Abstract—The main challenge with microsensors is limited space, because tiny batteries store little energy. Harvesting energy helps, but only when ambient energy is available. And even then, power is low because miniaturized transducers harness little power. This is why managing how and when to schedule functional tasks is so important. This paper proposes a schedule that requires the battery to hold only the energy necessary to sustain single events and allows the battery to drain across harvesting droughts. When input power returns, an inefficient starter charges a temporary supply that bootstraps the system quickly into an efficient state. This way, when harvested input, idle, and peak load power are 10, 0.5, and 1000 μW, a 0.7% efficient starter and an 87% efficient charger can recharge a 1.8-pF battery to 1 V in 220 ms with a 71-pF temporary supply and in 13 s without the temporary supply. The system therefore wakes 59× faster than without a temporary supply and 1880× faster and with 1800× less capacitance than when forcing the battery to survive a 2-hour power outage.

Keywords—Energy-drought recovery, energy harvesting, low-energy task schedule, microsensor, microsystem, startup, wake.

I. ENERGY-HARVESTING AND DROUGHT-RECOVERING WIRELESS MICROSYSTEMS

Wireless microsensors can add life-, cost-, and energy-saving intelligence to, among others, homes, hospitals, biological systems, vehicles, factories, and farms [1]–[5]. Key to their ubiquity is miniaturization because space in modern and emerging applications is increasingly scarce. Tiny onboard batteries, however, store little energy, and in the case of super capacitors, also lack considerable power [6]. So harnessing ambient energy is often a requirement for these microsystems.

Still, ambient energy $E_A$ is not always available [7]–[8]. And when available, miniaturized transducers might only avail 1 of the 1000 μW/mm² that the highest power-producing devices can [3], [5]. So with $E_A$, the harvesting source $v_H$ in Fig. 1 feeds a maximum power-point (MPP) charger that replenishes a battery $C_B$ with enough energy to supply the system. The power supply [9]–[10] then draws and conditions power to satisfy the sensor, digital-signal processor (DSP), power amplifier (PA), and other system components. The central processor (CP) assesses the state of the system to determine which blocks to activate and which to disable.

![Fig. 1. Energy-harvesting wireless microsensor system.](image)

Since $v_H$ does not always output power $P_H$, $C_B$ must be large enough to store the energy that the system requires across harvesting droughts. Or if allowed to drain, which is more practical because drought periods can be long and super capacitors leak, the system should recover fast enough when $E_A$ returns to leverage what $P_H$ avails before $E_A$ again fades.

Critical design considerations here are size and wake time. But for a small $C_B$, load power should be low. This is why Section II proposes a low-energy “just-enough” task schedule. A small $C_B$, however, drains across harvesting droughts. So Sections III and IV propose, analyze, and assess how the system should enter and recover from power outages.

II. LOW-ENERGY "JUST-ENOUGH" TASK SCHEDULE

To minimize how much energy the battery $C_B$ supplies, the system should schedule no more than one task at a time. And a task should only occur when $C_B$ has enough energy to sustain it. In a microsensor, sensing and processing what is being sensed constitute one such task and transmitting processed data another. Of these, transmissions usually require more energy with $E_T$ than sensing events do with $E_S$. So depending on the application, a system can sense and process more data by sensing $N$ times before every transmission.

For this, the system can monitor $C_B$’s voltage $v_B$ as a way of determining when $C_B$ stores enough energy to sustain a task. And as soon as $C_B$ collects sufficient energy, the system executes a task. So when the harvester charges $C_B$ to sensing threshold $V_{S(TH)}$, like Fig. 2 indicates and Fig. 3 shows at 180 ms, the system senses, processes, and stores data. The sensor and microelectronics involved in this process discharge $C_B$ below $V_{S(TH)}$. And when $v_B$ again reaches $V_{S(TH)}$ at 200 ms, the system senses again and the sequence repeats (and loops in Fig. 2) the $N$ times that the application requires.

![Fig. 2. Proposed low-energy "just-enough" task schedule flow chart.](image)

After sensing $N$ times, the system waits until $v_B$ reaches transmission threshold voltage $V_{T(TH)}$. Since transmissions are usually the most power-consuming task in the system, $V_{T(TH)}$ corresponds to the highest energy level that $C_B$ should hold. The smallest $C_B$ that will hold this energy will do so at the
highest voltage possible, which corresponds to the breakdown voltage $V_{BD}$ of the circuit. So when $v_B$ reaches $V_{BD}$, which happens at 350 ms in Fig. 3, the PA transmits the information collected across N sensing events.

![Image 3](image3.png)

**Fig. 3.** Power train and corresponding battery-voltage simulation.

Each event should discharge $C_B$ to no less than the energy and corresponding voltage required $V_{MIN}$ to sustain offline tasks. This is why $v_B$ drops to $V_{MIN}$ after every transmission and sensing event in Fig. 3. By waiting for $v_B$ to reach sensing and transmission thresholds $V_{S(TH)}$ and $V_{T(TH)}$, the system automatically adjusts the time $t_o$ between transmissions to keep average load power $P_{LD(AVG)}$ and power losses $P_{LOSS}$ at the level the harvesting source $v_H$ supplies with $P_H$:

$$P_H = P_{LD(AVG)} + P_{LOSS}. \quad (1)$$

The application and the state of the art of supporting technologies dictate the thresholds and values required to operate the system. CMOS circuits, for example, can sustain up to 1.8–4.5 V and consume 11 nW when idling, 2–50 µJ when sensing, and 38 nJ–58 µJ when transmitting [1]–[2], [4]–[5], as Table I shows. Chargers can deliver 0.15%–0.7% of the power drawn from a 40–500 mV source during startup and 87% from higher voltages in steady state [11]–[13]. Although startup does not affect the time $t_o$ requires to charge when enough ambient energy is present (in Fig. 3), low startup efficiency extends the time needed to recover from harvesting droughts.

<table>
<thead>
<tr>
<th>TABLE I. POWER LEVELS IN THE STATE OF THE ART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Harvesting Source</td>
</tr>
<tr>
<td>Harvesting Source Power</td>
</tr>
<tr>
<td>Idle Power</td>
</tr>
<tr>
<td>Sense Energy</td>
</tr>
<tr>
<td>Transmission Energy</td>
</tr>
<tr>
<td>Breakdown Voltage</td>
</tr>
<tr>
<td>Startup Efficiency</td>
</tr>
<tr>
<td>Steady-State Efficiency</td>
</tr>
</tbody>
</table>

III. DOWNTIME OPERATION, ANALYSIS, AND DESIGN

The charger in Fig. 1, like most harvesting chargers [13], monitors the power it delivers with $P_{CHG}$ to adjust and ensure it operates at its maximum power point (MPP). When this $P_{CHG}$ falls below the minimum operating threshold $P_{MIN}$ in Fig. 4, the PA transmits a system-offline report. The central processor (CP) then disables the power supply that feeds the sensor, DSP, PA, and other blocks until $P_{CHG}$ climbs back to $P_{TH}$. But if $P_{CHG}$ falls further to zero, $v_H$’s harvesting power $P_H$ is no longer able to sustain charger losses. So CP shuts the charger to enter standby or sleep mode the charger can draw and output power.

![Diagram](image4.png)

**Fig. 4.** Proposed downtime flow chart.

**A. Standby**

When designed for standby, the battery $C_B$ is large enough to keep $v_B$ from ever falling below the headroom level $V_{HR}$ that the charger requires to operate. This way, the charger can recover efficiently from a harvesting drought. Although almost nothing is on during the outage, low-power survival blocks in CP still operate, so $C_B$ nevertheless discharges, albeit slowly. Self-discharge in $C_B$ accelerates this process. So when the PA sends an offline transmission at 0.25 s in Fig. 5, the PA discharges $C_B$ quickly and survival blocks continue to discharge $C_B$ after that until the drought ends at 7200 s.

![Diagram](image5.png)

**Fig. 5.** Standby simulation.

Just before sending an offline report at 0.25 s, $C_B$ should store enough $0.5C_B^2V_B^2$ energy above $V_{HR}$ to sustain the transmission $E_T$ and supply survival blocks $E_Q$ and $C_B$’s self-leakage $E_LK$ across the longest possible drought, which can be hours or longer [7]. But since the drought can begin just after a transmission starts, $C_B$ should hold another $E_T$. So to minimize the space $C_B$ occupies, $v_B$ at this point can be near the breakdown voltage $V_{BD}$ of the circuit. This way, $C_B$ can be 3.2 mF when $E_T$ is 1 µJ, $E_Q$’s $P_Q$ is 0.5 µW and $E_LK$’s $P_{LK}$ is negligibly lower, $T_{ON}$ is 2 hours, $V_{BD}$ is 1.8 V, and $V_{HR}$ is 1 V:

$$C_B = \frac{2E_T + E_Q + E_{LK}}{0.5(V_{TH(0)}^2 - V_{HR}^2)} = \frac{0.5(V_{BD}^2 - V_{HR}^2)}{0.5(V_{BD}^2 - V_{VHR}^2)}. \quad (2)$$

**Wake:** After the drought, when $v_H$ is back at or above $V_{HR}$, CP enables the charger whose output raises $v_B$ above $V_{HR}$. The system then waits across wake time $t_w$ until $C_B$ has
enough $0.5C_B(V_{MIN}^2 - V_{HR}^2)$ energy above $V_{HR}$ at $V_{MIN}$ to send an offline transmission and supply survival blocks $P_Q$ and $C_B'$s self-leakage $P_{LK}$ across another drought, which can be near 1.8 V when $C_B$ is 3.2 mF, $E_T$ is 1 µJ, $P_Q$ is 0.5 µW and $P_{LK}$ is negligibly lower, $T_{ON}$ is 2 hours, and $V_{HR}$ is 1 V:

$$V_{MIN} = \sqrt{E_T + P_Q T_{ON} + P_{LK} T_{ON} + V_{HR}^2 + 0.5C_B}.$$  

(3)

CP also waits for $P_{CHG}$ to reach $P_{TH}$ to enable the power supply, and in so doing, bring the system back online. Since the charger delivers a $P_{CHG}$ portion of $P_B$ where $\eta_{BS}$ is steady-state efficiency, $V_{CHG}$ can reach $V_{MIN}$ in 6.9 minutes when $C_B$ is 3.2 mF, $V_{MIN}$ is 1.8 V, $V_{HR}$ is 1 V, $P_H$ is 10 µW, and $\eta_{BS}$ is 87%:

$$t_w = \frac{0.5 C_B (V_{MIN}^2 - V_{HR}^2)}{P_{CHG} \eta_{BS}}.$$  

(4)

### B. Sleep

When designed for sleep, $C_B$ can drain completely. $C_B$ should therefore hold the energy needed to sustain an offline transmission $E_T$ and supply survival blocks $P_Q$ and $C_B$'s self-leakage $P_{LK}$ across a transmission period $T_T$. But since the drought can begin just after a transmission starts, $C_B$ should hold another $E_T$. This time, however, $T_T$ is so short that $E_Q$ and $E_{LK}$ are negligible. So to minimize the space $C_B$ occupies, it can be at $V_{BD}$. $C_B$ can therefore be 1.8 µF when $E_T$ is 1 µJ, $T_{ON}$ is 1 ms, $P_Q$ is 0.5 µW, $V_{BD}$ is 1.8 V, and $V_{HR}$ is 1 V.

![Fig. 6. Sleep simulation.](image)

**Wake:** Like in standby, $C_B$ enables the charger after the drought. When $C_B$ collects the energy for an offline transmission and $P_{CHG}$ reaches $P_{TH}$, $C_B$ enables the power supply and the system. The difference here is, $C_B$ charges from 0 V, not from $V_{HR}$. This means, the charger must, initially, derive power from $V_{CHG}$, which can be so low at 40–500 mV that startup efficiency $\eta_{BST}$ is often less than 0.7% [12], [14]–[15].

In other words, the charger first delivers a $P_{CHG}$ portion of $P_B$ to raise $V_{CHG}$ to $V_{HR}$ and then a $P_{CHG}$ portion to raise $V_{CHG}$ to $V_{MIN}$ [14]. So with 1.8 µF, for example, $V_{MIN}$ is 1.45 V when $E_T$ is 1 µJ and $V_{CHG}$ is 1 V, where $T_{ON}$ is zero. $t_w$ can therefore be 13 s when $P_H$ is 10 µW, $E_T$ is 0.7%, and $\eta_{BS}$ is 87%:

$$t_w = \frac{0.5C_B V_{HR}^2}{P_{CHG}} + \frac{0.5C_B (V_{MIN}^2 - V_{HR}^2)}{P_{CHG} \eta_{BS}}.$$  

(5)

**Temporary Supply:** One way to accelerate the wakeup process is to keep the charger above its minimum headroom level $V_{HR}$ with a smaller temporary battery $C_T$ [12], [15]. This way, $C_T$ charges quickly above $V_{HR}$ and with more than $V_{HR}$ feeding the charger, the charger is more efficient and therefore faster. $C_T$, however, should only feed the charger until $V_{CHG}$ reaches $V_{HR}$ because, at and above $V_{HR}$, $C_B$ can do the rest.

To be more specific, $C_T$ supplies gate-drive and quiescent power $P_T$ lost in the controller, which is a $P_{K_C}$ portion of the $P_{BH}$ drawn. Across one switching cycle $T_{SW}$, $C_T$ should therefore store above $V_{HR}$ this controller energy $E_C$ or $P_{CT}$:

$$C_T = \frac{E_C}{0.5(V_{BD}^2 - V_{HR}^2) - 0.5(V_{BD}^2 - V_{HR}^2)}.$$  

(6)

Resistances, however, consume another $P_{LK}$ fraction. So of the $P_{BH} - P_{K_C}$ delivered, $C_T$ receives $P_{K_C}$ and $C_B$ charges with $P_H - P_{K_C} - P_{K_R}$ or $P_{CHG}$. $C_B$ in Fig. 7 therefore charges with $P_{CHG}$ to $V_{HR}$ in $t_{HR}$ across which time $C_T$ recharges $N$ times:

$$N = \frac{t_{HR} T_{SW}}{T_{SW} P_{CHG}}.$$  

(7)

Unfortunately, Ohmic losses are higher when using $C_T$ because the charger passes more power at $P_B$ than without $C_T$ at $P_B - P_{K_C}$, so $\eta_{BS}$ and $C_T$ is $(P_{K_C})/\delta_{H} / (P_{K_C} / \delta_{k} k_{k})$. So of $\eta_{BS}$ is $k_{k} k_{k}$.

When first waking with startup efficiency $\eta_{BST}$, however, $C_T$ should charge just high enough above $V_{HR}$ to supply $E_C$:

$$V_{CHG} = \sqrt{E_C + V_{HR}^2 - \frac{P_{CHG} T_{SW}}{0.5C_T} + V_{HR}^2}.$$  

(8)

So when waking, $C_T$ charges to $V_{CHG}$ with $P_{CHG}$ and $C_B$ first charges to $V_{HR}$ with $P_{CHG}$ or $P_{CHG}$ and $C_B$ then to $V_{MIN}$ with $P_{CHG}$ or $P_{CHG}$:

$$t_w = \frac{0.5C_T V_{CHG}^2}{P_{CHG} \eta_{BST}} + \frac{0.5C_B (V_{MIN}^2 - V_{HR}^2)}{P_{CHG} \eta_{BS}}.$$  

(9)

Following the same example, $C_T$, $N$, $V_{CHG}$, and wake time $t_w$ can be 71 µF, 1040, 1.8 V, and 220 ms when $T_{SW}$ is 100 µs and $k_C$ and $k_k$ in $\eta_{BS}$ are 8% and 5%.

![Fig. 7. Wake simulation with and without temporary supply $C_T$.](image)

### IV. DESIGN TRADEOFFS

Although modern survival circuits consume very little power, they still discharge the battery $C_B$ across long harvesting droughts [16]. Super capacitors suffer more because they also leak [6]. Unfortunately, keeping $C_B$ charged above the minimum headroom level $V_{HR}$ that circuits need to operate across these outages requires substantial capacitance, and as a result, volume and wake time, as Fig. 8 states. This is why sleep mode saves space, because while asleep, $C_B$ can drain. This way, $C_B$ and wake time can be 1800× and 1880× lower after a 2-hour drought when allowed to sleep than when kept in standby.

The challenge with sleeping is waking with a drained $C_B$ and a millivolt source. This is because chargers output less than 0.7% of the power drawn when supplied from 40–500-mV [12]. The problem with this is a long wake period. A smaller temporary battery $C_T$ can help because $C_T$ charges quicker, and
with $C_T$’s $V_T$ above $V_{HR}$, the charger is more efficient (at 87% [13]) and therefore faster. This way, $C_T$ delivers the energy lost in the controller. In the example cited, wake time is 59× times shorter with only 0.004% more capacitance for $C_T$.

![Diagram](image)

**Fig. 8.** Wake design flow chart.

In practice, startup efficiency $\eta_{ST}$ determines wake time $t_w'$ without $C_T$, steady-state efficiency $\eta_{SS}$ sets the counterpart $t_w''$ with $C_T$, and $k$/$k_*$'s reduction in $t_w''$ is normally insignificant. Wake improvement factor $k_i$ with $C_T$ therefore reduces to

$$k_i = \frac{t_w'}{t_w''} = \left(\frac{\eta_{ST}}{\eta_{SS}}\right) \left(\frac{V_{HR}}{V_{MIN}}\right)^2. \quad (10)$$

In other words, improvement $k_i$ hinges on how much $\eta_{SS}$ overweights $\eta_{ST}$ and how little $V_{MIN}$ surpasses $V_{HR}$. Because with lower $\eta_{SS}$, the losses that $C_T$ supplies climb, so $C_H$ receives less power. And with a higher $V_{MIN}$, $C_B$ requires more time to charge. This is why $C_T$ reduces wake time (i.e., $k_i$ exceeds 1) in Fig. 9 when $\eta_{SS}/\eta_{ST}$ outweighs $(V_{MIN}/V_{HR})^2$ and $k_i$ is 59× in the example cited, which represents a typical case.

![Diagram](image)

**Fig. 9.** Wake-time improvement factor.

V. CONCLUSIONS

To keep onboard batteries small, microsystems should perform only one task at a time, and do so as soon as stored energy is sufficient to sustain a task. And when ambient energy vanishes, the system should stand by or sleep. This paper proposes an algorithm and a design that allow microsystems to sleep and reduce battery size by 1800× and wake time by 180×. With only 0.004% additional capacitance for a temporary battery, the system wakes 59× faster from a no-charge condition. This way, with the low-energy "just-enough" schedule, analytical methods, and design process proposed here, smaller energy-harvesting sensor systems can sustain more events more frequently and recover from harvesting droughts more quickly.

ACKNOWLEDGEMENT

The authors thank Texas Instruments for sponsoring this research and Paul Emerson and Dr. Orlando Lazaro for their mentorship and support.

REFERENCES


