Grafting Energy-Harvesting Leaves onto the Sensornet Tree

Lohit Yerva[†], Bradford Campbell[†], Apoorva Bansal[†], Thomas Schmid[‡], and Prabal Dutta[†]

[†]Computer Science & Engineering Division
University of Michigan
Ann Arbor, MI 48109
{lohity,bradjc,apoorvab,prabal}@eecs.umich.edu

[‡]Electrical and Computer Engineering Dept
University of Utah
Salt Lake City, UT 84112
thomas.schmid@utah.edu

ABSTRACT

We study the problem of augmenting battery-powered sensornet trees with energy-harvesting leaf nodes. Our results show that leaf nodes that are smaller in size than today's typical battery-powered sensors can harvest enough energy from ambient sources to acquire and transmit sensor readings every minute, even under poor lighting conditions. However, achieving this functionality, especially as leaf nodes scale in size, requires new platforms, protocols, and programming. Platforms must be designed around low-leakage operation, offer a richer power supply control interface for system software, and employ an unconventional energy storage hierarchy. Protocols must not only be low-power, but they must also become low-energy, which affects initial and ongoing synchronization, and periodic communications. Systems programming, and especially bootup and communications, must become low-latency, by eliminating conservative timeouts and startup dependencies, and embracing high-concurrency. Applying these principles, we show that robust, indoor, perpetual sensing is viable using off-the-shelf technology.

Categories and Subject Descriptors

B.m [**Hardware**]: Miscellaneous

Keywords

energy-harvesting sensor node, low power wireless networking

1. INTRODUCTION

For the past decade, academia and industry alike have labored to make battery-powered, multi-hop wireless sensornets a reality. Now, with multi-hop networking firmly ensconced in the standards bodies, many have concluded that periodic node maintenance represents *the* biggest impediment to continued technology deployment. Therefore, some have turned to energy-harvesting operation. However, the two prevailing approaches – EnOcean [4] and ZigBee Green Power [23] – adopt a star topology that requires leaf nodes to be within a single hop of an always-powered base station. This insidious constraint, we argue, regresses on a decade of advances in multi-hop mesh networking – and its diverse benefits – including better spatial reuse of spectrum, higher reliability due to path diversity, and longer lifetime due to reduced transmission power. Moreover, we argue that energy harvesting and mesh networking are not exclusive, and that they can coexist within a unified network architecture.

Today, industry is advocating a suite of new and emerging technologies – energy-harvesting transducers [13], thin-film batteries [1], micropower integrated circuits [9], and nanopower microcontrollers [12] – coupled with star network topologies, to address the pressing challenges of energy harvesting operation. In contrast, we show how existing technologies – solar cells, simple capacitors, switching regulators, field-effect transistors, and low-power microcontrollers and radios – can be combined in new ways, and coupled with simple protocols and good engineering, to achieve hassle-free, energy harvesting operation, without sacrificing the benefits of interoperating with battery-powered meshes.

This paper shows how the addition of a stable clock [11] and some minor software improvements to existing battery-powered mesh nodes (*branch* nodes in our terminology) prepares them to interact with energy-harvesting *leaf* nodes. And the leaf nodes themselves are nothing more than branch nodes whose batteries have been replaced with a small solar cell, a few capacitors, a pair of voltage monitors, and some transistors. Of course, such simple leaf nodes only work when the

lights are on (even dimly), but night time operation would be possible from a source that could supply 3 V at 2.5 μW [15].

Although leaf nodes are constructed from widely-available electronic components, their integration requires some care. In particular, since leaf nodes operate from anemic power sources – as low as just few microwatts – their chief design constraint is *low-leakage*, *low-power* operation. Therefore, always-on switching regulators or even low-dropout linear regulators are often unsuitable due to their relatively high quiescent or ground currents. Our leaf design minimizes leakage and other losses by completely switching off and then cold booting the processor and radio during each activity cycle. We also eschew batteries due to their limited charge cycles and high cost, and miniature (surface mount) supercapacitors due to their high impedance, and choose instead capacitors that offer very low impedance. Despite these many restrictions, leaf nodes are still able to interoperate with mesh networks at indoor irradiance levels of about 15 μW/cm^2, which is at the very low end of what we observe indoors.

While low-leakage hardware is the *sine qua non* of energy harvesting operation, several networking problems must also be solved to augment battery-powered mesh networks with energy-harvesting leaf nodes. These problems include initial synchronization, ongoing synchronization, and bidirectional communications (branch-to-leaf, leaf-to-branch). The chief challenge in solving these problems lies in achieving *low-energy* operation. While nodes with copious reserves can consume energy with few restrictions, and battery powered ones must merely operate at low *average* power, our energy-harvesting leaf nodes are constrained by the energy in their (tiny) capacitors. This translates to low-energy (i.e. roughly 1 mJ) budgets for synchronization and communications. Fortunately, low-energy neighbor discovery protocols exist [2], and low-energy communications is possible through networking and software optimizations. Most of these optimizations focus on achieving *low-latency* cold boot, radio startup, and radio RX/TX turnaround. No single optimization is sufficient, but all are required. Collectively, they enable a leaf to discover and communicate with a branch, and deliver data readings every minute, even under low-light, using a solar cell that is just a few square centimeters in surface area.

While the low-energy constraint may seem like an artificial one, the reality is that low-energy operation will become critical as nodes continue to scale in size. Since batteries and solar cells currently dominate node volume, they must shrink in order to enable future leaps in the minimization of sensor nodes. However, some may wonder whether energy-harvesting really makes sense or whether batteries will suffice, especially in the context of this continued scaling. Therefore, it is worth exploring the question, *at what scale is indoor photovoltaics the better primary power source?* One way to attack this question is to assume the entire vol-
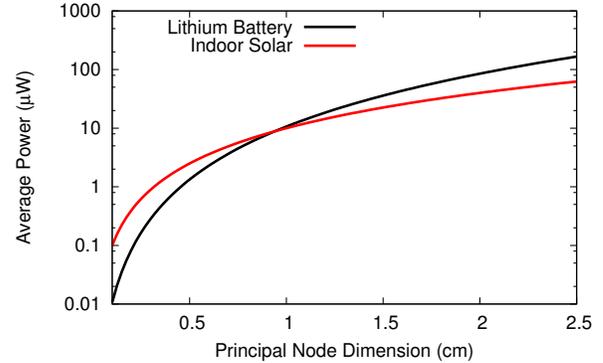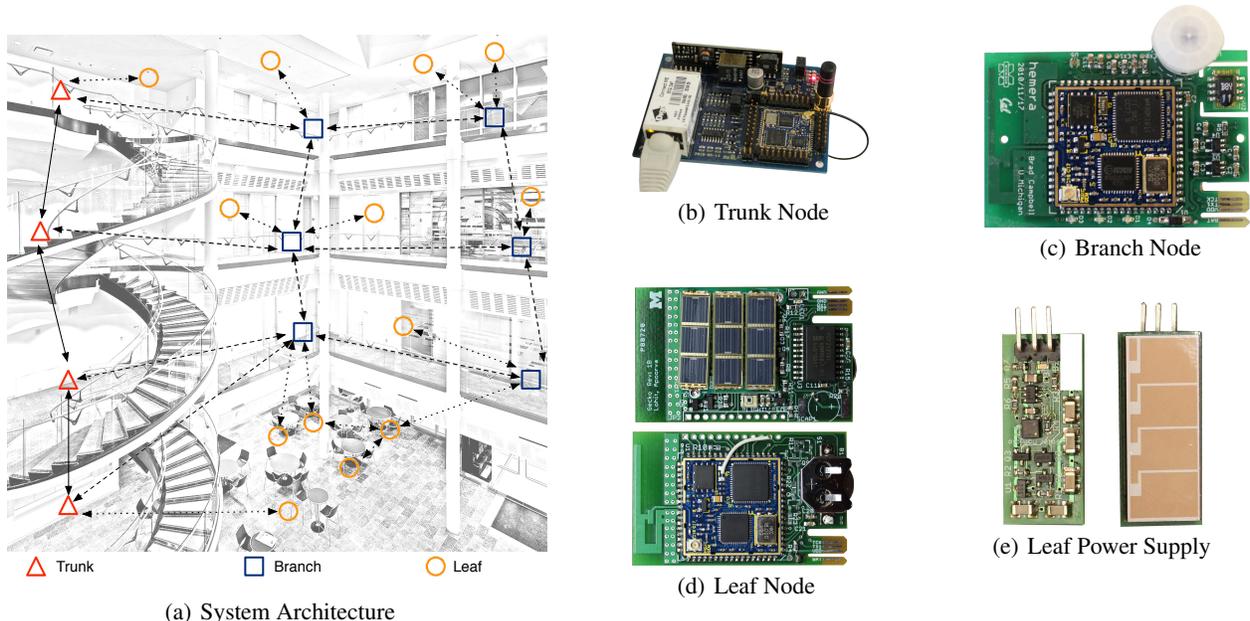

Figure 1: An energy-harvesting reality check. Shows how power harvested from indoor solar compares with power drawn from an internal battery. As a cubic sensor's length L falls below a centimeter, a solar cell of size L^2 can deliver higher average power than a Lithium battery of size L^3, over a seven year horizon. The key to continued sensor scaling lies in shifting the primary energy supply from battery to solar, and dealing with the implications of a dramatically reduced supply.

ume (L^3) of a cubic sensor of length L is devoted to energy storage and that volume is occupied by a non-rechargeable Lithium primary cell whose energy density, ρ, is 653 mW-h/cm^3 and whose useful life, T, is bounded to seven years (due to its shelf-life). The average power the battery could source is $P = \rho L^3 / T$. A conservative estimate of the average solar irradiance on an indoor surface, H_d, is 10 μW/cm^2, and the average power is $P = H_d L^2$ [6]. Setting these two expressions equal to each other and solving for L gives 1 cm as the crossover point where solar (1 cm^2) beats batteries (1 cm^3) over a seven year horizon, as Figure 1 shows.

Of course, this analysis ignores several factors, such as the overhead of battery packaging, the low efficiency of solar conversion, and the unrealistic node volume dedicated to the battery. However, the general trend is clear: when nodes shrink to centimeter scales and beyond, energy-harvesting will play a critical role.

The goal of this research is to understand the design space of low-maintenance, high-density sensor networks. We do so by designing and studying an energy-harvesting, low-energy node using current technology. We show that using already available parts, we can build a working solar-powered node with ultra-low leakage currents (1.7 μA). Further, we show that this node works in very low indoor lighting conditions (15 μW/cm^2), and delivers data every minute, which is comparable to existing battery-powered nodes. We adapt existing protocols to meet the challenges of networking an extremely energy-limited node with battery-powered meshes. We demonstrate robust techniques for handling resynchronization when nodes suffer extended periods of power loss, thus solving the critical network bootstrap problem.

(a) System Architecture

△ Trunk □ Branch ○ Leaf

(b) Trunk Node

(c) Branch Node

(d) Leaf Node

(e) Leaf Power Supply

Figure 2: Integrating energy-harvesting leaf nodes with battery-powered mesh nodes. (a) The system architecture shows how different node classes interact with each other. Mains-powered "trunk" nodes provide connectivity between the external world (e.g. Internet) and the sensornet "tree." Trunk nodes communicate wirelessly with battery-powered "branch" nodes, and branch nodes mesh with other "branch" nodes, just like in today's sensornet mesh architectures. Leaf nodes may communicate with trunk, branch, or in some cases even other leaf nodes depending on a leaf's capabilities (but not explored in this paper, however). (b) A trunk node with an Ethernet and an 802.15.4 interface (using the Epic Core). (c) A battery-powered branch node with temperature, humidity, light, and motion sensors. (d) An energy-harvesting leaf node that can keep accurate time (using a coin cell). (e) Leaf node power supply details (the same electronics exist under the three solar cells on the leaf node).

2. SYSTEM OVERVIEW

Figure 2 shows the overall system and network architecture, including wall-powered *trunk* nodes, battery-powered *branch* nodes, and energy-harvesting *leaf* nodes. Trunks and branches are roughly equal to basestations and mesh nodes, respectively, in contemporary architectures.

2.1 Leaf and Branch Platforms

Leaf nodes integrate a sensor node "core" like the Epic mote [3] with an energy-harvesting power supply and accurate time-keeping, as Figure 2(d) shows. We employ five basic design principles in leaf nodes to achieve high-efficiency operation. First, we minimize power transfer inefficiencies by operating near the solar cell's maximum power point. Second, we minimize power conversion inefficiencies by doing only one power conversion – from the solar cell buffer capacitor directly to the processor and radio supply voltage. Third, we minimize leakage by power cycling and cold booting the processor and radio. Fourth, we improve energy consumption efficiency by optimizing system software to minimize latencies. Fifth, we minimize communications costs by shifting most of the synchronization burden to more capable (but fewer in number) branch nodes, user hierarchical discovery, and employ temperature-compensated crystal oscillators (TCXOs) to further reduce that burden.

Branch nodes, shown in Figure 2(c), are very similar to leaf nodes. The key difference is that they replace the leaf node's power supply (including a solar panel, capacitors, transistors, voltage monitors, and regulator) with just a battery and regulator. The branch node also employs a real-time clock that is more stable and draws slightly more power than a leaf node's [11]. The greater clock stability allows branch nodes to limit communication guard times since the worst case drift is constrained by the *sum* of the magnitude of the two individual clocks. Branch nodes are otherwise identical to conventional sensor nodes like the Telos [16].

2.2 Leaf-to-Branch Communications

Leaf nodes wake up on a fixed period, take a sensor reading, transmit a packet, and listen for inbound traffic. If a leaf does not have enough energy to perform these operations, it simply skips one (or more) activity cycles and simply resumes its activities when it has accrued enough energy. The key is that the nominal activity period is fixed and has low jitter and drift. Branch nodes learn of leaf node schedules using a low duty-cycle neighbor discovery protocol as described in section 4.2.1. Branch nodes subsequently track the nominal wakeup times of leaf nodes and listen briefly (i.e. 20 ms, searching over a variable-length guard time) for leaf transmissions, and to deliver any leaf-bound traffic.

(a) Leaf Architecture   (b) Leaf Power Supply Details
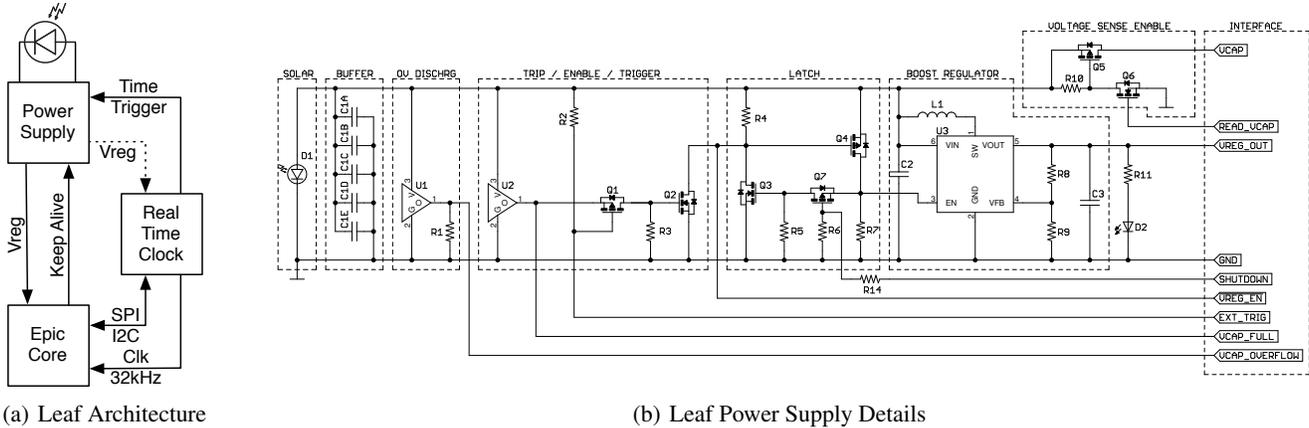
Figure 3: (a) Energy-harvesting leaf nodes include a power-supply circuit, a processor/radio core (Epic Core), and a real-time clock (RTC). The power supply switches the core on and off in response to time triggers and capacitor voltages, and the core can choose to disconnect power. (b) The power supply stores indoor solar currents of just a few microamps and provides periodic burst of regulated voltage. The power supply exports several monitoring and control signals to allow applications to tailor their operation to the key power supply events.

3. LOW-POWER LEAF NODE DESIGN

This section presents the leaf node hardware and software. A leaf node integrates a processor and radio [3], real-time clock, and energy-harvesting power supply, as Figure 3 shows.

Epic Core mote [3] incorporates an MSP430F1611 microcontroller and CC2420 radio (802.15.4-compliant). The MSP430 offers very fast wakeup and low active current, allowing us to minimize both the startup and active energy.

Real-Time Clock. A real-time clock (RTC) provides a periodic time trigger to instigate leaf activity. Our leaf node uses the NXP PCF2127A RTC [14], a temperature-compensated crystal oscillator which runs at 32.768 kHz. This device offers excellent time-keeping stability (± 3 ppm), low current draw (0.65 μA) on backup battery, and flexible triggering options. Collectively, these features support efficient synchronization between leaf and branch nodes. The RTC also offers a battery-backed SRAM, which allows a node to maintain state across activity cycles without keeping the microcontroller powered on. The prototype's batteries can be replaced with rechargeable super-capacitors to allow nearly infinite charge and discharge cycles at the cost of long term time keeping.

Power Supply. The energy-harvesting power supply (Figure 3(b)) integrates a solar panel, capacitors, overvoltage protection, various trip and enable triggers, a latching circuit, regulator, voltage sense circuits, and a rich interface to the power supply. We evaluate two different solar panels – an IXYS XOB17-4X3 (crystalline silicon) and a Sanyo AM-1437 (amorphous silicon) – for their response to different light sources. We chose (two 100 μF) tantalum capacitors as the primary energy buffer due to their high energy density and current transfer efficiency for voltage regulators. The

activity and overvoltage triggers use low-current voltage supervisors from Panasonic and Microchip, which draw only 0.25-0.5 μA, respectively. We use the MCP1640 boost converter with true load disconnect to provide a stable voltage and switch power to the Epic Core. Finally, a FET-gated sense path allows the microcontroller to measure the capacitor voltage.

3.1 Hardware Operation

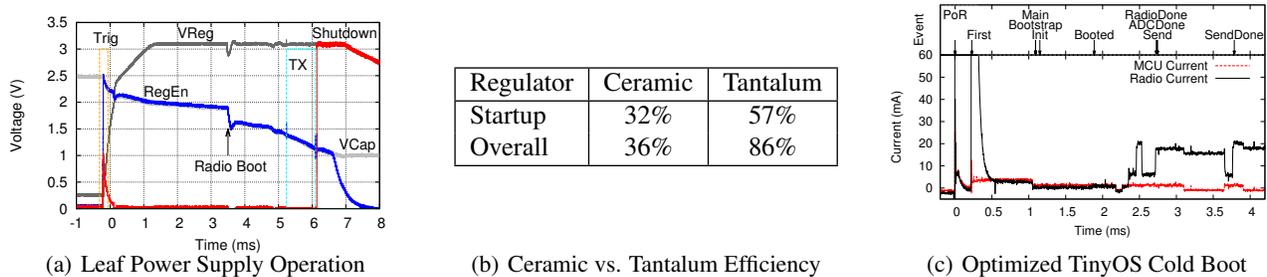The hardware goes through three distinct operating stages:

Charge. From a fully-discharged state, the solar cells begin to charge the buffer capacitors. Voltage supervisor U2, a Panasonic MS1382SEL rated to trip at 2.5 V, monitors the capacitor voltage. Since the solar cell maximum power point occupies a fairly narrow range of a few hundred millivolts, as Figures 4(a) and 4(b) show, we are able to use fixed voltage levels for the charge (and discharge) thresholds rather than needing or employing a costly maximum power point tracking (MPPT) scheme.

Startup. When the capacitors reach U2's trip voltage, U2's output goes high, initiating the active phase. Assuming `EXT_TRIG` is low when this happens,[1] P-FET Q1's source goes high, turning Q1 on, and passing the U2's high output to Q2's gate, turning Q2 on. This drives Q4's gate low, turning it on, and causing regulator U3's `VREG_EN` to be driven high, initiating its startup. Instantly, the default *on* Q7 enables the bypass Q3. The bypass latch (Q4 keeps Q3 on, which keeps Q4 on) keeps the voltage regulator enabled even

---

[1]This could be done efficiently by, for example: (i) connecting `EXT_TRIG` to the output of the RTC's periodic interrupt or (ii) connecting `VCAP_FULL` to an N-FET's gate, `EXT_TRIG` to an N-FET's drain, and `GND` to and N-FET's source. We use option (i) and preprogram the RTC during hardware bringup.
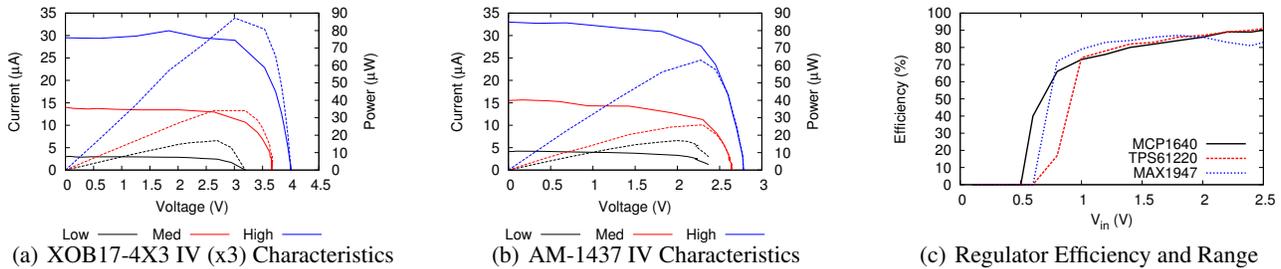
Figure 4: Solar cell current vs voltage, and power vs voltage, under low (12.7 μW/cm^2), medium (53.8 μW/cm^2), and high (101.7 μW/cm^2) indoor lighting levels. Although solar cell output current varies by an order of magnitude, the maximum power point varies within a small range of approximately (a) 2.6 V to 3.0 V and (b) 2.1 V to 2.3 V, or about (10-15%). This lets us use fixed-threshold voltage trip switches for charge/discharge operation under typical indoor lighting conditions. (c) MCP1640 was choosen due to its efficiency over a large input voltage range, and because it has output disconnect and low leakage current.

Regulator	Ceramic	Tantalum
Startup	32%	57%
Overall	36%	86%

(a) Leaf Power Supply Operation (b) Ceramic vs. Tantalum Efficiency (c) Optimized TinyOS Cold Boot

Figure 5: (a) Operation of the energy harvesting power system. The capacitor voltage is seen dropping from about 2.5 V to about 1 V. The boost regulator starts up and establishes a 3.0 V supply until the core requests shutdown after it finishes its radio transmission. *Radio Boot* refers to the execution of the first instruction, enable radio regulator. (b) Efficiency of tantalum vs ceramic capacitors as energy buffers for the boost voltage regulator. At startup and as the input voltage drops, capacitors struggle to provide the current needed by the regulator to maintain output power. Boost regulator *startup* refers to bringing output capacitors to the regulated voltage. The overall efficiency includes a 100 Ohm output load. (c) Stages of an optimized TinyOS cold-boot overlaid with the processor and radio current. Our design achieves cold-boot to packet transmission in less than 4 ms using all of the optimizations discussed. The net effect is a reduction in TinyOS cold boot by a factor of 69$\times$ in time and 14.6$\times$ in energy.

though U2 disables Q2, when the capacitor voltage eventually drops to 2.3 V (maximum U2's histeresis) as U3 goes into regulation. We use the MCP1640 boost regulator, which provides high conversion efficiency near our startup threshold voltage, and a slightly wider input voltage range as well for longer discharges, as Figure 4(c) shows. As U3 starts up and goes into regulation, it too discharges the buffer capacitor

Active. After startup, the processor boots, configures the radio, takes sensor readings, and transmits/receives packets. Once done, the processor drives SHUTDOWN high, which disables the regulator and disconnects power to the Epic Core, saving unused energy. Figure 5(a) details the operation of the startup and active stages.

3.2 Software Operation and Optimization

Low-energy and low-power leaf nodes need efficient and optimized system software support, and tight hardware-software interaction. Low-power operation begins in the first few instructions the processor executes and continues through system startup, communication, and sleep/shutdown. We detail some of the software operations that a leaf node employs to maximize energy efficiency.

Shutdown. Mechanisms for software to quickly disconnect the processor and radio and shut down the regulator to conserve charge once the software completes its work.

Oscillator Fast Start. The hardware starts using an internal RC oscillator with a low Q, which allows fast startup but poor frequency tolerance. However, frequency error is not a significant problem as all timing-based operations (e.g. timeouts) are converted into either interrupt-driven or polling-driven operations, and all I/O operations are synchronous (i.e. the processor provides the clock). The software sets the clock speed to the maximum allowed, which minimizes startup latency.

Optimized Startup. The default TinyOS distribution requires 237 ms and consumes 2.174 mJ from power-on to Radio.sendDone(). Most of the time is spent waiting for the 32 kHz clock to start oscillating. Adding an ex-

ternal 32 kHz source reduces the boot time to 48 ms and 1.81 mJ. Much of the remaining startup costs are spent initializing memory. Setting the microcontroller clock speed to the maximum possible frequency ($\sim$8 MHz in the case of the MSP430F1611) before initializing memory reduces the cold-boot to 18.6 ms and 0.61 mJ. This figure can be further reduced to achieve a cold-boot time of 3.4 ms and 149 μJ by removing several layers of the TinyOS radio stack, parallelizing radio startup and memory initialization, and using the bare radio interface to transmit a packet as shown in Figure 5(c).

Concurrent Initializations. System software employs concurrency to minimize latency during system initialization (contrary to the far more typical serialized system initializations elsewhere). The radio initialization is started, for example, before the C runtime zeroes out data memory, and the two operations proceed in parallel. Similarly, the packet to be transmitted is pipelined into the radio's transmit FIFO. Branch nodes use similar radio stack optimizations to respond rapidly to a leaf's probe with pending data.

The result of these software optimizations is shown in Figure 5(c). This figure shows a detailed timeline of a cold-boot to packet transmission. The labels indicate stages of the boot process. They are, from left to right, PoR: Power-on-Reset; First: first instruction; Main: call of `main()`; Bootstrap: after call to `platform_bootstrap()`; Init: after call of `PlatformInit.init()`; Booted: first instruction of `Boot.booted()`; RadioDone: Radio initialization done; ADCDone: ADC initialization done; Send: start of packet; SendDone: Return from send, ready to shut off. The corresponding latencies are: PoR to First: 0.17 ms, First to Main: 0.87 ms, Main to Bootstrap: 0.5 μs, Bootstrap to Init: 58 μs, Init to Booted: 436 μs, Booted to RadioDone: 842 μs, RadioDone to AdcDone: 8.1 μs, AdcDone to Send: 9.1 μs, Send to SendDone: 1.04 ms.

4. EVALUATION

This section evaluates the viability of energy harvesting operation, characterizes typical indoor lighting conditions, demonstrates initial and ongoing synchronization, and shows that leaf and branch nodes can communicate successfully.

4.1 Energy Harvesting Operation

The first question we explore is the relationship between irradiance and leaf node activity.[2] For this experiment, two leaf nodes are equipped with one type of solar cell and two others are equipped with a different type of solar cell. The four leaf nodes are programmed to transmit a packet and then disconnect the processor and radio from the power supply. The nodes are exposed to varying irradiance levels from four different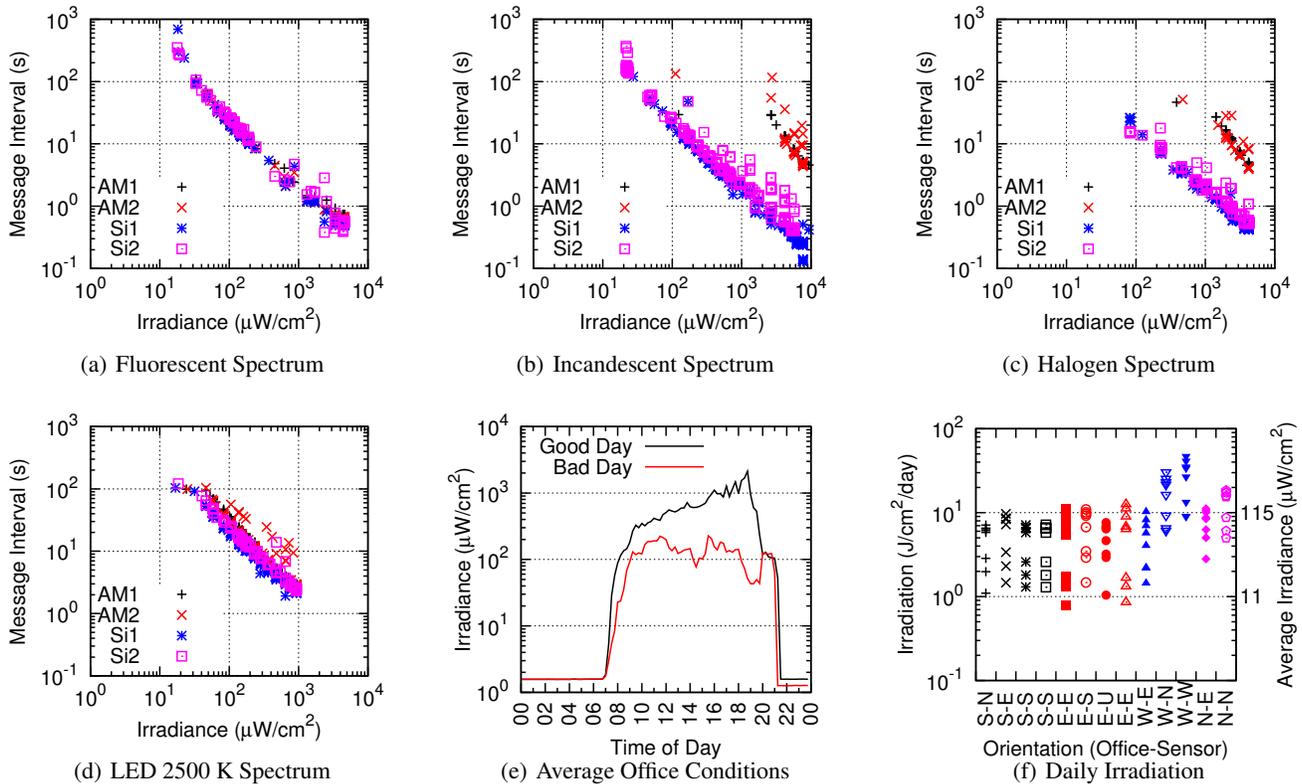 indoor light sources with different spectra. Our goal is to answer the question, *given a certain lighting level, how frequently can a leaf node transmit a packet?*

Figures 6(a) to 6(d) show the message interval vs irradiance for two different solar cells powering the same hardware under four different light sources across several decades of irradiance. Leaf nodes are configured to transmit a packet every time their capacitor voltage exceeds a threshold. The nodes are exposed to controlled light levels and have their packet transmissions logged. Nodes AM1 and AM2 use AM-1437 amorphous silicon solar cells while nodes Si1 and Si2 use *three* XOB17-4X3 crystalline silicon solar cells in series. Irradiance is measured using a custom sensor (based on the TAOS TSL230BR) calibrated with a professional meter.

Figure 6(a) shows that both amorphous and crystalline solar cells have similar conversion factors under fluorescent lighting conditions. Figures 6(b) and 6(c) show that crystalline solar cells exhibit better conversion in incandescent and halogen settings than amorphous solar cells by roughly an order of magnitude, as evidenced by the two distinct clusters that are visible. These results are to be expected since fluorescent lights have peaks in the 500 to 600 nm range which align with the spectral sensitivity of amorphous solar cells but tungsten-based lamps, like standard incandescent and halogen, emit much less energy in the 500 nm range than they do at longer wavelengths, like 700 nm. Figure 6(d) shows that both amorphous and crystalline solar cells perform more closely under LED lighting of 2500 K color temperature. Again, these results are to be expected since white LED lights have a peak near 450 nm from a blue GaN or InGaN monochromatic LED. The results show that baseline communications activity of leaf nodes is roughly linear with irradiance and that operation near fluorescent, incandescent, and LED indoor sources with irradiance levels below 20 μW/cm^2 is possible. The data also highlight the importance of considering the mix of light sources and matching them to the appropriate solar cells.

Figure 6(e) shows the average irradiance of four different offices (13 distinct locations) for a partly sunny ("good") and rainy ("bad") day. The data shows that even on a rainy day, average irradiance largely exceeds 100 μW/cm^{2}during daylight hours, suggesting many indoor locations are viable for leaf nodes. Figure 6(f) shows that daily irradiation can vary by more than an order of magnitude for a single location.[3] Most locations receive 1-10 J/cm^2/day (or an average of 11-115 μW/cm^2), which is enough to ensure frequent leaf activity. Figure 7 shows the instantaneous, rather than average, irradiance of the underlying data over a one week period. The traces also show that during Friday and Saturday evenings, the North Office was illuminated, and during Wednesday evening, the West Office was illuminated. This suggest evening or nighttime human activity in those rooms.

---

[2]Irradiance is a measure of radiation per unit area incident on a surface, and it is measured in μW/cm^2.

[3]Irradiation is accumulated irradiance per unit time, and it is measured in μJ/cm^2/day.

Figure 6: Characterizing leaf performance by solar cell type and indoor light levels. (a) Both types of solar cells respond well to indoor fluorescent lights. (b) Crystalline silicon solar cells (Si1 and Si2) respond well to incandescent but amorphous solar cells (AM1 and AM2) generate an order of magnitude fewer packets at the same irradiance. (c) Halogen lights also result in order-of-magnitude reduction in packet transmissions for amorphous solar cells compared to fluorescent lights. (d) LED lighting shows much more comparable performance between both the two types of solar cells and output in comparison with fluorescent lights. (e) shows the average irradiance of four typical offices over the course of a partly sunny day (good) and a rainy day (bad); even on a bad day, a leaf harvests enough energy for a packet transmission every few minutes. (f) shows daily irradiation across 13 sensors deployed in four offices over a one week period including sunny and rainy days. Thus, the IPV leaf design is viable.

In all three cases of human activity, the light levels exceeded $100 \, \mu W/cm^2$.

The U.S. Department of Labor's regulation on minimum office illumination [20] is 30 foot-candles, which corresponds to about 320 lux. The regulations are lower for indoor corridors and hallways at 5 foot-candles, or about 53 lux. From our own experiments using a commercial lux meter and a TAOS TSL230BR [19] light to frequency converter we find that 50 lux of a fluorescent light corresponds to $18.6 \, \mu W/cm^2$, 100 lux to about $29.1 \, \mu W/cm^2$, and 320 lux to $74.9 \, \mu W/cm^2$.

These figures suggest that our leaf nodes, as built, are capable of operating at minimum office illumination levels and, in many cases, substantially below them. Furthermore, when high-fidelity sensing is required – when people are present – typical light levels are far above the minimum levels needed for operation and nodes can easily report data every minute, without requiring supercapacitors or batteries. As processors and radios become more energy efficient, power supplies be-

come more efficient, and real-time clocks draw lower power, cubic-cm scale pervasive sensors will become increasingly practical.

4.2 Initial Synchronization

We evaluate two techniques to synchronize leaf and branch nodes. The first approach employs an asymmetric, asynchronous neighbor discovery protocol that shifts much of the neighbor discovery burden onto branch nodes. The second approach employs a shared, external event from the node's common environment to synchronize nodes.

4.2.1 Asynchronous Neighbor Discovery

We use a variant of the Disco [2] neighbor discovery protocol in which nodes transmit beacons on eventually overlapping schedules. In our case, only leaf nodes transmit and only branch nodes listen. The listen and transmission sched-
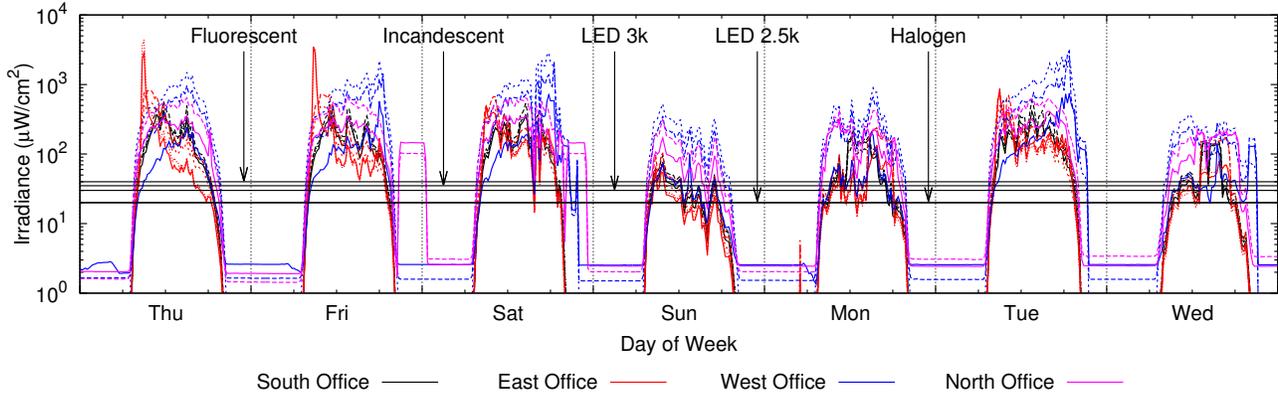
Figure 7: Irradiance (on a log-linear scale) over a typical week in four offices, oriented in four different directions. Each office is equipped with two to four typically wall-mounted sensors, each facing in different directions within the office, to capture different light environments. The horizontal lines correspond to the irradiance necessary for a leaf node to send a message every minute given the respective light source type. On most days, most leaf nodes have enough light to send a message once a minute, but on some days, some do not. This highlights the importance of node placement and solar cell sizing. It also illustrates the need to support a wide operating dynamic range.

ules are chosen such that the worst case discovery latency is about 50 min and the discovery burden is small for both.

Leaf nodes transmit beacons during a 5 ms window ("slot") every 60 s. This translates to a nominal duty cycle of 83 ppm (or 0.0083%). Figure 8(a) shows a free-running leaf transmits packets whenever it has sufficient energy but that a time-triggered leaf transmits packets at multiples of 60 s, allowing a branch node to both employ a compatible neighbor discovery schedule and predict future transmission times.

Branch nodes listen for beacons during a 5 ms window ("slot") every 245 ms (or 49 slots). This translates to a nominal duty cycle of 2.04% but inefficiencies and latencies in radio startup result in a higher duty cycle (2.5%), which translates to a roughly 1.5 mW power load due to discovery. The baseline discovery power is higher than ideal, but using newer radios that offer similar radio sensitivity at roughly 15% of the receive current, the baseline discovery power could be reduced to about 500 μW. Given the nominal duty cycles of 0.0083% and 2.04%, the worst case discovery latency is (5 ms)/(0.0083% $\times$ 2.04%), or about 50 min (2940 s) in the absence of communications failure, as Figure 8(b) shows.

4.2.2 Synchronous Event Triggered

To explore the viability of basic event-triggered synchronization, we design a simple, zero-power, light-activated trigger switch. It consists of a solar cell connected in parallel with a burden resistor. The solar cell's negative terminal is connected to an N-channel FET's source (and leaf GND), and the positive terminal is connected to the FET's gate. The FET's drain is connected to the power supply's external trigger or an interrupt line. This forces the leaf node to cold boot, and the branch node to wake up from sleep, in response to a sudden change in lighting. The nodes then exchange packets.

Figure 8(c) shows the operation of two nodes, a leaf (N1) and a branch (N2), in response to an external trigger. The leaf node transmits first, and then the branch node responds. The two nodes transmit packets back and forth until the leaf depletes its energy. This illustrates that for sufficiently "sharp" triggers, like light, accoustics, or sudden movement, it is possible for both leaf and branch nodes to synchronize.
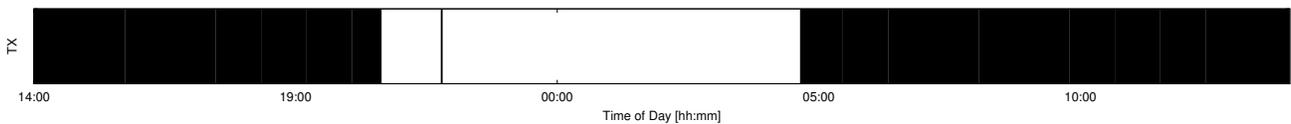
4.3 Ongoing Synchronization

Once a leaf and branch node have synchronized, they need to maintain ongoing synchronization to operate. In our design, the entire burden of maintaining synchronization rests with the branch node. The leaf node simply transmits synchronization beacons or data packets at every multiple of 60 s whenever it has enough energy to do so. The branch node is responsible for tracking the leaf node's transmission times and scheduling itself to be on during those times. Upon successful packet reception, a branch node can update its estimate of the leaf node's clock drift, if any. And, upon a missed transmission, the branch can adjust a guard window as needed.

In order to maintain a suitably low branch node duty-cycle, the maximum guard window size must be capped. When the clock drift requires a larger guard time, we use a local search pattern. In this search, the branch node iterates through the entire guard window by waking up for a different portion of the guard time in its periodic wakeups. For example, if the guard window is 100 ms, the branch would wake up for each 20 ms portion over a five minute period.

To evaluate the reliability of ongoing synchronization (and, by extension, data communications), a leaf node runs in a lit room. Initially synchronized, the nodes maintain synchronization for 6.6 hours. When the room becomes dark, the

(a) Time-Triggered Listening (b) Branch Radio Duty-Cycle (c) Event-Triggered Synchronization
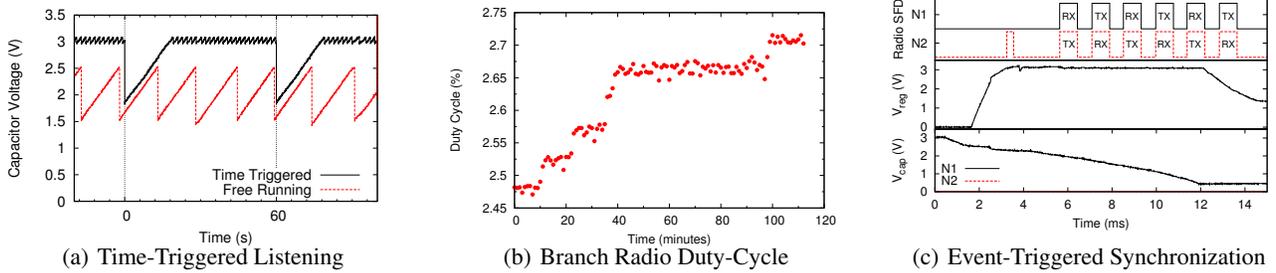
Figure 8: Achieving initial synchronization. Methods for achieving initial synchronization include both asynchronous neighbor discovery and synchronous event-triggered discovery. (a) A free-running leaf transmits a packet when it has sufficient energy. A triggered leaf transmits at a well-known interval. (b) Each point represents the duty cycle of each minute. Each step in the graph corresponds to the discovery of a leaf node. Even while synchronized with five leaf nodes, the branch node's duty cycle only slightly increases from 2.45% to 2.75%. (c) A trace of synchronous event-triggered discovery is shown. Leaf (N1) and branch (N2) nodes both respond to a sudden change in light level (triggered by a zero-power light detector circuit), choose a short backoff, and transmit packets back and forth until the leaf depletes the energy in its storage capacitor.

Figure 9: Maintaining ongoing synchronization. Each vertical line represents a successful leaf to branch packet transmission. Around 20:00 hrs, the leaf stops transmitting due to a dark environment. When light returns at 4:45 hrs, the branch begins to receive the packets without requiring resynchronization.

leaf ceases transmissions, except for a brief moment when it turns on for about two minutes. After seven hours, the leaf resumes transmissions, and the branch takes up to 4 minutes to find it again. The received packets are shown by vertical lines in Figure 9. The branch uses a fixed, 5 ms slot, but with clock stability's of ±1 ppm for a branch and ±2 ppm for a leaf, the guard times may need to be adjusted. In the worst case, $t_{guard} = (|\pm 1| + |\pm 2|) \times 3600 \times 8 = 86.4$ ms after eight hours, or a maximum of a five minute resynchronization delay due to the local search pattern.

4.4 Leaf to Branch Communications

Once a leaf and branch are synchronized, they may need to transfer data bidirectionally. Transmission from leaf to branch is relatively straightforward as a leaf simply wakes up at a multiple of 60 s and transmits. If the leaf does not have enough energy to transmit, as Figure 10(a) shows at time 120 s, it simply skips that transmission, but the branch still listens in anticipation.

Although communication data paths have been optimized for low-latency operation, some jitter still remains. To characterize this jitter, a leaf node is programmed to cold boot every 60 s and transmit a packet to a branch nodes. The branch node records the reception time using a 32 kHz TCXO. Figure 10(b) shows the cumulative distribution of leaf-branch synchronization jitter. A ±500 μs range captures the cold boot to packet transmission jitter. A branch node must account for this jitter with a 1 ms (or longer) guard time.

Finally, reliable communications requires that a leaf's transmissions be acknowledged. This is accomplished by transmitting a packet with an ACK request, which causes the branch to quickly transmit an ACK. Figure 10(c) illustrates this exchange. A leaf's packet transmission is seen as the first SFD pulse, from 4.8 to 5.4 ms, and ACK reception is the second SFD pulse. We use an 802.15.4 data packet for the ACK to allow the branch node to send additional data to the leaf, as discussed in [8]. Note that the leaf has sufficient energy to transmit its data packet and receive the data ACK packet.
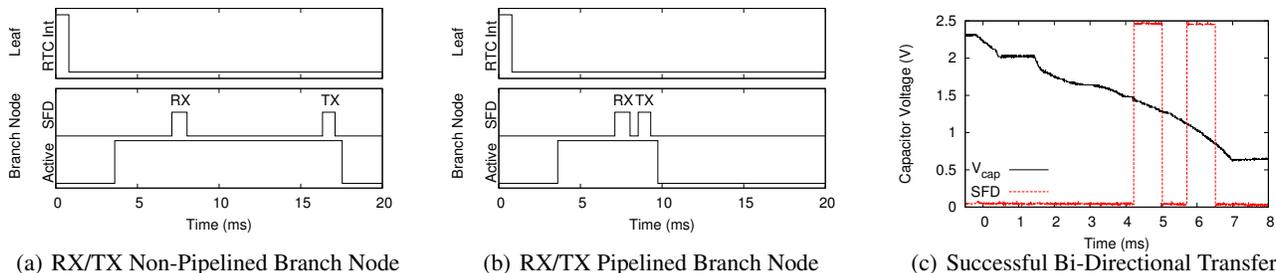
4.5 Branch to Leaf Communications

Data transmission from branch to leaf nodes is more challenging than transmissions in the other direction. A leaf node's transmission serves to deliver data, optionally request an acknowledgment, and probe for pending inbound traffic. In a typical radio stack, a received packet is buffered in the radio until it is received fully and then dispatched to higher layers for processing. A response packet may then be generated, passed through the network stack, and transferred to the radio, and finally transmitted. Unfortunately, these latencies add up, as Figure 11(a) shows, where the default TinyOS network stack takes nearly 10 ms between a leaf's transmission (the first SFD pulse) and the branch's reply (the second SFD pulse), causing a leaf to deplete its capacitor.

To enable data transmission from branch to leaf, we modify the branch to pipeline payload reception with transmit

(a) Leaf Transmissions   **(b)** Leaf-Branch Synchronization Jitter   **(c)** Leaf-to-Branch Communications
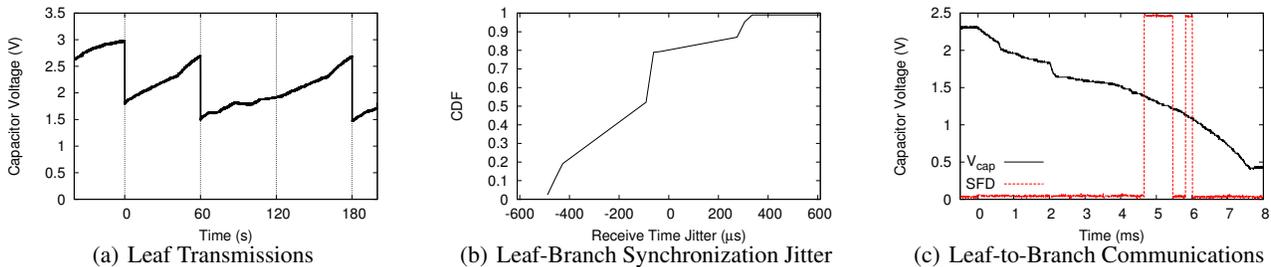
Figure 10: Transmitting packets from leaf to branch nodes. (a) A leaf node transmits a packet when it has a sufficiently high capacitor voltage (at 0 s, 60 s, and 180 s) but the leaf node skips a transmission (at 120 s) if it does not have sufficient energy. (b) A branch node is able to track the leaf node's transmission times with very low error. The cumulative distribution of branch-leaf synchronization times over a 3.5 hr time window of minute-level transmissions shows less than ±500 μs jitter. (c) A leaf node is able to successfully transmit a packet (first SFD pulse) and receive an acknowledgment (second SFD pulse) from a branch node before exhausting the limited energy in its capacitor. This demonstrates reliable transmission of data from leaf to branch nodes.

(a) RX/TX Non-Pipelined Branch Node   **(b)** RX/TX Pipelined Branch Node   **(c)** Successful Bi-Directional Transfer

Figure 11: Transmitting packets from branch to leaf nodes. (a) Leaf and branch nodes wake up triggered by their respective real-time clocks (only the leaf's RTC init is shown). A leaf transmits a probe that is received by branch. The packet transmits data to the branch and also indicates that it is awake to receive a packet. The branch responds, but the long latency in serialized packet processing – receiving, dispatching, processing, and responding – results in the leaf exhausting its energy supply before receiving the branch's transmission, resulting in communications failure (the branch does not transmit first since the common case requires the leaf to transmit sensor data). (b) Low-latency optimizations to the packet processing datapaths in the branch dramatically reduce delays and allow a fast RX/TX turnaround. The key optimization is parallelizing and pipelining the various datapaths so that writing to the TX FIFO begins before the packet has been completely read out from the RX FIFO. (c) The capacitor voltage and radio activity (SFD) of a leaf communicating with a low-latency branch node shows that the communication completes before the leaf exhausts its energy supply.

FIFO loading. This allows the branch node to reply with a full packet with a 0.67 ms turnaround time, as Figure 11(b) shows. This optimization allows the leaf node to successfully receive the branch's transmission before depleting its energy reserves, as Figure 11(c) shows. In addition to ensuring that communications from branch to leaf is viable, this changes also lower the energy cost of communication for the branch node itself, cutting the radio active time in half – from about 14 ms to 7 ms.

5. RELATED WORK

The design and implementation of standards-based, near-nanopower, wireless sensornets raises challenges across the entire system. This work complements prior efforts in sensornet architectures, platforms, power systems, discovery protocols, and synchronization primitives, and is related to ongoing work in wireless sensing and computational RFIDs.

Architectures: The canonical patch sensornet is the Great Duck Island (GDI) deployment [18]. The battery-powered nodes generated a message every 5 minutes. In this work, we show that purely energy-harvesting indoor nodes can send a message about every minute during daylight hours. Tiered architectures, similar to our trunk, branch, and leaf partitioning were proposed by Gnawali et al. in Tenet [5]. Tenet describes an architecture of unconstrained masters (trunk) and battery-powered nodes (branch). We extend this architecture one level further, into energy harvesting leaves, and describe how these leaves can interoperate with the existing architectures.

EnHANTs: [6] The EnHANTS project explores the challenges for a new tier of energy-neutral, self-reliant nodes. Their research goals are very similar to ours: build and network indoor solar-based energy-harvesting sensor nodes with severe energy, size, and cost constraints. However, the ap-

proaches differ considerably. While we use currently available parts in a small and thin form factor designed for wall mounting, their tags are custom fabricated hardware in a flexible sticker design. Both methods face two main challenges: hardware design and network communication.

For hardware, EnHANTs use Ultra-Wide Band (UWB) impulse radios with low transmit energy per bit requirements (1 nJ vs 150 nJ for current 802.15.4 radios) at the cost of range (3 m at 25 kbps). They also use organic solar cells (as opposed to our more traditional amorphous and crystalline solar cells) for physical flexibility at the cost of power transfer efficiency. UWB impulse radios and organic solar cells are still in active research and have ground-breaking future prospects. However, we show that current 802.15.4 radios and commercial solar cells are sufficient to network of small, energy-harvesting sensor nodes. Our node designs achieve similar hardware goals – namely small volume and practical efficiency – while using off-the-shelf components and offering a much lower overhead cost compared with custom silicon. Our sensor nodes can be deployed immediately for many reasonable indoor applications, and especially ones in which lighting is available when measurements are required.

For network communications, EnHANTs focuses on optimizing resource allocation – energy harvested, energy stored, and data communicated on local (per node) and global (network) scales [10]. This approach assumes that nodes are already time-slotted using some on some global time scale. More recent work focuses on synchronizing the tags to a beacon transmitting receiver (tag reader) [21]. The work does not address communication to other duty-cycled nodes nor does it address network communications with unreliable clocks or power intermittency. The work also sidesteps the issue of clocks – and especially accurate clocks – that may be too energy expensive to provide the necessary timebase, even though the work acknowledge the problem. Assuming a 40 hour work week, the cost of time-keeping requires a significant portion of the overall energy budget (40%):

P_{RTC}: 1.5 V * 1 μA = 1.5 μW
P_{leaf}: 3 V * 20 mA * 10 ms/60 s * (40/168) hrs/wk = 2.4 μW

In contrast, our work attacks the problem of energy-expensive clocks by letting nodes lose all sense of time by allowing for complete clock shutdown if energy reserves are depleted. This results in nodes churn, so robust and efficient network (re)synchronization is necessary as energy is harvested, stored, and depleted. To address this problem, hierarchical asynchronous neighbor discovery is a fundamental primitive in our system.

While EnHANTs seeks to push the boundary of low-power operation, we claim that practical energy harvesting sensor networks are viable today. We validate our claim by building and studying sensor platforms, protocols, and power systems using off-the-shelf components. We have shown that we can build energy harvesting sensor nodes that can measure, process, and transmit sensor data packets in a comparable fashion to current duty-cycled battery powered nodes. We have also shown that completely depleted and unsynchronized leaf nodes can recharge and join a network within a reasonable amount of time via asynchronous neighbor discovery. Our next challenge lies in making more robust leaf to branch discovery and ongoing communication protocols with large numbers of leaf and branch/trunk nodes. Eventually, we hope to enable leaf-to-leaf communications – a goal we share with the EnHANTs project.

Indoor Photo-Voltaic Systems TwinStar [22] is a mixed indoor-outdoor solar energy harvesting system that explores a capacitor-only energy storage design. The idea behind TwinStar is to use energy when it is available, and thus reduce energy leakage. The TwinStar platform uses two solar panels. A bootup panel triggers the DC/DC converter to solve the zero-energy bootup challenge, while a large solar panel ($\sim$36 cm^2) is used to charge the super capacitor. Our design is similar in that we use energy when it is available. However, our design is fully powered by a solar panel the size of TwinStar's bootup panel.

Raisigel et al. [17] describe an IEEE 802.15.4 compliant IPV platform. Raisigel's design uses a Schott solar panel with an active area of 8.5 cm^2, an Atmel AT86RF231 radio, and an Atmel ATMega644PV 8-bit microcontroller. The platform uses a manganese rechargeable lithium coin-cell battery and storage capacitors for peak currents. Using very shallow recharge cycles during nighttime sampling (minute intervals), they extended the life of the button cell battery beyond 10 years under nominal daytime lighting conditions. Raisigel's design focuses on ultra-low power sleep modes. Raisigel's battery/energy-harvesting hybrid sensors are fully capable sensors powered under indoor lighting conditions. Their design requires large batteries due to the shallow discharges necessitated by extending the life of the battery to 10 years. Their design is practical with batteries, however, our work explores the scenario in which complete energy loss is possible. This is a reasonable assumption for rooms with much less than 8 hours of light per day or 40 hours per week. For their design, this means that battery discharges will be far deeper and energy harvested needs far higher to replenish the battery, which affects battery aging adversely. With lower light levels, their nodes transition from energy-harvesting to battery-power. Therefore, their work demonstrates battery-powered nodes with longer life because of their hybrid approach. The work also does not consider protocol issues for duty-cycled nodes and instead assumes that an always-on receiver is available in range, thus sidestepping synchronization issues.

EnHANTs and Raisigel both explored the light intensity availabe via indoor solar. Our data also support their findings: if an office has a window, one can expect irradiance ranging from 10μW/cm^2 to 100μW/cm^2 [7].

EnOcean provides a commercial battery-less system with the STM300 and STM310 modules [4]. Both systems are powered by a solar panel of about 4.8 cm^2. The STM300 series provides a proprietary transceiver, while the STM310 is a transmitter only. The EnOcean system provides a proprietary solution for a leaf-to-trunk communications model, where the peer is an always-on node, and Raisigel's work describes an energy-harvesting platform without going into details of a system architecture. With this work, we delve deeper into the possibilities of system architectures in which leaf nodes communicate with battery-powered mesh nodes, as well as other energy-harvesting leaf nodes.

6. CONCLUSION

As the density of indoor wireless sensors increases, and their size decreases, battery-powered operation becomes increasingly less viable. Batteries have a finite lifetime, they incur replacement costs, and their average power delivery scales poorly compared with indoor photovoltaics. Hence, today many believe that energy-harvesting holds the key to long-term, cost-effective, and sustainable sensing. However, today's energy-harvesting sensors, like EnOcean and ZigBee Green Power, require that the nodes be within one hop of an always-powered base station or repeater. This regresses on a decade of advances in multihop mesh networking – and its diverse benefits – including better spatial reuse of spectrum, more path diversity, greater deployment flexibility, and lower transmission power.

This paper shows that it is possible to augment battery-powered mesh networks with energy-harvesting leaf nodes. Thus, we create a new tier of sensor nodes that are free from the constraints of battery power, but still retain the many benefits of interoperating with contemporary wireless multihop mesh networks. Furthermore, these new energy harvesting leaf nodes can be built from standard components that have been around for years. We simply combine them in new ways to achieve low-leakage operation, optimize the system software for low-latency cold boot and communications, and employ low-energy protocols to achieve synchronization and maintain communications. Using newer, and lower power, technology would only improve performance, and using tiny batteries or supercapacitors would allow continuous operation, even in the absence of harvestable energy. This work paves the way for a new tier of perpetual computing systems, shows the viability of the architectural approach, and demonstrates interoperability with existing sensor network nodes.

Acknowledgments

7. REFERENCES

[1] Cymbet Corporation. EnerChip CBC050-M8C: 50uah rechargeable solid state battery, Oct. 2011.

[2] P. Dutta and D. Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *SenSys '08: Proceedings of the 6th International Conference on Embedded Networked Sensor Systems*, pages 71–84, Nov. 2008.

[3] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler. A building block approach to sensornet systems. In *SenSys'08: Proceedings of the Sixth ACM Conference on Embedded Networked Sensor Systems*, nov 2008.

[4] EnOcean. STM310: General Purpose Sensor Transmitter Module, Apr. 2011.

[5] O. Gnawali, K. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler. The tenet architecture for tiered sensor networks. In *SenSys'06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 153–166. ACM, 2006.

[6] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman. Energy harvesting active networked tags (EnHANTs) for ubiquitous object networking. *IEEE Wireless Communications*, 17(6):18–25, 2010.

[7] M. Gorlatova, A. Wallwater, and G. Zussman. Networking ultra low power energy harvesting devices: Measurements and algorithms. In *Proceeding of IEEE INFOCOM'11*, Apr. 2011.

[8] J. W. Hui and D. E. Culler. Ip is dead, long live ip for wireless sensor networks. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 15–28, New York, NY, USA, 2008. ACM.

[9] Linear Technology. LTC3108: Ultralow voltage step-up converter and power manager, Apr. 2010.

[10] A. B. M. Gorlatova and G. Zussman. Performance evaluation of resource allocation policies for energy harvesting devices. IEEE Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'11), 2011.

[11] Maxim Integrated Products. DS32kHz: 32.768khz temperature-compensated crystal oscillator. http://datasheets.maxim-ic.com/en/ds/DS32kHz.pdf, Apr. 2010.

[12] Microchip Technology Inc. PIC16F1827: Flash mcu with nanowatt xlp, Oct. 2011.

[13] Micropelt GmbH. Micropelt Thermogenerator, Oct. 2011.

[14] NXP. PCF2127A: Integrated rtc, tcxo and quartz crystal, May 2010.

[15] Panasonic. ML621: Manganese lithium coin batteries, Oct. 2011.

[16] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks (IPSN/SPOTS)*, 2005.

[17] H. Raisigel, G. Chabanis, I. Ressejac, and M. Trouillon. Autonomous wireless sensor node for building climate conditioning application. SENSORCOMM'10, Venice/Mestre, Italy, 2010. VDE Verlag.

[18] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler. An analysis of a large scale habitat monitoring application. In *SenSys'04: Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems*, Nov. 2004.

[19] TAOS. Tsl230brd programmable light-to-frequency converters.

[20] US Dept. of Labor. Illumination regulations #1926.56 (standards - 29 cfr).

[21] Z. Wang, A. Tajer, and X. Wang. Communication of energy harvesting tags. IEEE Transactions on Communications.

[22] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang. Leakage-aware energy synchronization for wireless sensor networks. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 319–332, New York, NY, USA, 2009. ACM.

[23] ZigBee Alliance. New ZigBee green power feature set revealed, June 2009.