node i, the harvested energy is stored in the energy storage unit with capacity J_i. The energy level at the beginning of the first epoch and the target energy at the end of the operation period (with Υ epochs) are denoted by E_i^0 and E_i^{Target}, respectively. The energy level of node i at the end of epoch j is denoted by L_i^j.

III. Problem Formulation

We now formulate the problem we are addressing. Assume there are Υ consecutive epochs for which we have available energy predictions. Given a set of n wireless energy harvesting nodes, each equipped with DMS-capable radios and operating in an environment with a known energy harvesting profile, we aim to determine the modulation level for each node in every epoch, so as to *maximize the sum of energy levels at the end of the* Υ *epoch*, while *guaranteeing the deadline in every communication super-frame* and *ensuring the energy neutrality throughout the operation.* Specifically, the optimization problem can be formulated as:

$$\text{Maximize} \quad \sum_{i=1}^{n} L_i^{\Upsilon} \tag{3}$$

$$\text{Subject to} \quad \sum_{i=1}^{n} x_i^j \leq D, \forall j : 1 \leq j \leq \Upsilon \tag{4}$$

$$0 < L_i^j \leq J_i, \forall i, j : 1 \leq i \leq n, 1 \leq j \leq \Upsilon - 1 \tag{5}$$

$$E_i^{Target} \leq L_i^{\Upsilon} \leq J_i, \forall i : 1 \leq i \leq n \tag{6}$$

The objective is to maximize the sum of remaining energy levels across all nodes at the end of the Υ epoch, $\sum_{i=1}^{n} L_i^{\Upsilon}$, as specified in (3). The real-time communication performance requirement is encoded in the constraint set (4): all nodes should complete their transmission within time D, the length of the super-frame. The choice of the objective function is to increase the system's resilience against temporary changes in energy resource and potential prediction inaccuracies.

In addition, it is necessary to avoid both battery overflow and underflow conditions, as well as guarantee energy neutrality (constraint sets (5) and (6)). Prevention of the battery underflow condition assures the battery level never drops to zero. By preventing the battery overflow we make sure that the battery level at the end of each super-frame does not exceed the node's storage capacity. Finally, the energy neutrality constraint will ensure that the energy level at the end of the last epoch should not be less than a given target level E_i^{Target}. A natural choice is $E_i^{Target} = E_i^0$, which makes sure that the system is able to sustain itself by relying on the harvested power only [15].

Given that the modulation level for a given node i is only one of the B distinct values, we can use a binary indicator variable α_{il}^j to specify whether the l^{th} modulation level was selected for node i in epoch j, or not. In fact, after a series of additional algebraic manipulations, it is possible to re-encode the problem as an instance of a *Mixed Binary Integer Programming Problem*. The full steps of the derivation can be found in the appendix.

IV. Heuristic Algorithms

The Mixed-Integer Programming is known to be NP-Hard; however, problem instances of moderate size can be solved by existing optimization packages. Nonetheless it is very desirable to develop schemes that run fast and yield good performance while satisfying all the constraints. For this purpose, we developed and evaluated two fast algorithms, which are described next.

A. Uniform Modulation Level Assignment

A basic heuristic is to assign equal modulation levels (transmission times) for all nodes and epochs. This *uniform* assignment is not energy aware and does not discriminate between energy-poor and energy-rich nodes. Consequently, the slot assignment is the same for all epochs. Because the energy consumption is an increasing function of the modulation level, the problem reduces to finding the smallest modulation level that meets the deadline during a super-frame:

$$b^* = \min\{b_k | n \cdot \frac{M \cdot \rho}{R_s \cdot b_k} \leq D\}$$

The term $n \cdot \frac{M \cdot \rho}{R_s \cdot b_k}$ in the above equation is the total time that it takes for all nodes in the cluster to complete their transmission with modulation level of b_k. We are interested in the smallest value of b_k that meets the deadline constraint since further increasing the modulation level only increases the energy consumption which is clearly against the nature of the objective function. The minimum time-feasible *uniform* modulation level is also energy-feasible if and only if it also meets the energy constraints. Otherwise, no larger value of modulation level will meet those constraints. Smaller modulation levels ($b < b^*$) will not meet the time constraints. Evaluating b^* may involve scanning through all choices of modulation levels in the worst case. For a candidate b value, the scheme (whose pseudo-code is provided in Algorithm 1) computes the energy levels at the end of all the epochs for all nodes (the L_i^j values), and checks the energy neutrality conditions. Hence the overall time complexity is linear in the number of nodes, specifically, $O(B + n \cdot \Upsilon)$. The algorithm returns the index of the common modulation level and an indicator showing if the solution was energy-feasible or not.

Algorithm 1 *Uniform* Modulation Level Assignment

$r = 1$
while $(n \cdot M \cdot \rho) \cdot T_{bit}(b_r) > D$ **do**
  $r = r + 1$
$e(b_r) = (\gamma \cdot M \cdot \rho) \cdot E_{bit}(b_r)$
for $i = 1 : n$ **do**
  for $j = 1 : \Upsilon$ **do**
    $L_i^j = \min\{J_i, L_i^{j-1} + P_i^j \cdot D \cdot \gamma - e(b_r)\}$
for $i = 1 : n$ **do**
  for $j = 1 : \Upsilon - 1$ **do**
    if $L_i^j < 0$ **then**
      return (r, Infeasible)
for $i = 1 : n$ **do**
  if $L_i^{\Upsilon} < E_i^{Target}$ **then**
    return (r, Infeasible)
return (r, Feasible)

B. Greedy Modulation Level Assignment

One major drawback of *Uniform* is that some nodes are prevented from lowering their modulation levels (i.e., they do not exploit the maximum transmission slack that can be used by DMS). In fact, there may be problem instances where the energy-feasibility cannot be satisfied by assigning a uniform modulation level, while reducing the levels of individual nodes may lead to a feasible solution.

Our *Greedy* scheme (Algorithm 2) takes the output of the *Uniform* heuristic as the base case. Specifically, the lowest uniform modulation level b_r that meets the deadline constraint is taken as the initial assignment for all the nodes. Then the algorithm iterates over all the epochs and all the nodes, and attempts to reduce the modulation levels of selected nodes by exactly one level (i.e., their modulation levels are set to b_{r-1}). In each iteration, and for every epoch, the algorithm tentatively selects the nodes with the minimum amount of remaining energy for the purpose of slowdown. Decreasing the modulation level of a node within one epoch only by one level gives other nodes a chance to reduce their transmission speed as well. This is further justified by the convexity of energy consumption function and the objective of saving energy as much as possible. After such adjustments, the energy neutrality and feasibility conditions are re-checked.

Algorithm 2 *Greedy* Modulation Level Assignment

Set $r = Uniform()$
Set $b_i^j = b_r \ \forall i, j : 1 \le i \le n, 1 \le j \le \Upsilon$
if $r == 1$ **then**
  if $Uniform$ returned Infeasible **then**
    return (Infeasible)
  else
    return $(\{b_i^j\}) \forall i, j : 1 \le i \le n, 1 \le j \le \Upsilon$
/* $r > 1$ and *Greedy* starts iterations */
$t(b_r) = \gamma \cdot M \cdot \rho \cdot T_{bit}(b_r)$
$t(b_{r-1}) = \gamma \cdot M \cdot \rho \cdot T_{bit}(b_{r-1})$
for $j = 1 : \Upsilon$ **do**
  for $i = 1 : n$ **do**
    $L_i^j = \min\{J_i, L_i^{j-1} + P_i^j \cdot D \cdot \gamma - e_i^j\}$
  Sort all L_i^j values for all the nodes in the j^{th} epoch
  $Slack^j = D \cdot \gamma - \sum_{i=1}^{n} t_i^j$
  $Q = \{1, \ldots, n\}$
  while $Q \neq \emptyset$ and $Slack^j \ge (t(b_{r-1}) - t(b_r))$ **do**
    $Slack^j = Slack^j - (t(b_{r-1}) - t(b_r))$
    $index$ = node index with minimum L_i^j value in Q
    $b_{index}^j = b_{r-1}$
    $L_{index}^j = \min\{J_i, L_{index}^{j-1} + P_{index}^j \cdot D \cdot \gamma - e_{index}^j\}$
    $Q = Q - \{index\}$
  for $i = 1 : n$ **do**
    if $L_i^j < 0$ **then**
      return (Infeasible)
for $i = 1 : n$ **do**
  if $L_i^{\Upsilon} < E_i^{Target}$ **then**
    return (Infeasible)
return $(\{b_i^j\}) \forall i, j : 1 \le i \le n, 1 \le j \le \Upsilon$

The algorithm invokes *Uniform* at the beginning. Moreover, it requires sorting the remaining energy levels at the end of each epoch. Each sort takes $O(n \log(n))$, and there are Υ total epochs. At most $n-1$ nodes will require changing modulation level, the complexity of which is outweighed by the sorting complexity. As a result, the algorithm has an overall time complexity of $O(\Upsilon \cdot n \cdot \log(n) + B)$.

V. Experimental Evaluation

In order to evaluate the performance differences between the optimal algorithm and fast heuristics under a wide variety of experimental settings, we implemented a simulation system using Matlab. We used the IBM Cplex toolbox to solve the mixed integer programming problems.

For energy harvesting we used the solar profile trace measurements from [22]. The measurement was obtained by sampling solar radiation at intervals of 30 seconds during a two-month period in Hamburg, Germany. Due to space limitations, we present results for three representative days in that period. In a cluster with nodes distributed across an area,

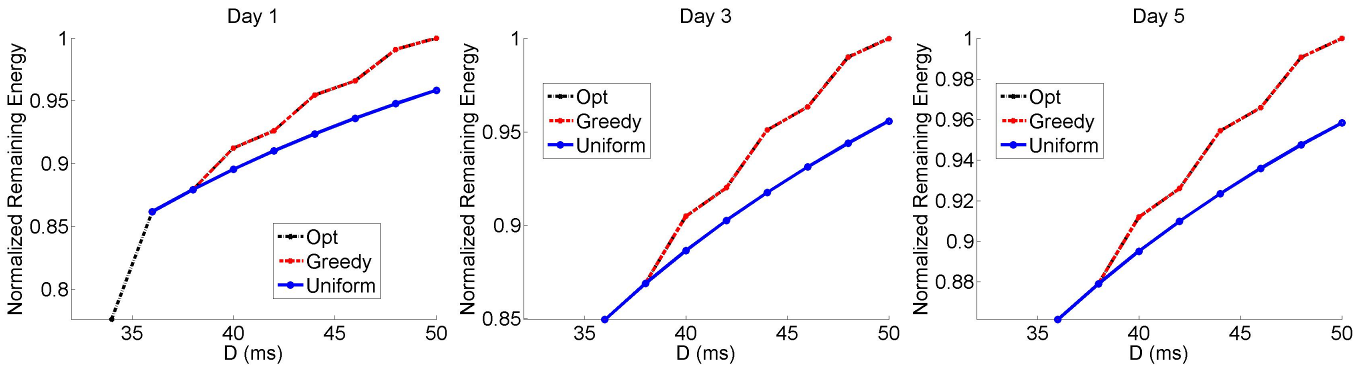

Fig. 3: Effect of the super-frame size (deadline) on the performance of different algorithms over the course of Day 1, 3, 5

nodes may not have completely homogeneous solar energy collection profiles. To reflect this effect, we define a spatial absorption coefficient for each node that varies randomly from 90 to 100 percent of the solar profile value. The initial energy level of the nodes (E_i^0) also varies by a factor up to 10% in different nodes. To enforce energy neutrality, we require $E_i^{Target} = E_i^0$ for all nodes.

For DMS, we adopted the radio parameters from [7] by assigning $C_s = 12 \cdot 10^{-9}$, $C_e = 15 \cdot 10^{-9}$ and the symbol rate of $R_s = 62500Hz$. Nodes communicate via packets of size $\rho = 128$ bytes. Each node transmits one packet within a single super-frame. The cluster has $n = 10$ nodes, and the daily energy profile is divided into $\Upsilon = 48$ epochs in our simulation. The length/deadline of a single super-frame is selected in the range of $[25, 80]ms$. There are four available modulation levels, selected from the set $\{2, 4, 6, 8\}$.

The impact of the super-frame size: We first analyze the impact of the super-frame size (deadline) on the performance. Fig. 3 shows the normalized energy levels for different algorithms over the course of three days. In data points that one scheme does not have a value, that scheme cannot come up with a feasible solution. In all these experiments, the initial energy is set to the half the battery capacity (i.e., all nodes start at 50% energy levels). Performance of the algorithms is measured in terms of the remaining energy of all nodes when a feasible schedule is available. In general, larger super-frame sizes produce higher levels of remaining energy. This is because large super-frame sizes imply larger deadlines, which enable the system to use slower transmission speeds for individual nodes using DMS, leading to more energy savings. Notice that for Day 1, only the Optimal algorithm could find a feasible schedule when $D < 37ms$. Above this threshold *Greedy* performs as well as the optimal approach, while the simpler *Uniform* achieved energy values within 10% of the other two. We repeated this experiment over a week and in four of the days, there was at least one point for which only *Optimal* was able to find a feasible solution. The days were chosen to show the observed patterns. The plots of the remaining days are removed to avoid repetition.

Minimum battery size needed for energy neutrality: Larger batteries both impose extra monetary cost and increase the weight and size of each energy harvesting node. It is therefore desirable to achieve the energy neutrality objectives with smaller batteries. The next experiments investigate the minimum battery size required to guarantee the feasibility for each algorithm on Day 1 and Day 3 as a function of the initial energy levels (Fig. 4). The super-frame size in this setting is $80ms$, and the energy values and battery storage capacities are normalized. *Optimal* requires the smallest battery capacity for every value of the initial energy among

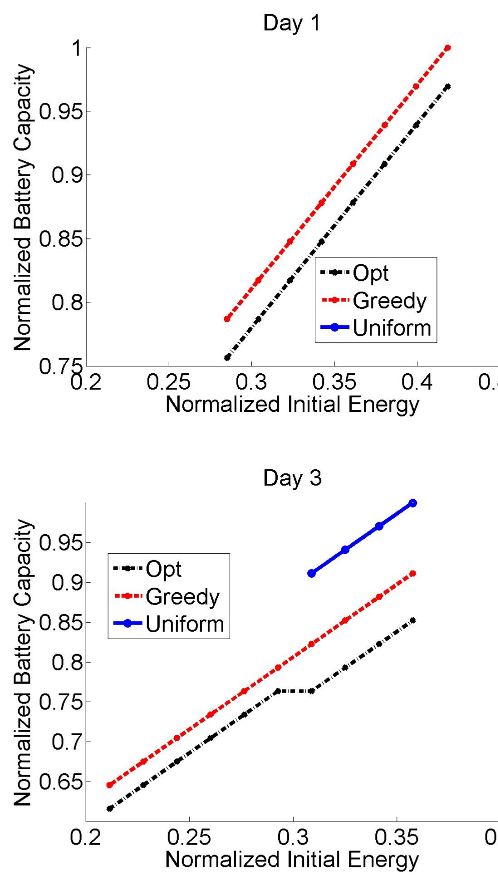

Fig. 4: Effect of the initial energy level on the minimum battery capacity necessary to guarantee time- and energy-feasibility

all schemes. The size of battery increases almost linearly with initial energy for all of the algorithms. This is because as we increase the initial energy, the minimum target energy level to guarantee energy neutrality also grows (due to the $E^{target} \geq E^0$ condition): the battery must be large enough to store the large initial energy as well as the energy required to sustain the system for dark epochs at the end of the day. Also it is interesting to note that for the entire Day 1 experiments, and Day 3 when $E_{initial} \leq 0.3$, *Uniform* fails to generate a feasible schedule, regardless of the battery capacity. This is because with low initial energy, the systems cannot sustain themselves by relying on harvested energy only, when the simple assignment of *Uniform* is used. This fact justifies again the use of more intelligent algorithms for modulation level assignments in conjunction with DMS.

Next, Fig. 5 shows how the super-frame size affects the minimum battery capacity to guarantee feasible solution for Day 1 and Day 3. As expected, for all schemes larger super-frame sizes required smaller battery storage capacities, as larger deadlines enable the system to save more energy through DMS. *Optimal* outperforms *Greedy* and *Uniform* for large super-frame sizes. It is interesting to note that, for Day 1, *Uniform* does not produce any feasible solution, irrespective of the super-frame size.

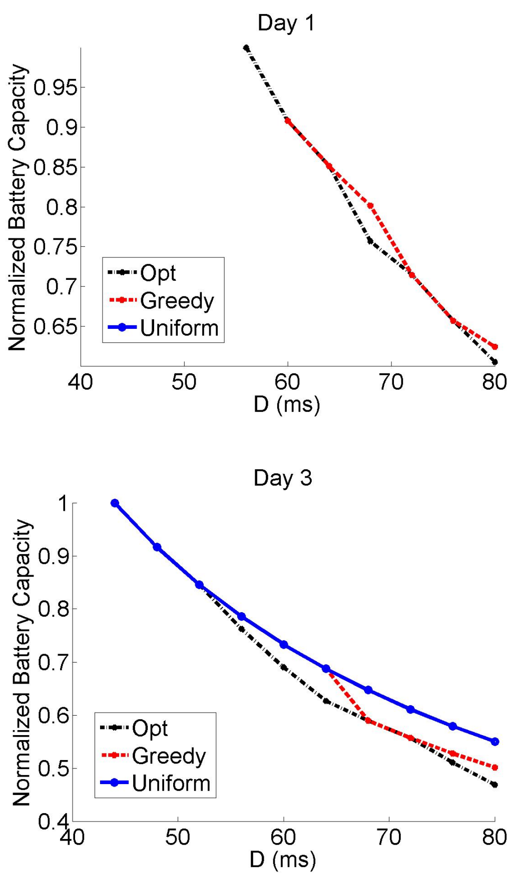

Fig. 5: Effect of the super-frame size on the minimum battery capacity necessary to guarantee time- and energy-feasibility

VI. Conclusions

In this paper, we considered cluster-based energy-harvesting wireless sensor networks with timeliness requirements. In this framework, the nodes have to transmit their readings to the cluster head in a timely manner to preserve the functionality. We considered the impact of Dynamic Modulation Scaling on these systems. The goal is to maximize the energy reserves while meeting the timing constraints under the energy neutrality conditions, which requires that the consumed energy never exceeds the sum of harvested and stored energy during operation. We proposed an optimal mixed integer programming model to dynamically set modulation periods over an epoch for which the amount of energy harvested could be predicted. We also proposed two fast heuristics that traded off optimality for a reduction in computational complexity. Using a real solar trace we then compared the performance of all three algorithms. We found that when very tight deadlines are required only the Optimal approach can produce feasible schedules, but as requirements are relaxed the heuristic approaches suffice. We determined that if minimizing battery size and capacity is the major requirement then the Optimal approach substantially improves the number of feasible schedules. We also showed that our proposed energy management framework considerably improves system life at a lower battery cost.

Acknowledgments

This work has been supported by the US National Science Foundation awards CNS-1421855 and CNS-1116122.

References

[1] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *Communications Surveys Tutorials, IEEE*, vol. 13, no. 3, pp. 443–461, 2011.

[2] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 3, pp. 443–461, 2011.

[3] P. Glatz, L. Hormann, C. Steger, and R. Weiss, “Hans: Harvesting aware networking service for energy management in wireless sensor networks,” in *Telecommunications, 18th Int. Conf. on*, 2011.

[4] B. Medepally and N. Mehta, “Voluntary energy harvesting relays and selection in cooperative wireless networks,” *Wireless Communications, IEEE Trans. on*, vol. 9, no. 11, pp. 3543–3553, 2010.

[5] B. Mamalis, D. Gavalas, C. Konstantopoulos, and G. Pantziou, “Clustering in wireless sensor networks,” *RFID and Sensor Networks: Architectures, Protocols, Security and Integrations*, pp. 324–353, 2009.

[6] M.-G. Park, K.-W. Kim, and C.-G. Lee, “Holistic optimization of real-time ieee 802.15.4/zigbee networks,” in *Advanced Information Networking and Applications, IEEE Int. Conf. on*, 2011.

[7] C. Schurgers, V. Raghunathan, and M. B. Srivastava, “Power management for energy-aware communication systems,” *Embedded Computing Systems, ACM Trans. on*, vol. 2, no. 3, pp. 431–447, 2003.

[8] P. Pillai and K. G. Shin, “Real-time dynamic voltage scaling for low-power embedded operating systems,” in *ACM SIGOPS Operating Systems Review*, vol. 35, pp. 89–102, ACM, 2001.

[9] D. Zhu, H. Aydin, and J.-J. Chen, “Optimistic reliability aware energy management for real-time tasks with probabilistic execution times,” in *Real-Time Systems Symposium, 2008*, pp. 313–322, IEEE, 2008.

[10] G. Sudha Anil Kumar, G. Manimaran, and Z. Wang, “Energy-aware scheduling of real-time tasks in wireless networked embedded systems,” in *Real-Time Systems Symposium, Proc. of the 28th IEEE Int.*, 2007.

[11] “IEEE Standard for Local and metropolitan area networks.” http://standards.ieee.org/findstds/standard/802.15.4g-2012.html.

[12] B. Zhang, R. Simon, and H. Aydin, “Harvesting-aware energy management for time-critical wireless sensor networks with joint voltage and modulation scaling,” *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 514–526, 2013.
[13] B. Fateh and M. Govindarasu, “Joint scheduling of tasks and messages for energy minimization in interference-aware real-time sensor networks,” *Mobile Computing, IEEE Transactions on*, vol. 14, pp. 86–98, Jan 2015.
[14] B. Fateh and M. Govindarasu, “Energy minimization by exploiting data redundancy in real-time wireless sensor networks,” *Ad Hoc Networks*, vol. 11, no. 6, pp. 1715–1731, 2013.
[15] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *Embedded Computing Systems, ACM Trans. on*, vol. 6, no. 4, 2007.
[16] N. Bartolini, T. Calamoneri, T. La Porta, C. Petrioli, and S. Silvestri, “Sensor activation and radius adaptation (sara) in heterogeneous sensor networks,” *ACM Trans. Sen. Netw.*, vol. 8, pp. 24:1–24:34, Aug. 2012.
[17] M. Bandari, R. Simon, and H. Aydin, “Energy management of embedded wireless systems through voltage and modulation scaling under probabilistic workloads,” in *Green Computing Conference (IGCC), 2014 International*, pp. 1–10, Nov 2014.
[18] C. Moser, J.-J. Chen, and L. Thiele, “An energy management framework for energy harvesting embedded systems,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 6, no. 2, pp. 7:1–7:21, 2008.
[19] K. Kinoshita, T. Okazaki, H. Tode, and K. Murakami, “A data gathering scheme for environmental energy-based wireless sensor networks,” in *Consumer Communications and Networking, 5th IEEE Conf. on*, 2008.
[20] A. Kansal and M. B. Srivastava, “An environmental energy harvesting framework for sensor networks,” in *Low Power Electronics and Design, Proc. of the Int. Symposium on*, ACM, 2003.
[21] D. K. Noh, L. Wang, Y. Yang, H. K. Le, and T. Abdelzaher, “Minimum variance energy allocation for a solar-powered sensor system,” in *Distributed Computing in Sensor Systems*, pp. 44–57, Springer, 2009.
[22] B. Latré, P. D. Mil, I. Moerman, B. Dhoedt, P. Demeester, and N. V. Dierdonck, “Throughput and delay analysis of unslotted ieee 802.15. 4,” *Journal of Networks*, vol. 1, no. 1, pp. 20–28, 2006.

APPENDIX

DETAILS OF THE MIXED INTEGER PROGRAMMING FORMULATION

In this section, we provide the details of the Mixed Integer Linear Programming formulation, which is basis for the optimal solution.

Objective Function

Our objective function is to maximize the overall remaining energy levels of all nodes at the end of last epoch (epoch #Υ).

$$Maximize \sum_{i=1}^{n} L_i^{\Upsilon}$$

A closed form expression for the epoch-end energy levels (L_i values) requires expressing the energy consumption of each node as a function of the assigned modulation level. Specifically, a node requires $e(b)$ amount of energy to send M packets of size ρ bits over the channel in one epoch, when using the modulation level b:

$$e(b) = \gamma \cdot M \cdot \rho \cdot \frac{C_s(2^b - 1) + C_e}{b}$$

Let us introduce a binary indicator variable α_{il}^j to indicate the corresponding modulation level choices from the set $\{b_1, \cdots, b_B\}$ for node i during epoch j. Specifically, α_{il}^j is set to 1 if and only if the l^{th} modulation level was selected for node i in epoch j; otherwise, it is 0. Then, the energy consumption of a node i, in a given epoch j, can be expressed as:

$$e_i^j = \alpha_{i1}^j \cdot e(b_1) + \alpha_{i2}^j \cdot e(b_2) + \cdots + \alpha_{il}^j \cdot e(b_l) + \cdots + \alpha_{iB}^j \cdot e(b_B) \tag{7}$$

We add a constraint to show that exactly one of the B options for modulation level will be chosen for node i in epoch j:

$$\sum_{l=1}^{B} \alpha_{il}^j = 1, \forall i, j : 1 \le i \le n, 1 \le j \le \Upsilon$$

Epoch Energy Levels

For any node, the available energy at the end of each epoch equals available energy at the end of the preceding epoch plus the difference between the harvested and consumed energy amounts during that epoch. Considering that node i cannot store more than J_i units of energy, the remaining energy level at node i at the end of epoch j may be obtained as:

$$L_i^j = \min\{J_i, L_i^{j-1} + \gamma \cdot D \cdot P_i^j - e_i^j\}$$

The overflow energy that may not be stored in the battery of node i at the end of epoch j due to the capacity limits can be expressed by variable θ_i^j (the excess energy which cannot be stored and which is dissipated as heat). Overflow variables are non-negative real numbers: $\theta_i^j \ge 0$. We can write:

$$\begin{aligned} L_i^j &= L_i^{j-1} + \gamma \cdot D \cdot P_i^j - e_i^j - \theta_i^j \\ &= E_i^0 + \sum_{k=1}^{j} \left(\gamma \cdot D \cdot P_i^k - e(b_i^k) - \theta_i^k \right) \\ &= E_i^0 + \sum_{k=1}^{j} \left(\gamma \cdot D \cdot P_i^k \right) \\ &\quad - \frac{\gamma \cdot M \cdot \rho}{b_i^k} \left(C_s \cdot \left(2^{b_i^k} - 1 \right) + C_e \right) - \theta_i^k \end{aligned}$$

Then the objective function can be re-written as:

$$Maximize \sum_{i=1}^{n} E_i^0 + \sum_{k=1}^{\Upsilon} \left(\gamma \cdot D \cdot P_i^k - e_i^k - \theta_i^k \right)$$

By substituting the value of e_i^j from Eq. (7), we obtain:

$$Maximize \sum_{i=1}^{n} E_i^0 + \\ \sum_{k=1}^{\Upsilon} \left(\gamma \cdot D \cdot P_i^k - \frac{\gamma \cdot M \cdot \rho}{b_i^k} \left(C_s \cdot \left(2^{b_i^k} - 1 \right) + C_e \right) - \theta_i^k \right)$$

Since E_i^0 and $\gamma \cdot D \cdot P_i^k$ are not a function of the modulation level, this is equivalent to:

$$Minimize$$
$$\sum_{i=1}^{n}\sum_{k=1}^{\Upsilon}\left(\frac{\gamma\cdot M\cdot\rho}{b_i^k}\left(C_s\cdot\left(2^{b_i^k}-1\right)+C_e\right)+\theta_i^k\right) \tag{8}$$

By using our binary indicator variables $\{\alpha_{il}^k\}$, this is equivalent to:

$$Minimize$$
$$\sum_{i=1}^{n}\sum_{k=1}^{\Upsilon}\sum_{l=1}^{B}\left(\frac{\alpha_{il}^k\cdot\gamma\cdot M\cdot\rho}{b_l}\left(C_s\cdot\left(2^{b_l}-1\right)+C_e\right)+\theta_i^k\right) \tag{9}$$

Time Constraints

The real-time characteristic of the application requires all nodes to complete their transmission within the deadline of each super-frame, D:

$$\sum_{i=1}^{n}\frac{t_i^j}{\gamma}=\sum_{i=1}^{n}\left(\frac{M\cdot\rho}{R_s\cdot b_i^j}\right)\leq D, \qquad \forall 1\leq j\leq\Upsilon \tag{10}$$

Energy Constraints

The energy consumption characteristics must prevent both battery overflow and underflow conditions while guaranteeing energy neutrality. The prevention of battery underflow assures that the battery level never drops to zero. We may combine the energy neutrality condition with the battery underflow condition by defining variable ζ_i^j that shows the minimum allowable battery level of node i at the end of epoch j.

Obviously, $\zeta_i^j=0$, $1\leq j\leq\Upsilon-1$, and, $\zeta_i^\Upsilon=E_i^{Target}=E_i^0$:

$$\begin{aligned}L_i^j &= E_i^0+\\ &\sum_{k=1}^{j}\left(\gamma\cdot D\cdot P_i^k-\frac{\gamma\cdot M\cdot\rho}{b_i^k}\left(C_s\cdot\left(2^{b_i^k}-1\right)+C_e\right)-\theta_i^k\right)\\ &\geq\zeta_i^j, \qquad \forall i,j:\,1\leq i\leq n,1\leq j\leq\Upsilon\end{aligned}$$

Preventing battery overflow makes sure that the excess energy which cannot be stored in the battery is not taken into account to guarantee the energy constraints of the subsequent epochs. In other words, the battery level at the end of each super-frame should not exceed the node's battery capacity. Combining all these constraints, we get:

$$\begin{aligned}\zeta_i^j &\leq E_i^0+\\ &\sum_{k=1}^{j}\left(\gamma\cdot D\cdot P_i^k-\frac{\gamma\cdot M\cdot\rho}{b_i^k}\left(C_s\cdot\left(2^{b_i^k}-1\right)+C_e\right)-\theta_i^k\right)\\ &\leq J_i \qquad \forall i,j:\,1\leq i\leq n,1\leq j\leq\Upsilon\end{aligned}$$

Putting all the constraints together, we obtain the following mixed integer programming formulation:

$$\begin{aligned}\underset{\theta,\alpha}{\text{Minimize}}\quad &\sum_{i=1}^{n}\sum_{k=1}^{\Upsilon}\sum_{l=1}^{B}(\frac{\alpha_{il}^k\cdot\gamma\cdot M\cdot\rho}{b_l}(C_s\cdot(2^{b_l}-1)+C_e)\\ &+\theta_i^k)\\ \text{Subject to}\quad &\sum_{i=1}^{n}\sum_{l=1}^{B}\left(\frac{\alpha_{il}^j\cdot M\cdot\rho}{R_s\cdot b_l}\right)\leq D,\,\forall j:\,1\leq j\leq\Upsilon\\ &\zeta_i^j\leq E_i^0+\sum_{k=1}^{j}(\gamma\cdot D\cdot P_i^k\\ &-\sum_{l=1}^{B}(\frac{\alpha_{il}^k\cdot\gamma\cdot M\cdot\rho}{b_l}(C_s\cdot(2^{b_l}-1)+C_e))-\theta_i^k)\\ &\leq J_i,\,\forall i,j:\,1\leq i\leq n,1\leq j\leq\Upsilon\\ &\alpha_{il}^j\in\{0,1\},\\ &\forall i,j,l:\,1\leq i\leq n,1\leq j\leq\Upsilon,1\leq l\leq B\\ &\sum_{l=1}^{B}\alpha_{il}^j=1,\,\forall i,j:\,1\leq i\leq n,1\leq j\leq\Upsilon\\ &\theta_i^j\geq 0,\forall i,j:\,1\leq i\leq n,1\leq j\leq\Upsilon\end{aligned}$$