Deltaflow: Submetering by Synthesizing Uncalibrated Pulse Sensor Streams

Meghan Clark, Bradford Campbell, and Prabal Dutta
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI 48109
{mclarkk,bradjc,prabal}@umich.edu

ABSTRACT

Current submetering systems suffer from prohibitive device costs, invasive installations, and burdensome maintenance. In this paper we present Deltaflow, a submetering system that can estimate the power draw of individual loads by augmenting aggregate measurements with very simple sensors. The key insight is that we can drastically reduce sensor complexity by encoding information in the mere existence of a radio transmission, rather than the contents of that transmission. A sensor consisting simply of a radio and an energy-harvesting power supply tuned to harvest a side-channel emission of energy consumption (e.g. light, heat, magnetic field, vibration) will exhibit an activation frequency that is correlated with the power draw of the load to which it is affixed. These sensors report their activations to the data-processing backend, which can determine the actual power draw by incorporating ground truth aggregate measurements such as those provided by utility meters. The server maps sensor activations to energy consumption by observing when the aggregate measurement and the sensor activation frequency change simultaneously. The server iteratively partitions the system history into discrete states which are used to construct and solve instances of a linear optimization problem. Solutions to the problem reveal the mapping from pulse frequencies to individual load power draw. This systems approach to submetering results in deployments that are easy to install and maintain, while contributing zero additional load, enabling building owners and occupants to simply affix tags to energy consumers and automatically begin receiving real-time power draw readings.

Categories and Subject Descriptors

B.m [**HARDWARE**]: Miscellaneous—*Miscellaneous*

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Energy harvesting, Power metering, Data aggregation, Intermittent power, Submetering, Non-intrusive load monitoring

e-Energy'14, June 11–13, 2014, Cambridge, UK.
ACM 978-1-4503-2819-7/14/06.
http://dx.doi.org/10.1145/2602044.2602070.

1. INTRODUCTION

Buildings account for a significant share of resource use in the U.S.: they consume 39% of the energy, 73% of the electricity, 55% of the natural gas, and 12% of the water. Powering them results in $400 billion in annual expenditures. If current trends continue, by 2025 buildings worldwide will consume more energy than the transportation and industry sectors combined. Moreover, "between 60 and 80 percent of the energy used in commercial office buildings is consumed by tenants within their spaces," yet, "no widely available tools exist to help tenants understand their energy consumption or to compare it against their peer groups," claims a recent U.S. National Science and Technology Council (NSTC) study [17].

The same NSTC study states that the "refined measurement of electricity and water use represent a key enabler for the improved performance of new and existing buildings," and adds that, "For building operators, a detailed record of system performance provides a critical means of. . . focusing future design and retrofit activities on the most cost-effective energy and water system improvements," and, "For building occupants, detailed information on consumption promotes resource conservation through behavioral changes." [17] A recent U.S. National Science Board (NSB) study adds, "One critical focus area is developing measurement science to enable the development of zero-energy buildings." [16]

Unfortunately, today's metering technologies are ill-suited to the measurement task at hand. Existing whole-building metering systems provide little visibility into the contributions of individual loads. Non-intrusive load monitoring (NILM) techniques provide some additional insights, but they do not scale beyond a few loads, they often require training or association of loads to their signatures, and they can typically identify only the largest loads. Plug-load meters provide much greater visibility but they are costly and limited in their coverage to easily accessible electrical loads. Hybrid approaches also exist that augment NILM with additional sensors that aid disaggregation, but the devices are often costly, the sensors require periodic battery maintenance, and the algorithms often assume appliance states are known *a priori*.

To address the challenges with prior approaches, this paper proposes not to replace, but rather to augment, existing whole-building or panel-level metering techniques with a new class of simple and easily-deployed energy-harvesting sensors and a novel algorithm that combines data from both new and existing meters to infer the contributions from individual loads. We envision a future in which building owners or occupants can simply and directly tag end loads like a ceiling light, shower head, or range top with small and inexpensive sensors. The sensors *indirectly* monitor the load by harvesting the energy that a load emits when operating (e.g. the light emitted from a bulb). The sensor activations are received by a base station, time-stamped, and forwarded to the cloud for processing.

Our prior work has shown the viability of energy harvesting energy meters [4]. In particular, we have shown that the energy emitted from many electrical loads is roughly proportional to the energy consumed by that load, and that by harvesting this side-channel energy, it is possible to intermittently power sensors whose activation interval (and in turn packet transmission period) approximately encodes the underlying energy use. We call this the "Monjolo" principle and envision many different kinds of "pulse" sensors that could be constructed using this principle.

Such sensors represent a modern reinterpretation of the decades-old pulsed-output meters that measure electricity, gas, and water, but with one crucial difference. Whereas traditional meters are calibrated to output one pulse per useful billing quanta (e.g. 1 W-hr, 1 CCF, or 1 gal), our proposed meters would usually not be, and often could not be, calibrated *a priori*. We hypothesize that by using statistical methods to correlate the timing of the packet pulse train from instrumented loads with readings from existing aggregate electricity, gas, or water meters, we can obtain individual estimates without intrusive metering.

In an ideal measurement setting, all loads would be instrumented and energy-harvesting sensors would respond instantly to changes in load power draws. However, satisfying these conditions is usually undesirable for reasons of cost and coverage, and potentially impossible for energy-harvesting sensors that must accrue energy for some time before becoming active. To mitigate the effects of delayed sensor response, we employ algorithmic techniques to partition the power draw history into periods of rapid change and relative stability. Then, by analyzing the sensor pulse trains, we identify those adjacent periods in which it is likely that only the monitored loads changed, discarding those changes attributable to uninstrumented loads. Representative data points for the remaining stable periods are then used to construct a regression problem that determines the sensor calibration. This allows us to deploy uncalibrated sensors, which reduces cost, and then employ parameter estimation to calibrate the sensors *in situ*.

While today's centimeter-scale prototype sensors are built from off-the-shelf parts, we imagine that in the future the sensors will be built from integrated circuit technology laminated into smart labels with energy-harvesting front-ends, eventually making them small, inexpensive, and easy-to-deploy. Indeed, contemporaneous research has already demonstrated the viability of fully-functional sensor systems at the millimeter-scale [13]. Our approach imposes minimal sensor requirements, allowing the future sensors to be drop-in replacements. We believe that one of Deltaflow's greatest strengths is sensor agnosticism. Since the disaggregation algorithm simply analyzes uncalibrated pulse trains, pulse counting serves as a layer of abstraction over the physical details of how the pulse measurement is generated. This means the system naturally supports a heterogeneous collection of pulse-counting devices. It makes no difference whether the devices measure electricity, water, or gas–or even whether the measurements are aggregated at the level of plug loads, individual circuits, panel boxes, whole buildings, or entire campuses–the same techniques apply. Collectively, these Deltaflow properties address cost and coverage challenges, and enable scalable deployment and widespread adoption.

To evaluate the viability and drawbacks of this approach, we employ several types of sensors in various configurations with real loads. We find that our pulse-counter calibration methods are able to disaggregate power draw in cases with complete sensor coverage, with unmonitored loads, and with heterogeneous sensor types. Further, we show that our approach is able to provide breakdowns of total energy usage suitable for solving the submetering problem. Finally, we identify limitations and opportunities for future work.

2. RELATED WORK

Numerous methods exist to measure and disaggregate electrical loads in residential and commercial buildings including non-intrusive load monitoring, plug-load monitoring, and hybrid approaches. In this section, we discuss these approaches, identify their drawbacks, and contrast them with Deltaflow.

2.1 Non-Intrusive Load Monitoring

Whole-building meters [5, 25] provide an overall view of electricity or water use, but they do not disaggregate the data in a way that allows either precise or approximate attribution to the various individual loads comprising the whole. Analytics can take meter data and disaggregate the readings using appliance signatures with a technique called non-intrusive load monitoring or NILM [8]. This approach works well when the loads are sufficiently few (e.g. typically no more than 6-7), mostly large (e.g., air conditioners and stoves), and have distinctive signatures (e.g. refrigerators and ovens). NILM has difficulty with smaller loads (e.g. electronics) or multiple instances of a particular load (e.g. several 60 W light bulbs), or a large collection of loads (e.g. in an office building).

Extensions of the general NILM approach use higher frequency sampling (e.g. MHz) of the current and voltage waveforms, higher dimensional data (e.g. real and reactive power), and complex signatures (e.g. wideband spectra arising from the flick of a switch or a toggle of a faucet) to identity individual loads [2, 6, 7, 19], but these approaches still require training to associate the loads to their signatures, are susceptible to small changes in the environment (e.g. moving a load from one outlet to another), and are costly due to the use of high-rate sampling and processing. In contrast with NILM techniques, our approach can scale to a much larger number of loads of any type, even if they have similar or identical signatures. By instrumenting individual loads with inexpensive, indirect, and inaccurate sensors, Deltaflow can distinguish identical loads and disaggregate the contributions of individual loads.

2.2 Plug-Load Metering

At the other extreme, plug load meters allow individual loads to be measured [9, 14, 18, 20, 22, 24, 26]. Standalone meters display usage data locally, which supports casual use but not automated aggregation and analysis. Networked energy meters send their data to servers for analysis and visualization. However, plug load metering faces some coverage and cost disadvantages. Some loads are built-in or hard-to-access, including ceiling lights, HVAC equipment, and major appliances, making them ill-suited to such meters (although NILM techniques can sometimes identify these loads). Furthermore, at a price point of $25-$50 for standalone meters and $75-$250 for networked meters, covering a home or office can cost thousands or tens of thousands of dollars, making widespread monitoring prohibitively expensive.

In contrast with intrusive plug load meters, our approach supports indirect measurements of a load—like a ceiling light's radiant output or an air conditioner's vibrations—rather than their electrical inputs, which makes instrumenting built-in or difficult-to-access loads easier. In addition, the kinds of indirect sensors we envision—that harvest energy from light, heat, magnetic fields, and vibrations—could be constructed at the chip-scale and built into "peel-and-stick" sensor tags, much like RFID chips today. Although our mesoscale sensors deployed in modest numbers today are centimeter-scale systems, prototypes of some of the basic sensors, complete with an energy-harvesting front end, energy storage, and a processor and a radio, have been demonstrated at the millimeter-scale [13], providing some evidence that peel-and-stick sensors may soon be viable and inexpensive.

2.3 Hybrid Approaches

Due to the drawbacks of both the NILM and direct metering approaches, some recent efforts have explored hybrid models in which additional sensors augment NILM and aid with disaggregation [11, 12, 15, 23]. The extra sensors help NILM scale beyond a half-dozen loads by providing it an additional signal that reflects the state of an individual load. Sensors that detect the on-off states [10] or more finely quantized energy emissions of appliances including light, sound, and magnetism [21], have been shown to aid greatly in disaggregation. However, the former approach requires foreknowledge of the states of the instrumented loads, and the latter approach requires modeling the physical transfer functions for each type of energy emission sensor. Additionally, a major scalability impediment of earlier hybrid approaches is the cost of the sensors and the overhead of periodic battery replacement. Low-cost, mains-powered sensors that detect appliance state can address these problems, but they are currently limited to electrical plug loads [27].

In contrast with prior work that attempts to directly measure side-channel emissions (including ones that harvest the side-channel energy and use it to power active sensors [1, 3, 15]), or inexpensively measure plug loads, we propose to leverage emerging "Monjolo" sensors that simply harvest the side-channel energy and transmit a radio packet when enough energy has been accumulated to do so [4, 28], making the activation rate a proxy for power. Harvesting just enough to send a packet requires less energy than revenue-grade sensing and is easier to install, thus enabling smaller and less expensive—but also cruder—sensors. Such simple sensors, while easier to deploy and operate, are not easily calibrated.

However, much of what the sensors lose in individual quality, they gain through sheer numbers in conjunction with data fusion and optimization algorithms. Like the NILM and hybrid disaggregation techniques, Deltaflow uses a whole-building meter (or, in the case of large buildings, unit-, zone-, or floor-level meters). In our model, data from the whole-building meter is combined with the activation frequency data from a multitude of inexpensive energy-harvesting sensors to infer the contributions of individual loads. We model the relationship between a sensor's activation rate and the underlying energy flow using a combination of time series partitioning heuristics and linear optimization techniques.

3. OVERVIEW

The goal of the Deltaflow system is to provide a breakdown of the total electrical energy consumption of a building at the individual load level. Deltaflow accomplishes this using a calibrated, accurate aggregate meter and an array of simple, pulse-based sensors attached to each load. Each pulse sensor operates according to a simple principle: the higher the power draw of the load, the greater the frequency of the pulses. The Deltaflow system uses these pulse streams as hints about each load to disaggregate the aggregate measurement into each individual load's contribution.

The Deltaflow system architecture is shown in Figure 1. A power meter that is monitoring aggregate energy flow upstream from the target loads (e.g., a utility meter) reports aggregate measurements to the Deltaflow server. Additionally, the individual loads to be metered are instrumented with a suitable pulse sensor that transmits a representative pulse stream to the Deltaflow server. By augmenting the aggregate measurements with pulse frequencies that change as the individual load's power draw changes, the Deltaflow server characterizes the sensor's response to changes in the power state of the load it is attached to. The server uses these models of the sensors to perform calibrated disaggregation and provide power draw estimates for the individual loads being monitored.

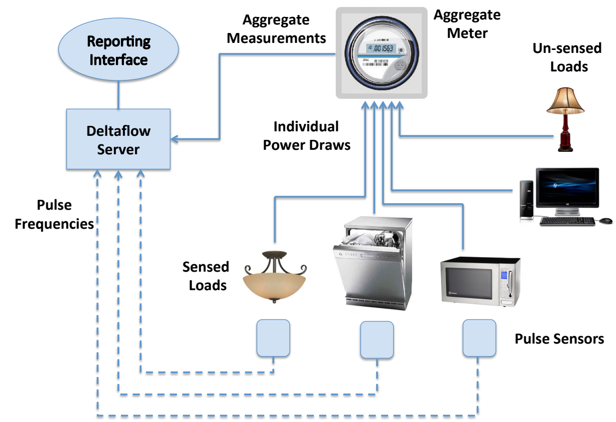


Figure 1: System architecture. The Deltaflow server takes in aggregate power draw measurements and the activation frequencies of energy-harvesting sensors attached to individual loads. By augmenting aggregate power measurements with sensor activation (or pulse) frequencies that are correlated with power draw, Deltaflow determines the individual energy consumption of the sensed loads. The system is able to function even if some loads remain unsensed.

4. DESIGN

In this section we describe common types of aggregate meters, the pulse sensors we use to monitor loads, and the algorithm used by the Deltaflow server to disaggregate loads.

4.1 Aggregate Meters

Aggregate meters provide a calibrated stream of ground truth measurements that represent the sum of all of the individual loads or consumers in the system or subsystem. Common household aggregate meters exist for electricity, water, and natural gas. In some cases unit-, zone-, or floor-level aggregates may be more appropriate than whole-building meters. Currently, while many utility meter readings are difficult to access and provide a temporal resolution that is quite coarse, being revenue-grade devices, they are quite accurate. Trends suggest that in the future detailed readings may become more readily accessible. In the meantime, commercial whole-house meters that provide higher temporal resolution, like The Energy Detective [5], are available on the market.

4.2 Energy-Harvesting Sensors

The pulse sensors the Deltaflow system uses are based around a simple observation: the act of energy consumption often emits side channels of energy that can be harvested to intermittently power a sensor node. For example, powering an AC load creates a changing magnetic field around the wires running to the load, lighting a room generates a harvestable light source, and drawing a hot bath causes pipes to be warmer than the surrounding air. Capturing these side-channel energy sources to power a sensor node creates a suitable pulse sensor that follows the principle in Section 3. As the power draw of the energy-consuming load increases, so does the magnitude of the side-channel emission, and this in turn causes the sensor node to activate more frequently. These activations are the "pulses" from the pulse sensors. Even though we employ energy-harvesting sensors to generate these pulses, any meter that operates on this principle (including most utility meters in existence today) will satisfy our requirements.

(a) Coilcube (b) Split-core (c) Gecko

Figure 2: Pulse sensors. Examples of the energy-harvesting pulse sensors we use attached to the loads they meter. Monitored loads include a plug-load, a circuit in a panel box, and a recessed can light. The latter two loads are easily monitored using our approach.

The Deltaflow pulse sensors are based on the Monjolo principle [4], a simple energy-harvesting design that entails three basic components: a harvester, a processor with non-volatile storage, and a radio. The energy-harvesting power supply accumulates charge over time and activates the computational core of the node when enough energy has accrued. Upon activation, the node transmits a wireless packet, discharges any remaining energy, and resumes charging once again. The node also stores the activation count to stable storage and transmits this information in each packet. The Deltaflow server receives these packets and interprets each packet as one (or more) pulse(s) for disaggregation purposes.

While simple, this design offers two major advantages: it requires no batteries and can be used to measure any energy source that can be harvested. Removing batteries simplifies deployment and eliminates the maintenance cost of replacing them. Harvesting side-channel emissions of power draw, such as magnetic inductance, heat, light, or vibration enables sensing of otherwise difficult-to-measure energy sources, such as ceiling lights, shower heads, or built-in appliances. Additionally, these sensors abstract the heterogeneity of their power sources and sensor types by eschewing direct measurements in exchange for a single homogeneous interface—pulse rate as proxied by activations or radio packet transmissions.

We adjust our specific implementation to address two common network problems: packet loss and flooding. First, because radio packets can be dropped, each sensor keeps a local counter of its own activations in nonvolatile memory and transmits this count every time it sends a packet. This way, the receiver can still calculate the rate of activations even if a packet is lost. Second, if one or more of the sensors has a very high activation rate it could easily flood the wireless channel by sending packets at every activation. To prevent this, we use a "timing" capacitor with a large resistor in parallel. At every activation, the sensor checks the voltage on the capacitor. If it is below a threshold, the node transmits and then recharges the timing capacitor. If it is above a threshold, the node simply increments the counter and waits for its next activation. The following sections describe the specific devices we employ as pulse sensors for three representative load classes.

4.2.1 AC Power Meter

In order to harvest from an AC power source, we use a current transformer. The magnetic field of the AC line induces a current in the transformer and the output of the transformer is then rectified and harvested. We call the AC power pulse sensor Coilcube, which comes in two forms. The plug-load (Figure 2(a)) version sits between the wall outlet and the load, and contains the phase line leading to the AC load wrapped around a current transformer. The coil, the harvesting power supply, and the sensor node core fit in a small enclosure into which the load is plugged. Although

Figure 3: Recessed ceiling lights. Lights such as these are traditionally difficult to meter due to limited access to their wiring and uncertainty about the topology of the lighting circuit. Our Gecko sensors are easily and unobtrusively installed in fixtures such as these. See Figure 2(c) for a closeup view of the sensor mounting.

it might seem odd to design a plug-load pulse sensor, it allows us to both illustrate the viability of the approach and deploy a near-zero power sensor [4]. The second version uses a split-core current transformer that can be clipped easily around a wire running to a breaker in a circuit panel (Figure 2(b)). The other electronics are small and can be installed with the transformer in the panel box. This design is ideal for unobtrusively monitoring an entire circuit.

4.2.2 Light Meter

The light meter, called Gecko, enables Deltaflow to measure the energy contribution of ceiling lights, such as those in Figure 3 that are otherwise challenging to measure. Harvesting is based on a small amorphous solar cell. All of the other required electronics fit in a space approximately the same size as the solar cell. This results in a small sensor that can easily be deployed near a light bulb to unobtrusively meter the light as shown in Figure 2(c).

4.3 Disaggregation Methodology

As Figure 1 shows, the Deltaflow server receives two input streams: aggregate ground truth measurements (e.g. Watt or Whr) and pulse frequencies from each sensor attached to the instrumented loads. The server uses each of the pulse streams to decompose the aggregate measurements into the contributions from each load.

4.3.1 Sensor Calibration

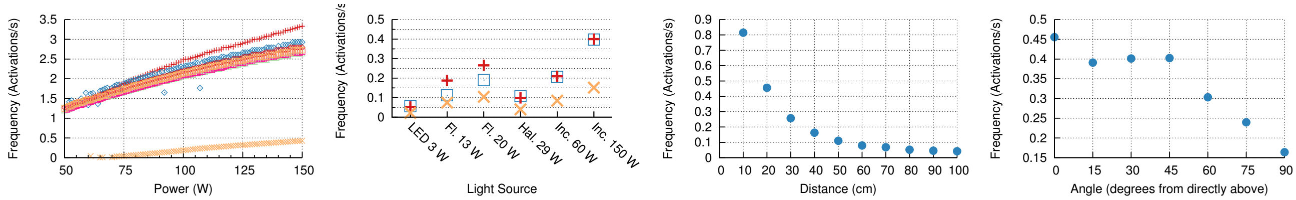
The server starts by trying to estimate a *calibration function* for each sensor. This calibration function must be determined at runtime, and not *a priori*, for two main reasons: manufacturing differences, as shown in Figures 4(a) and 4(b), and sensor placement, which is unknown until after installation yet affects the response of the light-based pulse meters, as shown in Figures 4(c) and 4(d). We determine the sensor specific calibration function by first noting that the sensor pulse frequency, s_i, of load i is a function of the power draw of load i

$$s_i = f_i(p_i) \tag{1}$$

Examples of this relationship are shown for Coilcube sensors in Figures 5(a) and 5(b) and for Gecko sensors in Figures 5(c) and 5(d). Given that we know s_i and want to determine p_i, we can express the relationship as:

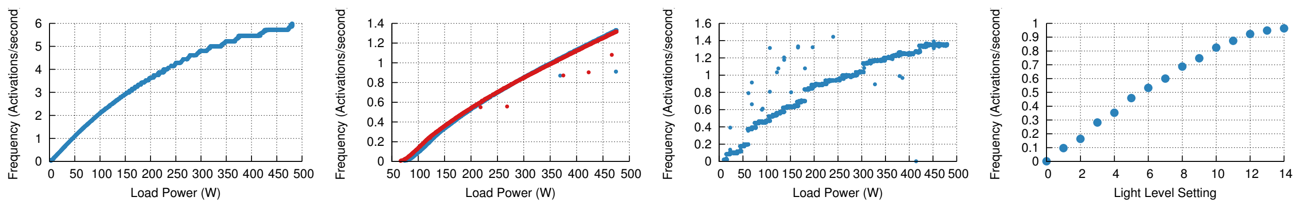
$$p_i = f_i^{-1}(s_i) = g_i(s_i) \tag{2}$$

Then, given a pulse rate, we can determine a power estimate for each sensed load using the calibration function g_i.



(a) Coilcube Variation (b) Gecko Variation (c) Gecko Distance Dependence (d) Gecko Angle Dependence

Figure 4: Motivation for runtime calibration. Figures 4(a) and 4(b) show that the pulse rate of different instances of the same sensor type varies over the same range of loads. This suggests that a single calibration function cannot be used for all instances of a particular sensor type. Figures 4(c) and 4(d) show the effect of placement on the pulse rate of Gecko sensors. As the distance and angle from the light source varies, so does the pulse rate. Therefore, factory calibration is inadequate; calibration must be performed after installation.



(a) Coilcube (b) Split-core (c) Gecko Incandescent (d) Gecko Fluorescent

Figure 5: Pulse sensor activation rate vs load power. The activation rate of Coilcube, Split-core, and Gecko sensors is shown as the primary load is swept across a range of values. All sensors demonstrate a roughly monotonically increasing relationship with primary load power, but some noisy outliers are clearly visible as well.

To determine each g_i we use information about how the pulse streams change when the aggregate measurements change. However, to avoid searching through the infinite space of possible functions, we choose a general form of g_i with which to work. We additionally assume that the general form will be the same for every instance of a particular type of sensor. Requiring foreknowledge of the function for each class of sensor is reasonable because it can be determined once, at design time. This shifts the burden of configuration from sensor users to manufacturers.

We use datasets like those in Figure 5 to estimate the sensor calibration function. We find that the best-fit functions are second-order polynomials, and therefore make the assumption that the calibration function can be approximated by a monotonically increasing polynomial of degree two or less. Thus Equation 2 becomes:

$$p_i = \alpha_i s_i^2 + \beta_i s_i + \gamma_i \tag{3}$$

Now, to determine the calibration function, Deltaflow need only choose the coefficient parameters α_i, β_i, and γ_i for each sensor that best fit with historical data.

4.3.2 *Full Coverage*

To determine the polynomial coefficients, we start with the simplest case: assume all n loads comprising the aggregate are metered. If we let M_t be the ground truth aggregate meter reading at time t and let $s_{i,t}$ be the pulse frequency for sensor i attached to load i at time t, then we have the relationship:

$$M_t = \sum_{i=1}^{n} p_i(s_{i,t}) \tag{4}$$

Using the least squares method, let $\mathbf{M}$ be a vector of aggregate measurements M_t taken at different times t. Let $\mathbf{A}$ be a matrix of calibration function terms derived from pulse frequencies, where each row $\mathbf{A}_i$ is a vector of all the terms in each sensor's calibration function at time t, stripped of their coefficients.

$$\mathbf{A}_i = \begin{bmatrix} s_{1,t}^2 & s_{1,t} & 1 & s_{2,t}^2 & s_{2,t} & 1 & \dots & s_{n,t}^2 & s_{n,t} & 1 \end{bmatrix} \tag{5}$$

Let $\overrightarrow{x}$ be the vector of coefficients where $x_{i,j}$ is the coefficient of the jth term in the calibration function for sensor i.

$$\overrightarrow{x} = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & \dots & \alpha_n & \beta_n & \gamma_n \end{bmatrix} \tag{6}$$

Given data points for aggregate power draw and sensor pulse frequencies from some m different points in time, Equation 4 can be written in matrix form as:

$$\mathbf{M} = \mathbf{A}\overrightarrow{x} \tag{7}$$

$$\mathbf{M} = \begin{bmatrix} s_{1,t_1}^2 & s_{1,t_1} & 1 & \dots & s_{n,t_1}^2 & s_{n,t_1} & 1 \\ s_{1,t_2}^2 & s_{1,t_2} & 1 & \dots & s_{n,t_2}^2 & s_{n,t_2} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ s_{1,t_m}^2 & s_{1,t_m} & 1 & \dots & s_{n,t_m}^2 & s_{n,t_m} & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \\ \vdots \\ \alpha_n \\ \beta_n \\ \gamma_n \end{bmatrix} \tag{8}$$

With these definitions, we can formulate our optimization problem as:

$$\min_{\overrightarrow{x}} \quad ||\mathbf{M} - \mathbf{A}\overrightarrow{x}||_2^2 \tag{9}$$

The solution to this problem is the value of $\overrightarrow{x}$—i.e. the set of coefficients for our calibration functions—that gives load estimates that best match our aggregate readings.

Note that while our calibration functions may be non-linear, by providing the values of each non-linear term we have reduced calibration to a linear least squares problem. Use of the l_2 norm makes the regression more susceptible to noisy outliers than the l_1 norm, but it is much easier to compute. Outlier identification techniques can be used if the system converges to an unsatisfactory result.

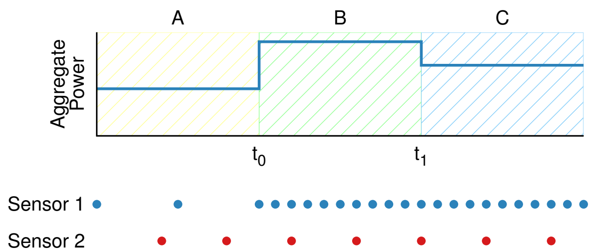


Figure 6: Effect of hidden loads. The change in aggregate power at time t_0 is attributed to the load metered by Sensor 1 whose frequency changes at that time. The aggregate also changes at t_1, but since neither sensor changes, we attribute the change at t_1 to a hidden load and ignore state C. This allows us to discard the effects of non-aliasing uninstrumented hidden loads.

4.3.3 Partial Coverage

The above approach describes finding the calibration coefficients in the ideal case when all loads are instrumented. Unfortunately, that approach does not work if even a single load is left uninstrumented, as Equation 4 no longer holds. The aggregate M_t now becomes the sum of both sensed loads and *hidden loads*.

$$M_t = \sum_{i=1}^{n} p_i(s_{i,t}) + \sum_{i=1}^{k} h_{i,t} \tag{10}$$

Let us assume that there are no aliasing hidden loads—that is, no hidden load changes at the same time as a sensed load. In that case, we can choose to select just those data points that represent states where sensed loads changed. Figure 6 illustrates such a situation. At time t_0 the pulse frequency of Sensor 1 changes along with the aggregate. However, at time t_1, the aggregate changes but both Sensor 1 and Sensor 2's frequencies stay the same. We attribute the change in aggregate power to a hidden load and ignore state C. We examine aliasing loads further in the discussion section.

For a data set in which between any two times $t = 0$ and $t = 1$ only sensed loads change, the following holds true:

$$M_1 - M_0 = \left(\sum_{i=1}^{n} p_i(s_{i,1}) + \sum_{i=1}^{k} h_{i,0}\right) - \left(\sum_{i=1}^{n} p_i(s_{i,0}) + \sum_{i=1}^{k} h_{i,0}\right)$$

$$\Delta M_t = \sum_{i=1}^{n} p_i(s_{i,1}) - p_i(s_{i,0})$$

$$\Delta M_t = \sum_{i=1}^{n} (\alpha_i s_{i,1}^2 + \beta_i s_{i,1} + \gamma_i) - (\alpha_i s_{i,0}^2 + \beta_i s_{i,0} + \gamma_i)$$

$$\Delta M_t = \sum_{i=1}^{n} \alpha_i (s_{i,1}^2 - s_{i,0}^2) + \beta_i (s_{i,1} - s_{i,0}) \tag{11}$$

When calculating the change in estimated load power draw, the constant term γ_i drops out of the model. However, due to the energy-harvesting nature of our pulse sensors, we know that when the load power draw p_i is zero, the pulse rate s_i is also zero. This means that γ_i must be zero as well for all i. Our calibration function is now of the form:

$$p_i = \alpha_i s_i^2 + \beta_i s_i \tag{12}$$

If we define a new matrix $\Delta\mathbf{A}$ such that

$$\Delta\mathbf{A}_i = \begin{bmatrix} (s_{1,t_1}^2 - s_{1,t_0}^2) & (s_{1,t_1} - s_{1,t_0}) & \dots \end{bmatrix} \tag{13}$$

then

$$\Delta\mathbf{A}\vec{x} = \Delta\mathbf{M} \tag{14}$$

Solving the optimization problem

$$\min_{\vec{x}} \quad ||\Delta\mathbf{M} - \Delta\mathbf{A}\vec{x}||_2^2 \tag{15}$$

will give us the coefficients for all of the calibration functions. Many existing libraries provide optimizers that can easily solve optimization problems of this formulation. In our implementation we used the non-negative linear least squares function provided by Python's SciPy library.

4.3.4 State Identification

The data points used to construct the optimization problem critically determine the performance of the regression. For example, when a load changes power states, the aggregate may reflect the change sooner than the pulse sensor which only activates once enough energy has accrued to power the sensor, thus causing a synchronization error. We find that identifying steady states in the data (using heuristic thresholds) and selecting a median or mean data point to represent the state compensates for this error. These data points are then used to determine the deltas between distinct adjacent stable states when constructing the calibration Equation 14.

5. EVALUATION

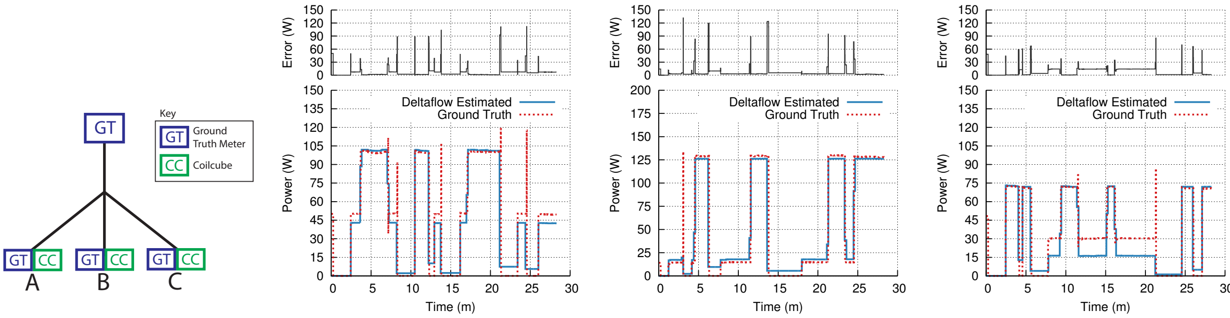
In this section, we evaluate Deltaflow's ability to identify the individual contributions of multiple loads comprising an aggregate under several conditions: when all loads are instrumented, in the presence of uninstrumented (hidden) loads, when monitoring loads with non-unity power factors, and when monitoring a load tree as one might find in a home.

5.1 Methodology

To evaluate Deltaflow we conduct several experiments consisting of a set of loads designed to test Deltaflow in different conditions. In each test, each load is monitored by a pulse sensor and a calibrated meter to collect ground truth. All loads are additionally plugged into a common calibrated meter to obtain aggregate measurements. Each load is operated to expose its power states and to simulate normal usage. The resulting ground truth and pulse data streams are collected during the experiment and then saved to be processed offline. The processing step removes the power overhead of the ground truth meters, runs the disaggregation algorithms, and computes statistics about the estimations including absolute and percentage error.

5.2 Full Instrumentation of Loads

We evaluate Deltaflow's baseline performance with an experiment in which all loads in the system are metered with pulse sensors. Figure 7(a) shows the setup of the three loads, all light bulbs with four states each. The loads are switched between states at random intervals in such a way that all state combinations are visited. Figures 7(b) to 7(d) show Deltaflow's disaggregation. Deltaflow is able to identify transitions and steady states of the loads to determine the calibration function for each pulse sensor. It then applies that calibration function to the subsequently observed pulses. In the steady state for each load, the absolute error in Deltaflow's estimates is no greater than 15 W, with average error of 5.9 W, 6.3 W, and 7.5 W, and average percent error of 12.3%, 12.3%, and 11.5%, for Loads A, B, and C, respectively. When a load transitions, Deltaflow identifies the transition and then updates its estimate, while momentarily exhibiting a higher estimation error.



(a) Load Layout (b) Load A (c) Load B (d) Load C

Figure 7: Deltaflow operating on stateful loads with instrumentation of all loads. Three loads are metered as shown in Figure 7(a). Figures 7(b) to 7(d) show the ground truth for each load and the resulting Deltaflow estimates, with graphs of the absolute error above. In steady state the error is ≤ 15 W with the only significant errors occurring during load state transitions.

The error spikes are due to the fact that Deltaflow's accuracy is constrained by the update latency of the pulse sensors. These sensors may not report a change in pulse frequency instantly, which in turn results in a race condition: aggregate measurements arrive before the pulse sensors react, causing Deltaflow to exhibit an estimation lag, which results in short periods of high error until the pulse sensors "catch up."

This experiment demonstrates Deltaflow's ability to disaggregate within reasonable error bounds. While not sufficient for scientific or revenue-grade metering, Deltaflow supports the goal of submetering loads and providing information about the energy consumption contribution of loads in a building. The delay surrounding transitions could be addressed with minor modifications to the pulse sensors themselves. Reconfiguring the pulse meters to transmit a pulse immediately after a sudden change in load power would enable Deltaflow to respond to changes quicker.

5.3 Selective Instrumentation of One Load

To explore selective instrumentation—for example, when one might want to instrument only one or just a few loads—we evaluate the case when not all loads are metered. We run the same experiment as in Section 5.2 but ignore the pulse data from the sensors attached to loads A and C. Figure 8 shows the ground truth for the single metered load and the power estimate curve from Deltaflow overlaid on the aggregate power trace for the three loads.

Even in the presence of transitions in the aggregate with no matching change in any pulse stream, Deltaflow is able to provide a power estimate for the sensed load comparable with the estimate provided in the case of full coverage. With one metered load, the average error of the power estimates is 12.5%, marginally worse than the case with full coverage, which exhibits a 12.3% estimation error.

The calibration function that Deltaflow estimates compensates for the sensed load and hidden loads changing at the same time. At $t = 7.5$ in Figure 8, the aggregate increases by 45 W while the sensed load increases only 14 W. Deltaflow correctly does not attribute the entire increase to the sensed load.

5.4 Power Factor Effects

As observed in prior work [4], Coilcube sensors have a different relationship between load power and pulse frequency depending on a load's power factor. To investigate how well Deltaflow handles loads with non-unity power factors, we construct the following setup: two loads with very similar power draws but with different power factors are run independently with the same Coilcube sensor. The first load, an AC fan, draws approximately 11.1 W with a power factor of 0.6. The second load, an incandescent light bulb, draws 10.8 W with a power factor of 1.0. The power factor of the two loads and the activation rate of the sensor are shown in Figure 9(a). While the power draws are very similar, the different power factors cause the same sensor to activate at different rates.

Deltaflow adapts to the different activation rates and generates different calibration functions for the AC fan and light bulb as shown in Figure 9(b). This demonstrates that Deltaflow can adapt to the difference in power factor. Figure 9(c) shows the resulting power estimates for each load. The estimates for the AC fan and light bulb differ from ground truth by 6.0% and 6.9%, respectively. From two different activation rates from the same sensor, Deltaflow accurately maps the rates to the correct power magnitudes by adapting to the load.

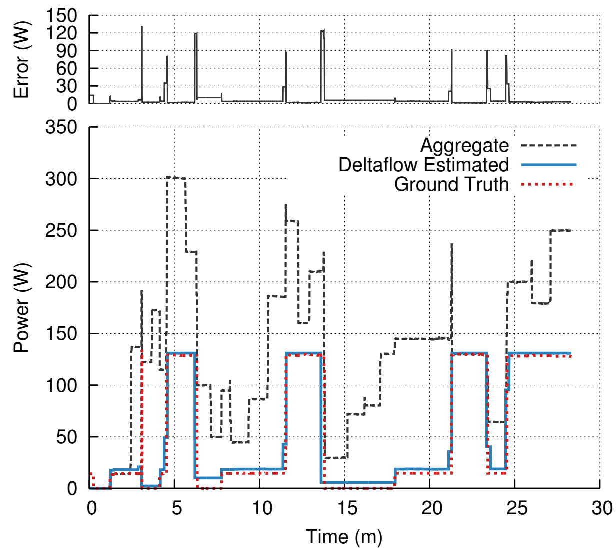
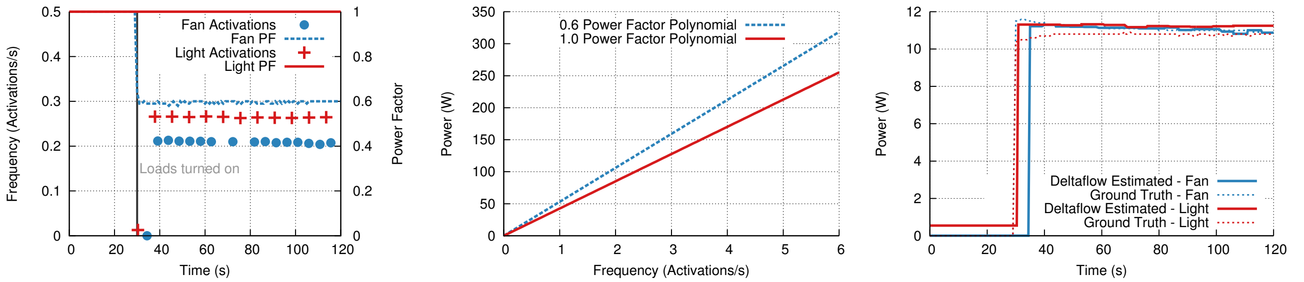
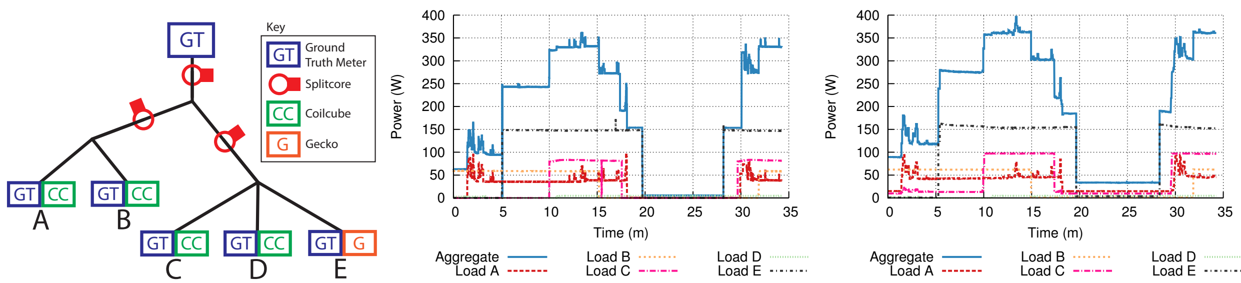


Figure 8: Deltaflow monitoring one sensed load and multiple hidden loads. This figure shows the ground truth and Deltaflow estimated power for the single sensed load, as well as the ground truth aggregate for a tree containing the sensed load and two unmonitored (hidden) loads. In steady state, Deltaflow tracks ground truth reliably, with a slight overestimate. Significant error occurs in the transitions until the sensor frequency stabilizes and Deltaflow is able to correct. Deltaflow successfully ignores state transitions that belong to hidden loads.



(a) Ground Truth (b) Deltaflow Polynomials (c) Deltaflow Estimates

Figure 9: Effect of different power factors on the Deltaflow system. Two loads, an AC fan and an incandescent light bulb, with very similar power draws but different power factors are measured. Figure 9(a) shows the power factor of each load over time and the corresponding activation frequency when the same Coilcube sensor is attached to each load. Although the power draw is the same for both loads, the different power factors cause the sensor to report different activation frequencies. Figure 9(b) shows the polynomials that Deltaflow generates for each of the loads. These polynomials differ due to the activation rate differences. Figure 9(c) shows the estimated power draw matches ground truth for each load even though the same Coilcube generates different activation frequencies when measuring the same power draw.



(a) Tree Layout (b) Ground Truth (c) Deltaflow Estimates

Load	Ground Truth (W·h)	Estimated (W·h)	Error (%)	Ground Truth (%)	Estimated (%)	Error (%)
(A) Desktop Computer	14.39	20.17	40.17	14.22	17.23	21.16
(B) 60 W Bulb	17.22	18.18	5.57	17.02	15.53	8.75
(C) 24" Monitor	16.63	23.8	43.11	16.44	20.33	23.66
(D) Smartphone	1.92	1.87	2.60	1.89	1.59	15.87
(E) 150 W Bulb	51.02	53.04	3.96	50.43	45.31	10.15

(d) Results

Figure 10: Deltaflow performance on a realistic tree of loads. Figure 10(a) shows the hierarchy of loads and the corresponding sensors. Figure 10(b) shows the ground truth power traces for the aggregate and each load. Figure 10(c) shows the Deltaflow estimates for all five loads and an estimated aggregate for comparison purposes. Figure 10(d) shows the breakdown of overall energy usage by each load and the estimate provided by Deltaflow, as well as the percent of the total energy usage each load represents. While the error for loads (C) and (E), which change power states at similar times, is high (> 40%), Deltaflow handles loads that transition independently successfully. Also, Deltaflow is able to successfully integrate streams from multiple types of pulse counting sensors.

5.5 Load Tree

To mimic a realistic load tree as one might find in a residential or commercial building, we construct a load hierarchy and instrument it with pulse sensors as shown in Figure 10(a). The five loads at the leaves of the tree are (A) desktop computer, (B) 60 W incandescent light bulb, (C) 24" monitor, (D) smartphone charger, and (E) 150 W incandescent light bulb. The 150 W light bulb is monitored with a Gecko sensor and the other four loads are attached to Coilcube sensors. Three split-core current transformer sensors monitor the aggregate and each main branch of the tree. Each load and the aggregate is also metered with a ground truth meter, so we have full knowledge of the power draws.

Figure 10(b) shows ground truth power for each load as well as the aggregate, and Figure 10(c) shows the resulting estimates of power draw from the Deltaflow system and the sum of the estimates to form an estimate of the aggregate. The primary error in Deltaflow's results is overestimation. Notably, between $t = 20$ min and $t = 28$ min, all loads except for (D) are off, causing the ground truth aggregate to be very low. However, Deltaflow estimates the off loads at a higher power state, causing the estimated aggregate to be significantly higher. This error primarily stems from the design of the pulse meters. When no current is flowing to each load, the energy-harvesting pulse meters are unable to report any pulses. Without data from the pulse streams, Deltaflow does not update its estimate for the load power until it receives a new pulse, which it then uses to backfill the missing data. This causes it to overestimate during periods in which most loads are turned off and there are no transitions. To compensate for this error, heuristics could be added that predict the likelihood the load is off given no incoming pulses and update the estimates accordingly. Alternately, the pulse sensors could be modified to report their state if they detect a sudden loss in harvestable energy income.

To quantify how Deltaflow performs in a use-case likely to be relevant to consumers, we explore how well Deltaflow estimates kilowatt-hours (kWh), the unit of energy by which consumers typically buy their electricity. Further, we evaluate how well Deltaflow is able to disaggregate the total kWh consumed into the percent contribution of each load. This information would be particularly useful for a consumer to determine where to direct effort when seeking energy efficiencies. Figure 10(d) shows this breakdown. The percent error of the estimated kWh values for loads (B), (D), (E) is under 6%. These loads tended to change independently of other loads, allowing Deltaflow to better determine their calibration functions and disaggregate the total to these loads. Loads (A) and (C), however, are likely used together and transitioned power states at similar times. This adversely affects Deltaflow's ability to successfully disambiguate two loads leading to the relatively high (> 40%) errors in the kWh estimation. These errors would likely be reduced with a longer running experiment that exhibits a greater number of transitions that Deltaflow could use for calibration.

Also included in Figure 10(d) is a breakdown of each load based on the fraction of the total kWh each load consumes. While the errors in absolute kWh are still present in the breakdown, general trends about load consumption are still clear in the Deltaflow estimates: load (E) dominates the contributions of the other loads, loads (A), (B), and (C) have approximately the same contribution, and load (D) is an insignificant fraction of the total. These results show that Deltaflow supports critical comparative analyses of this nature that are essential for determining how best to invest constrained resources toward increasing efficiencies.

Figure 10 also demonstrates the effectiveness of integrating pulse meters that are based on different types of sensors and energy-harvesting front ends. Monitoring load (E) with only a Gecko node does not hinder Deltaflow's ability to disaggregate the load. Although we use only three different kind of pulse sensors in these experiments, we envision many other kinds might be used in the future to monitor not only electricity, but water and gas as well.

6. DISCUSSION

Designing a system with few constraints on input data and few assumptions about sensor calibration results in some limitations, but it also offers many opportunities for improvement.

6.1 Limitations

The Deltaflow system has some obvious limitations, which we now discuss, as well as possible ways to address them.

6.1.1 *State Identification*

A major challenge in achieving high accuracy is correctly identifying when a load is in a power state that would be useful for the regression engine, particularly if Deltaflow is processing in real-time. Poor state identification leads to inaccurate curves being fit to model the pulse-counter frequencies. We believe that techniques such as stepwise approximation algorithms and histogramming can be added to Deltaflow to improve state identification and make Deltaflow more robust to highly variable loads.

6.1.2 *Environmental Bias*

Pulse sensors may be triggered by more than simply the targeted load, causing pulses that are not correlated to the power draw of the load being metered. One example of this is the solar cell-based sensors that can be activated by ambient light, as shown in Figure 11. This type of error could be compensated for by using additional sensors that are designed to only capture environmental bias conditions and subtracting their pulses from the affected sensors.

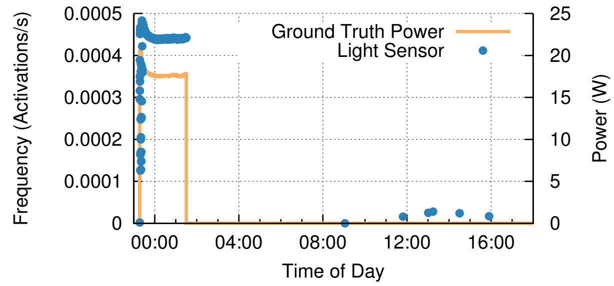


Figure 11: Effect of ambient light on Gecko sensors. A Gecko node is placed in a lamp shade near a window. When the light turns on shortly before midnight, the sensor correctly activates in response. Long after the light is turned off, however, the sensor activates several times between 9:00 AM and 4:00 PM in response to ambient sunlight. These activations are false positive and they must be accounted for to ensure robustness disaggregation.

However, differentiating between the load and environmental sensors may require supplying the system with additional metadata, which may complicate deployments.

6.1.3 *Aliased Loads*

It is possible for an unmonitored (hidden) load to change power states simultaneously with a monitored load such that the change in the aggregate that corresponds to the metered load is masked. Similarly, two monitored loads that change power states simultaneously will cause a single change in the aggregate, making disaggregation between those two loads difficult. In the first scenario, we assume that this case will not occur too frequently, and while it will be a source of error when it happens, over time the error will be mitigated. In the second case, it may be acceptable to group the power draw of items that transition simultaneously, as that likely represents a composite load that can be thought of as a single load, like a TV and surround sound speaker system.

6.1.4 *Privacy*

Conveying information via the characteristics of the transmission channel instead of the content obviously raises security and privacy issues, due to potential leakage of the occupant activities as they are performed within a private space.

While content can be encrypted, it is much more difficult to hide from a malicious observer the transmission frequency. In order to obscure the timing characteristics of sensor wakeups so that they are not detectable by a third party but are still available to the intended recipient, we must convey timing information over a regular encryptable data channel, not a side-channel.

To report wakeups over a regular data channel, the sensors need to transmit periodically and provide a recent history of wakeup times within each transmission. However, storing and reporting time-series data is a challenge as the intermittently-powered energy-harvesting sensors that we use are unable to support real-time clocks (RTCs). To provide these highly-constrained sensors with a notion of temporal history, we propose an unconventional method for representing the passage of time.

The main resources that our sensors have for carrying the effects of past states into the present is their timing capacitor and flash memory. If the sensors record the voltage level of their timing capacitor upon wakeup, then Deltaflow need only convert a report of these voltage levels into the corresponding wallclock times to obtain the time-series and subsequent sensor wakeup frequency.

To map voltages to time, we propose a two phase approach consisting of *model acquisition* and *secure sensing*. During model acquisition, when a sensor powers itself it immediately transmits an encrypted packet containing the current timing capacitor voltage level. These transmissions allow Deltaflow to build a model of the load power-timing capacitor relationship. During secure sensing, the sensor records voltage levels upon wakeup. The sensor periodically transmits that history to the server which reconstructs it using the initial model, thus preserving some level of privacy.

6.2 Future Work

We envision four key areas for future work. These areas focus on improving the deployability and automation of the system by providing real-time processing capabilities, supporting redundant sensors, removing the need to specify the parameterized form of the calibration function, and miniaturizing the sensors.

6.2.1 Real-time Processing

In a practical deployment, the disaggregation system will likely need to update in real time. This presents unique challenges over static data processing because a sensor's activation frequency at a given time is unknown until the next pulse is received. If a pulse stream suddenly stops, Deltaflow loses its information source.

One possible solution is to assume that a load will stay constant. This means that the activation rate will also stay constant, and Deltaflow can predict the latest time by which the next pulse should arrive. If the pulse does not arrive by the expected time, there are four possible explanations: the load has reduced its power draw, the load has turned off completely, a packet has been lost, or the sensor has been lost (i.e. is broken or has been removed).

If Deltaflow estimates the expected arrival time of the next sensor pulse for each sensor, it can identify missing pulses and accommodate them by incrementally reducing the power estimate for the load until the server receives the next pulse or the estimate reaches zero. If the load reduces its power draw, then this process will draw a line that slopes down to the new estimate. If the load turns off, then this method will converge to a power estimate of zero.

If a packet is lost, then using this method will cause Deltaflow's estimate to dip during real-time operations. However, packets from the pulse sensors contain sequence numbers that allow the system to retroactively correct the estimate when the next pulse is received. How *post facto* corrections are used is a different matter.

If the sensor is lost, then Deltaflow will generate an incorrect estimate for that load. The aggregate will not drop, however, so the error of the estimated aggregate will increase. When the error is greater than a certain threshold, Deltaflow is alerted to the fact that its system model is no longer accurate. Problematic sensors can be identified using techniques like the largest normalized residual test, which is used for detecting faulty sensors during the bad data processing phase of power system state estimation. Once the bad sensor is identified, Deltaflow can alert the user and re-run the regression problem without that sensor to determine an updated system model. This method of monitoring the error and re-running when the error exceeds a threshold can be used to compensate for other error sources, such as changing power factors, new sensors, and changes in load characteristics.

6.2.2 Multiple Sensors

One potential method for reducing error in the system is to add multiple sensors per load, e.g. using both a Gecko and Coilcube sensor on the same lamp. This would both provide additional hints for the Deltaflow system and remove the current constraint that allows at most one sensor per load.

Figure 12: A mm-scale sensor node that integrates a solar cell, processor, radio, and battery that provides much of the pulse sensor functionality. This illustrates that the "peel-and-stick" sensor tags we envision will soon be viable.

Extending Deltaflow to support multiple sensors would involve adding a method for determining which sensors represent the same load. This could be done by observing that a set of sensors are synchronized in their pulse frequency deviations, perhaps by observing the sensor covariance matrix. However, due to differences in sensor response times, determining their synchronized behavior may be difficult, requiring that sensor traces be partitioned into stable and volatile segments. Manually augmenting the system with metadata would be an effective but labor-intensive and perhaps error-prone fallback should efforts at automation prove too complex.

6.2.3 Adaptive Models

Currently, Deltaflow requires that the monotonically increasing calibration function for the sensors be provided in a general form *a priori*. Our implementation uses non-negative least squares regression to enforce the monotonicity constraint, but this may rule out monotonically increasing functions with negative coefficients. A more general approach may be to use semi-definite programming to establish the monotonicity constraint. It may also be possible to eliminate all foreknowledge of the calibration function. Calculus of variations is an analytical tool that can discover arbitrary functions that minimize some other function. If these techniques could be used to discover the functions at runtime that minimize the error of the estimated aggregate, then we could eliminate assumptions about the form of the calibration function.

6.2.4 "Peel-and-stick" Sensors

While the current generation of sensors are functional, we envision a future generation of pulse sensors that are truly peel-and-stick, like RFID tags are today. Figure 12 shows an example of such a sensor. At this scale, the sensor could be integrated into laminated tags and easily affixed to loads.

7. CONCLUSIONS

A recent U.S. National Science and Technology Council report states that the ability to submeter electricity and water in modern buildings is critical to meeting Federal sustainability targets, but the same report also notes the difficulty of submetering at scale. This paper explores a new approach to submetering that does not suffer from many of the drawbacks of current systems. We leverage recent advances in inexpensive—but low-quality—energy-harvesting sensors that can be deployed broadly and propose new algorithms to processing their intermittent, inaccurate, and noisy data streams. Our results show that it is possible to deploy small, unobtrusive, inaccurate, and uncalibrated sensors, and yet still be able to estimate individual contributions to an aggregate load. Miniaturizing the sensors and supporting other modalities will soon pave the way to pervasive, fine-grained, "peel-and-stick" submetering solutions.

8. ACKNOWLEDGMENTS

Special thanks to Jake Abernethy for introducing the initial regression formulation, Quentin Stout for sanity checks on partial coverage and state detection, Jeremy Hoskins for suggesting calculus of variations as a way to discover calibration functions, and the students of Lab 11 for all the reviews, food, and support. This work was also supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP), a Semiconductor Research Corporation program sponsored by MARCO and DARPA. This material is based upon work partially supported by the US Agency for International Development, the National Science Foundation under grants CNS-0964120, CNS-1111541, and CNS-1350967, and generous gifts from Intel, Qualcomm, and Texas Instruments.

9. REFERENCES

[1] T. Campbell, E. Larson, G. Cohn, J. Froehlich, R. Alcaide, and S. N. Patel. WATTR: A method for self-powered wireless sensing of water activity in the home. Ubicomp '10, 2010.

[2] G. Cohn, S. Gupta, J. Froehlich, E. Larson, and S. N. Patel. GasSense: Appliance-level, single-point sensing of gas activity in the home. Pervasive '10, 2010.

[3] Cooper Power Systems. Energy Harvesting (EH) Power Supply and Repeater. http://www.cooperindustries.com/content/public/en/power_systems/products/automation_and_control/amr_ami/energy_harvesting.html, 2014.

[4] S. DeBruin, B. Campbell, and P. Dutta. Monjolo: An energy-harvesting energy meter architecture. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, 2013.

[5] Energy, Inc. TED The Energy Detective. http://www.theenergydetective.com/, 2013.

[6] J. Froehlich, E. C. Larson, T. Campbell, C. Haggerty, J. Fogarty, and S. N. Patel. Hydrosense: infrastructure-mediated single-point sensing of whole-home water activity. In *UbiComp'10*, pages 235–244, 2009.

[7] S. Gupta, M. S. Reynolds, and S. N. Patel. Electrisense: Single-point sensing using emi for electrical event detection and classification in the home. In *UbiComp'10*, 2010.

[8] G. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.

[9] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity ac metering network. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, IPSN '09, pages 253–264, 2009.

[10] D. Jung and A. Savvides. Estimating building consumption breakdowns using on/off state sensing and incremental sub-meter deployment. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 225–238, 2010.

[11] Y. Kim, T. Schmid, Z. M. Charbiwala, J. Friedman, and M. B. Srivastava. NAWMS: nonintrusive autonomous water monitoring system. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 309–322, 2008.

[12] Y. Kim, T. Schmid, Z. M. Charbiwala, and M. B. Srivastava. Viridiscope: design and implementation of a fine grained power monitoring system for homes. Ubicomp '09, pages 245–254, New York, NY, USA, 2009.

[13] Y. Lee, S. Bang, I. Lee, Y. Kim, G. Kim, M. H. Ghaed, P. Pannuto, P. Dutta, D. Sylvester, and D. Blaauw. A modular 1 mm^3 die-stacked sensing platform with low power I^{2}C inter-die communication and multi-modal energy harvesting. In *IEEE Journal of Solid-State Circuits*, volume 48, 2013.

[14] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 119–127, 2007.

[15] P. Martin, Z. Charbiwala, and M. Srivastava. DoubleDip: Leveraging thermoelectric harvesting for low power monitoring of sporadic water use. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 225–238, 2012.

[16] National Science Board. Building a sustainable energy future: U.S. actions for an effective energy economy transformation, Aug. 2009.

[17] NSTC–Committee on Technology. Submetering of building energy and water usage: Analysis and recommendations of the subcommitte on buildings technology research and development, Oct. 2011.

[18] P3 International. Kill-A-Watt wireless. http://www.p3international.com/products/consumer/p4220.

[19] S. Patel, T. Robertson, J. Kientz, M. Reynolds, and G. Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. Ubicomp '07. 2007.

[20] N. B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao. Tiny web services: Design and implementation of interoperable and evolvable sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 253–266, 2008.

[21] A. Rowe, M. Berges, and R. Rajkumar. Contactless sensing of appliance state transitions through variations in electromagnetic fields. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, 2010.

[22] Smarthome. iMeter Solo. http://www.smarthome.com/2423A1/iMeter-Solo-INSTEON-Power-Meter-Plug-In/p.aspx.

[23] V. Srinivasan, J. Stankovic, and K. Whitehouse. Fixturefinder: Discovering the existence of electrical and water fixtures. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pages 115–128, 2013.

[24] Watts up? Watts up? .net. https://www.wattsupmeters.com/secure/products.php?pn=0&wai=0&spec=1.

[25] Wattvision. Wattvision: The Smart Energy Sensor. http://www.wattvision.com/, 2013.

[26] T. Weng, B. Balaji, S. Dutta, R. Gupta, and Y. Agarwal. Managing plug-loads for demand response within buildings. In *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '11, pages 13–18, 2011.

[27] T. Wu and M. Srivastava. Low-cost appliance state sensing for energy disaggregation. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '12, 2012.

[28] L. Yerva, B. Campbell, A. Bansal, T. Schmid, and P. Dutta. Grafting energy-harvesting leaves onto the sensornet tree. In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, IPSN '12, pages 197–208, 2012.