# Orchestrated Application Quality and Energy Storage Management in Solar-Powered Embedded Systems

Nga Dang, Hossein Tajik, Nikil Dutt, Nalini Venkatasubramanian, Eli Bozorgzadeh
University of California, Irvine, USA
E-mail: {ngad, tajikh, dutt, nalini, ebozorgz}@uci.edu

## Abstract

While energy harvesting technology is a promising solution toward achieving self-sustainable low power systems, the efficient energy storage for these energy harvesting systems is still a challenge because of high self-leakage (e.g., supercapacitors) and limited life cycles (e.g., batteries). In this work, we propose an adaptive quality-aware energy management middleware framework for energy harvesting embedded systems. Our hybrid energy storage model takes into consideration the battery life cycle, supercapacitor self-leakage, and power loss in the harvesting circuit. The framework has an offline planning phase and a runtime adaptation phase. By incorporating abstract models for battery state of health (*SoH*) and supercapacitor self-leakage, the offline stage determines the budget for charging and discharging distribution of each storage component and accordingly adapts the application quality of service (*QoS*). The runtime adaptation phase dynamically adjusts the charging and discharging distribution to the dynamic changes in energy harvesting profile. In comparison with related work, our proposed framework is able to capture the lifetime and characteristics of the energy storage components more accurately during adaptation and hence, resulting in a more sustainable system with realistic QoS.

## Keywords

Energy harvesting, hybrid energy storage, battery lifetime, Quality of Service

## 1. Introduction

Renewable energy technology is a viable and promising solution for low power networked embedded systems. Environmental energy sources are ubiquitous in our surroundings and each system can be equipped with an energy harvesting circuitry to scavenge the energy from such sources. However, the spatial and temporal variations in scavenging energy from the environment lead to uncertainty in energy availability during operation. Temporal variations refer to energy harvesting condition changes in time (e.g., day vs. night) while spatial variations refer to energy harvesting condition changes in space (e.g., locations under shadow vs. locations with direct sunlight). These variations continue to challenge the energy sustainability in embedded systems.

A solution to mask this uncertainty is using energy storage as a buffer to smooth down the variations in energy harvesting. The choice of energy storage is important

This work is partially funded by NSF CAREER #0846129.

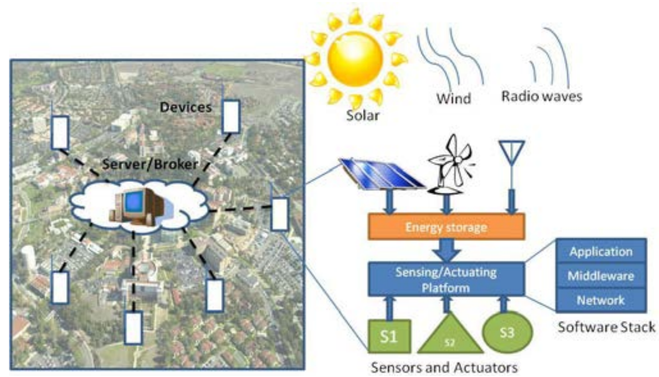


**Figure 1:** Networked Energy Harvesting Multi-Sensor Systems

because each type of energy storage has its own advantage and disadvantage. While batteries provide energy storage for a long period of time, they are constrained by limited *battery lifetime* (i.e., limited charging/discharging cycle counts) and *power density* (maximum charging/discharging current). On the other hand, lifetime of supercapacitors, another type of energy storage, is not a concern. Yet they have non-negligible *leakage* and lower *energy density*. As a result, using combination of supercapacitors and batteries, i.e., hybrid energy storage is currently the state-of-the-art practice in energy harvesting systems [5][2].

Our target solar-powered embedded system is a multi-sensor platform supporting simultaneous multiple sensing and/or processing applications (see Figure 1). The system transfers data from its sensors to a server via wireless connection. The energy scavenged from solar panels is stored in a hybrid energy storage subsystem. Our model of the hybrid energy storage has a switch-based architecture, composed of a rechargeable battery and a supercapacitor (similar to [4]).

Continuous sensing/processing while handling energy harvesting variations and hybrid energy storage constraints mandate harvesting embedded systems to employ an energy storage management scheme. The management scheme not only needs to be aware of the amount of energy being harvested but also the load and energy demand from the applications. If the applications demand energy aggressively without being aware of the energy storage status, the energy storage may get exhausted and this can hurt the application quality. Continuous aggressive charge and discharge might also severely affect the health of storage components and hence shorten their lifetime. Without a proper energy management scheme, the system cannot deliver sustainable operation for a long period of time which contradicts with the objective of providing energy sustainability in systems.

While adapting application quality to adjust energy consumption at middleware has been a promising solution for energy minimization and energy harvesting management, we focus on quality-aware energy harvesting management framework in concert with hybrid energy storage management. Many existing work at middleware consider the energy storage as a buffer with maximum capacity [8][9][6]. However, the power density constraint, the storage lifetime constraint, and energy efficiency in the hybrid energy storage architecture cannot be neglected. To the best of our knowledge, this is the first effort to bring hybrid-energy-storage-awareness into application quality adaptation at middleware. We propose a holistic middleware framework to orchestrate the application quality management and hybrid energy storage management for embedded systems such as multi-sensor platforms with solar energy harvesting capability.

In this paper, we present: 1) an abstract model for hybrid energy storage with battery and supercapacitor to capture the characteristics of each storage type during charge and discharge, including battery lifetime and supercapacitor leakage, 2) an offline slot-based energy budget management to adjust application quality (*QoS*) according to energy availability and storage status, providing energy charge/discharge planning for each storage type with the objective of maximizing the application *QoS* under a given energy storage lifetime constraint, and 3) a semi-online adaptation phase captures the inaccuracy during offline approximation and prediction and dynamically adapts the charging and discharging distribution as well as the application *QoS*. Lastly, a controller directs charge and discharge at runtime.

In comparison with quality-aware frameworks [6][9] which do not consider the storage power density and storage lifetime, our experimental results show that such simple model can make the system fall short in realizing the expected application QoS for a long system lifetime. Our approach emphasizes that considering battery *SoH* (State-of-Health) and leakage of supercapacitor during offline planning along with an online scheme can help systems achieve high application QoS while reliably meeting battery lifetime constraint. In particular, this energy management framework effectively extends battery lifetime *1.6x-2x* compared to [6][9].

## 2. Related work

There are several architectures proposed for energy harvesting systems with hybrid energy storage. In Chulsung and Chou [4] and Carli et. al [5], hybrid energy storage management is implemented in hardware using simple voltage thresholds (threshold to charge/discharge battery and supercapacitor) to avoid energy overflow and underflow. These methods are neither aware of energy efficiency nor battery lifetime. Hybrid energy storage systems for automotive or mobile system have been proposed in [1][7]. In [1], their approach increases the battery lifetime by reducing battery charge average and standard deviation, and using supercapacitor as a buffer to smooth out charging and discharging current from the battery. This system architecture assumes a charge transfer interconnect between charging source (or load) and hybrid energy storage subsystem. It allows charging or discharging to/from both supercapacitor and battery at the same time However, their system avoids charging and discharging simultaneously which often happens in energy harvesting systems. [16] is an improved extension of [1]. In [7], the authors focus on the rated capacity effect (energy efficiency) of the battery. This work statically schedules discharging activities of a hybrid energy storage system and does not consider recharging activities or lifetime of batteries. Both [7] and [16] do not consider load adaptation. In our work, we dynamically adjust workload in orchestration with hybrid energy storage management to maximize application *QoS* while meeting storage lifetime and harvesting constraints.

Furthermore, [14] considers harvesting system with only rechargeable battery. They utilize Phase Change Memory to do load matching in energy harvesting WSNs, reducing number of charging/discharging cycles to battery. Their goal however is not to optimize application QoS. Previous works on energy budgeting for harvesting sensor networks [9][6] optimize application QoS or consumption of harvested energy. They are not aware of hybrid energy storage characteristics, neither power density constraint nor battery lifetime constraint. [6] assumes ideal storage and therefore overestimates energy availability and aggressively optimizes application QoS. As a result, it can cause significant battery lifetime degradation and shorten useful lifetime of the whole system.

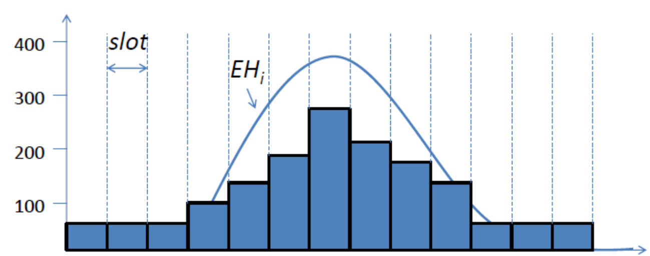


**Figure 2:** Slot-based energy harvesting and budget allocation example

## 3. Our proposed framework

In this section, we present an overview of our proposed framework. The framework is for operation during a hyper-period of harvesting and application activities. For solar-powered embedded systems, we chose a day long hyper-period. Each period is then divided into $n$ equal slots, each slot of duration $\Delta T$. We leverage existing harvesting prediction algorithm [12] to gain knowledge about future energy harvesting in each slot. Figure 2 is an example of slot-based energy harvesting prediction and budget allocation. The optimization problem addressed in this multi-phase framework is formally stated as:

1) Given a set of QoS levels for each application, the hybrid energy storage characteristics, a target battery lifetime $T_{life}$, and predicted energy harvesting for each time slot,
2) Assign QoS level for each slot as well as charge/discharge distribution for energy storage components such that
3) The total QoS of the system is maximized.

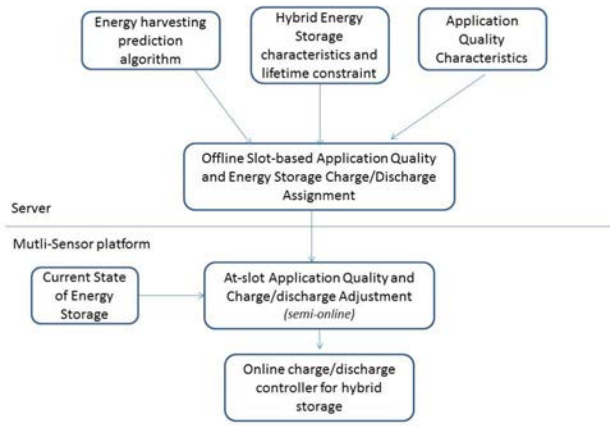


**Figure 3:** Our Proposed Middleware Framework for Energy-Storage-aware Application Quality Adaptation

Our framework structure is shown in Figure 3. Using prediction algorithm to estimate energy harvesting profile (i.e., solar) and prior knowledge on event and application activities, an offline phase is proposed to allocate charge/discharge budget to each energy storage component while maximizing the total QoS in the next hyper-period. Our distinct contribution is that the proposed offline algorithm is not only a QoS-aware energy budget allocation but also takes into consideration the hybrid energy storage characteristics. It considers battery lifetime constraint, the leakage in supercapacitor, and efficiency of chargers and converters. Increasing the energy efficiency of the storage leads to maximizing energy availability and hence, maximizing QoS while the target system lifetime is guaranteed by considering battery status and lifetime constraint. Given the high complexity of this problem, the offline phase is driven by a server and the results are transferred to each node before the next hyper-period.

These limits and bounds from the offline optimization are guidance to an online energy storage charge and discharge controller. In addition, given that energy harvesting profile is more accurately predicted with short-term prediction (i.e., extrapolating next slot energy harvesting from harvesting data of previous $k$ slots [17]), a semi-online adaptation algorithm is deployed to correct error from offline prediction. The state of hybrid energy storage at the current time slot may not be the same as the expected value due to lack or surplus of energy in the past slots. Therefore, the semi-online algorithm provides incremental updates on the expected QoS and charge/discharge distribution for the current slot. The proposed algorithm is lightweight and can be deployed at the middleware layer of the node.

## 4. Target system and application model

This section provides an overview of application quality model and hybrid energy storage model deployed in this work.

### 4.1 Application Quality Model

We assume each sensing application $j$ has a multi-level quantized QoS model. Each level $k$ is characterized by a configuration of application and/or system parameters represented as a tuple $<c_{jk1}, c_{jk2}, .., c_{jkm}>$ where $m$ is the number of system and application parameters, and $c_{jkx}$ is configuration value for parameter $x$ at level $k$ of application $j$. For example, data rate and error margin are configurable application parameters, while frame rate and resolution are configurable system parameters whose different values determine various QoS levels. The corresponding energy consumption $EC_{jk}$ for each QoS level $k$ of application $j$ is determined through multiple measurement experiments, taking the average out of those measurements.

### 4.2 Hybrid energy storage model

Figure 4 shows the hybrid energy storage architecture. We denote the output from the harvesting circuit as $V_{src}$ and $I_{src}$. $\gamma_{battery}^{regulator}$ and $\gamma_{supercap}^{regulator}$ are the efficiency of the regulators controlling the charging process to the battery and supercapacitor, respectively. On the other side, the voltage and current required to power the load is denoted as $V_{load}$ and $I_{load}$. The efficiency of the converters matching voltages of battery and supercapacitor with the required load voltage are $\gamma_{battery}^{converter}$ and $\gamma_{supercap}^{converter}$. State of Charge (*SoC*) is the percentage of charge remained in the battery, compared to the full rated capacity. *SoC* is updated based on charging and discharging current according to Columbus counting method.

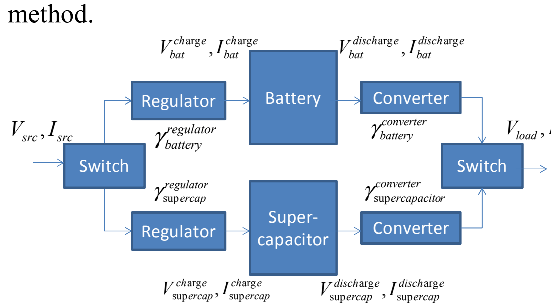


**Figure 4:** Hybrid Energy Storage Subsystems

Battery lifetime model: Battery has a limited number of charging and discharging cycles, i.e. limited lifetime. To estimate battery lifetime, we adopt the model in [3][1]. This model allows us to use *SoC* average and standard deviation to estimate the State of Health (*SoH*) degradation. *SoH* degradation is a way to measure the ability of battery to store and deliver energy. This degradation gradually increases from *0* to *1* during battery lifetime, with *0* as a fresh battery and *1* as a battery without capacity. Conventionally, a battery with a degradation of *0.2* is considered failed. *SoH* degradation of battery in a hyper-period *P* is computed as:

$$D(P) = (C_1 N e^{\alpha SoC_{dev}} + C_2) C_3 e^{\beta SoC_{avg}} (1 - D^s) \quad (1)$$

where $C_1$, $C_2$, $C_3$, $\alpha$, and $\beta$ are empirical constants specific to each battery. $N$ is the number of effective charging/discharging cycles in a hyper-period $P$. $D^s$ is the *SoH* degradation at the start of this hyper-period. The *SoH* degradation over multiple hyper-periods is the sum of degradation over each individual hyper-period:

$$D(kP) = \sum_{i:1 \to k} D(P_i) \quad (2)$$

To meet the target lifetime $T_{life}$, assuming equal degradation in each period, equation below shows the constraints for $SoH$ degradation in each hyper-period.

$$D(P) \leq \frac{0.2P}{T_{life}} \quad (3)$$

All three parameters, $N$, $SoC_{dev}$ and $SoC_{avg}$ in Equation (1) are dynamically adjusted in our proposed framework to meet the battery lifetime constraint in Equation (3).

Supercapacitor leakage model: For an accurate estimation of supercapacitor leakage energy, there are models using complex circuits with intensive calculations. We use the approximation model for supercapacitor leakage in [7] for low complexity and ease of integration.

$$P_{leakage} = \mu e^{\rho V_{sup}} \quad (4)$$

where $V_{sup}$ is the voltage of the supercapacitor and μ and $\rho$ are empirical parameters of the supercapacitor.

## 5. Algorithms for application quality and energy charge/discharge assignment

In this section, we first describe our offline optimal solution for slot-based QoS assignment and storage charge/discharge distribution. We then propose semi-online algorithm for QoS re-adjustment at the beginning of each slot and an online charge and discharge controller.

### 5.1 Offline Planning

We propose an optimal solution using ILP solvers. The objective is to assign each time slot $i$ an expected application QoS and charge/discharge distribution. We summarize here the various set of constraints:

Capacity and Power Density Constraints: Let $ES_i$ and $EB_i$ denote the energy in the supercapacitor and the battery at the beginning of time slot $i$. To avoid overflow and underflow, $ES_{max}$ and $EB_{max}$ are the upper bounds; $ES_{min}$ and $EB_{min}$ are the lower bounds of $ES_i$ and $EB_i$, respectively.

$$ES_{min} \leq ES_i \leq ES_{max} \quad (5)$$

$$EB_{min} \leq EB_i \leq EB_{max} \quad (6)$$

Similarly, the charging and discharging energy budget of the battery and supercapacitor in each slot $EB_i^{charge}$, $EB_i^{discharge}$, $ES_i^{charge}$, $ES_i^{discharge}$ are subject to minimum and maximum values (power density).

$$EB_{min}^{charge} \leq EB_i^{charge} \leq EB_{max}^{charge} \quad (7)$$

$$EB_{min}^{discharge} \leq EB_i^{discharge} \leq EB_{max}^{discharge} \quad (8)$$

$$ES_{min}^{charge} \leq ES_i^{charge} \leq ES_{max}^{charge} \quad (9)$$

$$ES_{min}^{discharge} \leq ES_i^{discharge} \leq ES_{max}^{discharge} \quad (10)$$

Remaining Charge Update: Energy in battery and supercapacitor at the beginning of slot $i+1$ is updated based on its charge, discharge and leakage in slot $i$.

$$EB_{i+1} = EB_i + \gamma_{battery}^{regulaor} EB_i^{charge} - EB_i^{discharge} \quad (11)$$

$$ES_{i+1} = ES_i + \gamma_{supercap}^{regulaor} ES_i^{charge} - ES_i^{discharge} - ES_i^{leakage} \quad (12)$$

Where $ES_i^{leakage}$ is the leakage energy from the supercapacitor. We use piecewise linear approximation to estimate leakage in Equation (4).

Energy Harvesting Constraints: The total energy charging to battery and supercapacitor in each slot, i.e., $EB_i^{charge}$ and $ES_i^{charge}$ is limited by the predicted energy harvesting $EH_i$.

$$EH_i \geq EB_i^{charge} + ES_i^{charge} \quad (13)$$

Application Quality constraints: For each slot $i$, an application $QoS_{i,j}$ is assigned for application $j$ and it must meet the minimum QoS requirement. The total energy consumption for all applications in each slot comes from the power supply, $EB_i^{discharge}$ and $ES_i^{discharge}$.

$$\sum_{\forall j} EC_{jQoS_{i,j}} = \gamma_{battery}^{converter} EB_i^{discharge} + \gamma_{supercap}^{converter} ES_i^{discharge} \quad (14)$$

Battery Lifetime Constraints: Charging and discharging currents from battery, $I_{bat}^{charge}$ and $I_{bat}^{discharge}$ are functions of battery $SoC$ and the charging/discharging power.

$$I_{bat}^{charge} = f(SoC, P^{charge}) = \frac{-VOC(SoC) + \sqrt{VOC(SoC)^2 + 4R \times P^{charge}}}{2R} \quad (15)$$

where VOC is a function of SOC (see [3][1]) and $R$ is a empirical constant of the battery. Similar function is for $I_{bat}^{discharge}$ and $P^{discharge}$. Since this is a non linear function, we use Taylor approximation to approximate charging and discharging currents from battery in each slot $i$, choosing $SoC_{pivot} = 0.5$ and $P_{pivot}^{charge} = EB_{max}^{charge}/(2\Delta T)$ as the pivot points.

$$I_{bat,i}^{charge} = f\left(SoC_{pivot}, P_{pivot}^{charge}\right) + \left(SoC_i - SoC_{pivot}\right) \times f_{SoC}\left(SoC_{pivot}, P_{pivot}^{charge}\right) + \left(P_i^{charge} - P_{pivot}^{charge}\right) \times f_{P^{charge}}\left(SoC_{pivot}, P_{pivot}^{charge}\right) \quad (16)$$

Where $P_i^{charge} = EB_i^{charge}/\Delta T$

The charging and discharging currents are used to compute effective cycles $N$ and battery $SoC_{dev}$ and $SoC_{avg}$ as shown in the following equations.

$$N_{i+1} = N_i + \left(I_{bat}^{charge} + I_{bat}^{discharge}\right) \times \Delta T/(2Q_{bat}) \quad (17)$$

$$SoC_{i+1} = SoC_i + (I_{bat}^{charge} - I_{bat}^{discharge}) \times \Delta T/Q_{bat} \quad (18)$$

$$SoC_{avg} = \frac{1}{n} \sum_{i:0..n-1} SoC_i \quad (19)$$

$$SoC_{dev} = \frac{2\sqrt{3}}{n} \sum_{i:0..n-1} \left|SoC_i - SoC_{avg}\right| \quad (20)$$

$N_i$ is the accumulated effective cycles in the first $i$ slots while $SoC_i$ is the battery $SoC$ at the beginning of slot $i$. The battery lifetime degradation in each hyper-period is estimated from the values of $N$, $SoC_{dev}$ and $SoC_{avg}$ according to constraint (1). Substitute the equation (1) into (3) we have this battery lifetime constraint:

$$(C_1 N_n e^{\alpha SoC_{dev}} + C_2) \times C_3 e^{\beta SoC_{avg}} \times (1 - D^S) \leq \frac{0.2P}{T} \quad (21)$$

This constraint is also nonlinear, we simply it as followed. When $N_n \geq 1.0$, $C_2$ is much smaller than other components, hence we safely ignore it. Equation (21) is equivalent to:

$$\log C_1 + \lg N_n + \alpha SoC_{dev} + \log C_3 + \beta SoC_{avg} \leq \log \frac{0.2P}{T_{life}(1 - D^s)} \quad (22)$$

When $0.1 \leq N_n \leq 1$, $C_2$ is estimated as $C_2 = \frac{1}{2} C_1 N_n e^{\alpha SoC_{dev}}$. Therefore, Equation (21) is equivalent to:

$$\log 1.5 + \log C_1 + \log N_n + \alpha SoC_{dev} + \log C_3 + \beta SoC_{avg} \leq \log \frac{0.2P}{T_{life}(1 - D^s)} \quad (23)$$

Objective Function: Given these energy harvesting constraints, hybrid energy storage capacity, battery lifetime constraints, and application quality constraints, the objective of the ILP is to maximize overall application QoS:

$$\textbf{Maximize} \sum_{i,j} Q_{i,j}$$

## 5.2 Semi-online Adjustment

The predicted energy harvesting can be different from actual values due to uncertainty and change in the weather condition and workload. Battery *SoC* could also be different from offline prediction because of approximation error. Hence, at the beginning of each time slot, we propose an efficient algorithm to adjust the QoS and charge/discharge distribution in energy storage. The idea is to keep (or to reduce) battery *SoC* as close as possible to the offline plan to meet the lifetime constraint. Extra harvesting energy if any is routed through the supercapacitor to improve QoS.

At run time, we keep track of variables related to battery lifetime including effective cycles *N* and *SoC*. The deviation from offline approximation is computed in each slot as follow:

$$\Delta N = N_i^{online} - N_i^{offline} \quad (24)$$

$$\Delta SoC = \int_0^{i\Delta T} SoC(t) - \sum_{j:0..i-1} SoC_j^{offline} \times \Delta T$$

This deviation can lead to battery lifetime constraint violation; hence it must be kept track and adjusted to stay within a safe bound. Charging and discharging activities of the battery directly affect battery's *N* and *SoC*. This impact is reflected in the observation below.

Increase in charge to the battery leads to increase in *N* and *SoC* and vice versa. Increase in discharge from the battery, however, leads to increase in *N* but decrease in *SoC* and vice versa. Therefore, to keep battery's *ΔN* and *ΔSoC* within an accepted bound, adjustment to charging and discharging from the battery is important.

The short-term adaptation algorithm (Figure 5) first detects the cases in which battery's *ΔN* and *ΔSoC* is beyond a safe bound and in a long run, it can lead to battery lifetime constraint violation. In these cases, the system has used or has stored more energy in the battery than the offline planning. Applying the observation above, the algorithm computes the corresponding adjustment in charging/discharging currents needed to bring battery's *N* and *SoC* to be within bound (line *1-5*). To achieve this target reduction, the charging current to the battery is first reduced and the charge budget for battery in slot *i* is updated (line *6-8*). If the target reduction in current is not met, the discharging current from the battery is adjusted next. From the observation, the discharging current has opposite effects on battery's *N* and *SoC*. If effective cycles *N* has to reduce, the algorithm reduces the discharge current from the battery. If battery *SoC* has to reduce, the algorithm increases the discharge current from the battery instead. The budget for battery charge and discharge is updated (line *10-16*).

**Short-term Adaptation: at beginning of slot i**
**Input:** $\Delta \boldsymbol{N}, \Delta \boldsymbol{SoC}$
**Output:** Updated $QoS_i$
1. **if** ($\Delta N > \varepsilon N$)
2. $\quad \Delta I = 2Q_{bat} \times \Delta N / \Delta T$
3. **elseif** ($\Delta SoC > \varepsilon SoC$)
4. $\quad \Delta I = 2Q_{bat} \times \Delta SoC / \Delta T^2$
5. **endif**
6. **if** ($\Delta I < I_{bat,i}^{charge}$)
7. Reduce $I_{bat,i}^{charge}$ by $\Delta I$
8. Update $EB_i^{charge}$
9. **else**
10. Reduce $\Delta I$ by $I_{bat,i}^{charge}$ and $I_{bat,i}^{charge} = 0$
11. **if** ($\Delta N > \varepsilon N$)
12. Reduce $I_{bat,i}^{discharge}$ by $\Delta I$
13. **elseif** ($\Delta SoC > \varepsilon SoC$ )
14. Increase $I_{bat,i}^{discharge}$ by $\Delta I$
15. **endif**
16. Update $EB_i^{charge}$ and $EB_i^{discharge}$
17. **endif**
18. $\Delta EH = EH_i - EB_i^{charge} - ES_i^{charge}$
19. **if** ($\Delta EH < 0$)
20. Reduce $EB_i^{charge}$ and $ES_i^{charge}$
21. **else**
22. Increase $ES_i^{charge}$
23. Increase $EB_i^{charge}$ if $\Delta N < 0$ and $\Delta SoC < 0$
24. **endif**
25. Update $EB_i^{discharge}$e and $ES_i^{discharge}$
26. Update QoS = lookup($EB_i^{discharge} + ES_i^{discharge}$)

**Figure 5:** Short-term adaptation algorithm

Once the adjustment to meet battery lifetime constraint is carried out, the algorithm checks the harvesting condition (line *18*). Energy harvesting for slot *i* is updated using short-term prediction algorithm [17]. If updated energy harvesting is less than the planned charge budget to battery and supercapacitor, the algorithm reduces charge budget to battery which in turn helps to reduce both battery *N* and *SoC* (line *19-20*). If energy harvesting is more than what was expected, the charge budget to the supercapacitor is increased up to its limit. In addition, if $\Delta N < 0$ and $\Delta SoC < 0$, there are some slack cycles to charge more energy to the battery (line *22-23*). The discharge budget from supercapacitor/battery has a corresponding adjustment (line *25*). Finally we update application QoS according to the new discharge budget form supercapacitor and battery (line *26*).

The proposed algorithm is an efficient algorithm that opportunistically harvests, stores and improves application QoS under battery lifetime constraint and dynamic changes in energy harvesting profile.

## 5.3 Online Charge/Discharge Controller

At run time, given the harvesting power, the controller decides whether to charge the supercapacitor or the battery. The decision is based on the guideline given by the offline and semi-online phases. The controller charges the supercapacitor first up to its charge budget, then it switches to the battery. However, it discharges from the battery first up to its discharge budget then switches to the supercapacitor. Delaying charging to the battery and expediting discharging from the battery help to reduce the battery $SoC_{avg}$ while keeping effective cycles $N$ the same.

# 6. Experiments

In this section, we first explain the experimental setup and then evaluate our proposed energy management middleware for hybrid energy storage harvesting systems. We evaluated our approach under various weather conditions for a period of seven days in two typical seasons, summer and winter. We compare with existing work on adapting data quality and hybrid energy storage management and show that it is necessary to consider application QoS optimization and system lifetime constraint in an orchestrated manner in both offline planning and online phase of an energy harvesting system.

## 6.1 Experimental Setup

We implemented our hybrid energy storage management in a simulated harvesting system consisting of a *220mAhr* Li-on battery and a *50F* supercapacitor. The simulation is conducted in state-of-the-art commercial networked system simulator, Qualnet [10]. A network of wireless sensor nodes is simulated; each node has a temperature sensor, a humidity sensor, an acoustic sensor and a low-power image sensor.

We model the physical phenomenon such as temperature as a random process between a lower and upper bound, with a random variation in each step. Query model includes both periodic queries and Poisson-distribution sporadic queries. Specific settings are provided in our previous work [6]. Through extensive simulation, we quantify different levels of QoS, each with an error margin and energy consumption (see Table 1). Energy consumption is average value from multiple simulation runs.

**Table 1:** Quantized QoS levels for temperature sensor

| QoS level | Error margin | Energy consumption per slot (mJ) |
|---|---|---|
| 1 | 7.0 | 37901.79 |
| 2 | 6.0 | 37950.02 |
| 3 | 5.0 | 38057.57 |
| 4 | 4.0 | 38257.28 |
| 5 | 3.0 | 38717.11 |
| 6 | 2.0 | 39876.28 |
| 7 | 1.0 | 44342.70 |
| 8 | 0.0 | 51122.13 |

The low-power image sensor has a QoS model defined by resolution and frame rate. [13] shows that different resolution and frame rate changes the video quality perceived by users. We select *3* modes and corresponding energy consumption shown in Table 2 for a lower power image sensor (Aptina [11]). Assuming power consumption per pixel is constant, corresponding power for QoS level *1-3* is *80mW, 106mW* and *120mW* respectively. Assume it takes *2ms* to capture an image and the system is idle until the next capture.

**Table 2:** QoS levels for low-power camera

| QoS level | Configuration | Energy consumption per slot (mJ) |
|---|---|---|
| 1 | 15 fps, 640x480 | 4298 |
| 2 | 60 fps, 352x288 | 22447 |
| 3 | 90 fps, 320x240 | 39272 |

We test our approach with solar profiles including a winter week (February) and a summer week (August) at Elizabeth city, North Carolina. Data is retrieved from National Renewable Lab website [15]. The solar irradiance is converted to harvested energy by linear conversion considering solar panel size of $30.3\ cm \times 20.2\ cm$, solar cell efficiency of *10%* and harvesting efficiency of *80%*. The winter days in this data set consistently has lower solar irradiation and thus, lower energy harvesting than summer days. During the summer days, although the irradiation is higher, we also observe significant fluctuation within each day and very low harvesting on a rainy day.

We integrate a long-term solar energy prediction algorithm which gives slot-based harvesting prediction of the next day based on the past *3* days [12]. The inaccuracy of this per-slot prediction algorithm goes up to *40%* in our experiments. The short-term energy prediction for the next slot based on previous *3* slots is adopted from [17].

## 6.1 Experimental Results

To show the importance of orchestrating application QoS and hybrid storage management, we compare our work with two baseline approaches coupling with other existing techniques to enhance them.

- Battery-aware approach: The offline planning is only aware of energy storage capacity, similar to other work such as [6][9][8]. In addition, we adopt the charging and discharging controller in [4] to switch between battery and supercapacitor based on their voltages.
- Online-SoH approach: Inspired by the battery *SoC* model in 0[3], we incorporate an online *SoH*-aware heuristic. The heuristic keeps track of accumulated cycles $N$, $SoC_{avg}$ and $SoC_{dev}$ at run time and allows charging to the battery only if projected battery lifetime degradation based on current battery status meets the lifetime constraint.

The target lifetime in our experiments is *3* years. The metrics we use are application QoS (normalized to the maximum possible QoS in a day) and *SoH* degradation.

Figure 6 shows the results comparing application QoS and daily battery lifetime degradation of our approach with two variations above for a week in winter. Since the goal of the other two baselines is to optimize application QoS without being constrained by the battery lifetime, it is not a surprise that they have better QoS than our approach. On average, Battery-aware approach has *15%* higher QoS and Online-SoH has *14%* higher QoS than our approach. However, as shown in Figure 6, our approach has significant less daily *SoH* degradation. On average, our daily *SoH* degradation is *50%* lower compared to Battery-aware approach and *48%* lower compared to Online-SoH approach. The Online-SoH approach tries to reduce the daily battery lifetime degradation at runtime. However, because of the complicated correlation between *SoH* parameters

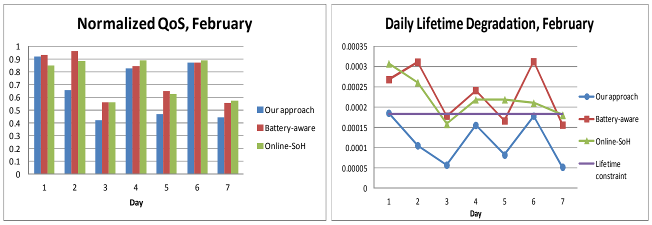


**Figure 6:** Winter QoS and Daily SoH/Lifetime Degradation

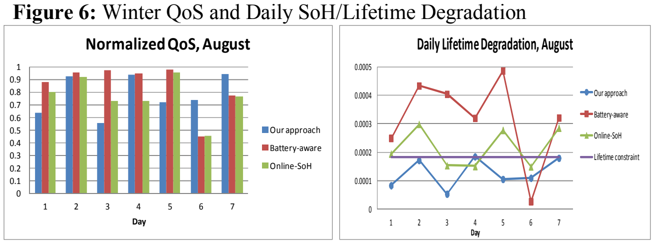


**Figure 7:** Summer QoS and Daily SoH/Lifetime Degradation

including effective cycles *N*, average *SoC* and *SoC* deviation, online heuristics are not sufficient to guarantee battery lifetime constraints. Both Battery-aware and Online-SoH approaches often miss battery lifetime constraint as shown in the Figure 6. On the other hand, our approach always meets the battery lifetime constraint without degrading application QoS. This reinforces our approach of orchestrating application QoS and hybrid storage management in both offline and online phase.

Figure 7 shows the comparison results for a week in summer. On average, Battery-aware approach has only *7%* higher QoS compared to our approach while Online-SoH is *1%* lower in QoS than our approach. This partly comes from the sixth day of the summer week where harvesting profile is significant low while energy harvesting prediction is high. Battery-aware and Online-SoH aggressively optimize QoS and consume large amount of energy consumption till the short-term adaption realizes the shortage in harvested energy compared to prediction. Their application QoS is dropped significantly after this and it affects the next day when energy reserve at the beginning of the seventh day is low. In terms of daily battery *SoH* degradation, our approach consistently has lower degradation compared to the other two. On average, our approach has 50% less degradation than Battery-aware approach and *38%* less degradation than Online-SoH approach.

If this trend in the summer and winter continues, our approach is expected to have *2x* longer lifetime compared to Battery-aware approach and *1.6x-1.9x* longer lifetime compared to Online-SoH approach. Our approach always meets the daily *SoH* degradation constraint and is very likely to have useful lifetime of more than *3* years. On the other hand, the average daily *SoH* degradation of Battery-aware and Online-SoH approach suggests that their system can only sustain *1.7-2.5* years.

## 7. Conclusion

In this paper, we propose a two-phase hybrid energy storage management and application quality adaptation for harvesting capable devices. The offline phase plans energy charge and discharge based on prediction of energy harvesting profile in order to maximize application QoS while meeting energy storage lifetime constraint. The online phase implements a heuristic to control and adapt charging, discharging activities to the hybrid energy storage. Our experiments prove that offline planning and online adaptation tightly coupled with knowledge of hybrid energy storage is crucial to realize application quality optimization on a sustainable harvesting device.

## 8. References


[1] X. Qing et. al, "State of health aware charge management in hybrid electrical energy storage systems." In *DATE 2012*.

[2] M. Pedram, N. Chang, Y. Kim, and Y. Wang, “Hybrid Electrical Energy Storage Systems”, in ISLPED 2010

[3] A. Millner, “Modeling Lithium Ion Battery Degradation in Electric Vehicles”, in CITRES 2010.

[4] P. Chulsung, P. H. Chou. "Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes." In SECON 2006.

[5] D. Carli, D. Brunelli, L. Benini, M. Ruggeri, “An effective multi-source energy harvester for low power applications”, in DATE 2011

[6] N. Dang, E. Bozorgzadeh, N. Venkatasubramanian “QuARES: Quality-aware data collection in energy harvesting sensor networks”, in IGCC 2011

[7] A. Mirhoseini, F. Koushanfar, “HypoEnergy: Hybrid supercapacitor-battery power-supply optimization for Energy efficiency”, in DATE 2011

[8] D. K. Noh, L. Wang, Y. Yang, H. K. Le, and T. Abdelzaher, “Minimum Variance Energy Allocation for a Solar-Powered Sensor System”, in DCOSS 2009

[9] C. Moser, J.-J. Chen, and L. Thiele, “Power management in energy harvesting embedded systems with discrete service levels”, in ISLPED’09

[10] Qualnet, http://web.scalable-networks.com/content/qualnet

[11] Aptina MT9V11, http://www.aptina.com

[12] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, V. Raghunathan, “Adaptive Duty Cycling for Energy Harvesting Systems,” in ISLPED 2006

[13] S. Mohapatra et. al. “Integrated power management for video streaming to mobile handheld devices”. In ACM Multimedia 2003

[14] Ping Zhout, Youtao Zhang, Jun Yang, “The Design of Sustainable Wireless Sensor Network Node using Solar Energy and Phase Change Memory”, in DATE, 2013

[15] National Renewable Lab, http://www.nrel.gov/midc/

[16] Y. Wang et. al, “Minimizing State-of-Health Degradation in Hybrid Electrical Energy Storage Systems with Arbitrary Source and Load Profiles”, in DATE 2014.

[17] J. Recas Piorno, Carlo Bergonzini, David Atienza, and T. Simunic Rosing. "Prediction and management in energy harvested wireless sensor nodes." In Wireless VITAE 2009.