Accurate Real-Time Occupant Energy-Footprinting in Commercial Buildings

Yun Cheng⋆, Kaifei Chen†, Ben Zhang†, Chieh-Jan Mike Liang‡, Xiaofan Jiang‡, Feng Zhao‡

⋆ Harbin Institute of Technology † University of California, Berkeley ‡ Microsoft Research Asia

chengyun@hit.edu.cn, {kaifei, benzh}@berkeley.edu, {liang.mike, fxjiang, zhao}@microsoft.com

Abstract

Buildings consume a significant portion of the total delivered energy. While the community has been working on monitoring the building energy usage, we argue that an accurate accounting of individual occupants' energy expenditure in real-time is still the missing piece. And, this missing piece makes incentivizing energy reduction a challenge. We take a systematic approach and identify the lack of real-time association between human actions and observed energy consumption as one roadblock. To this end, we first present a model that enables real-time accounting of appliances with delay characteristics. Building on top of this model, we introduce a complete system that fairly attributes energy usages of shared resources, and reveals per-person energy footprint information. And, our system has several ways to analyze and visualize individuals' energy footprint. Finally, we present the deployment experience at an office building.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Measurement, Economics

Keywords

Building Monitoring, Energy Allocation Model, Real-time Feedback, Triggers

1 Introduction

Buildings, and commercial buildings in particular, are responsible for a large portion of world's total delivered energy consumptions [4]. As the energy crisis becomes increasingly urgent, we have seen a renewed interest on building energy monitoring from both academia and industries in recent years. Some of these work focuses on real-time monitoring of house or building level energy consumptions, using wired or wireless panel level meters [13, 5]; some uses novel non-intrusive load monitoring (NILM) algorithms to break down aggregate energy consumptions into each appliances [16]; and some uses a network of plug-load meters to directly measure appliance energy consumptions [10]. While these previous work has made a significant progress in understanding the building energy consumptions, we are still missing a critical step - understanding the energy footprint of individual occupants as they live out the day inside a commercial building.

We argue that it is this missing step of accurately accounting for individual occupants' energy footprint that makes incentivizing energy usage reduction a challenge. In other words, the lack of visibility hinders real-time and meaningful feedback to the public. To this end, we present the model, architecture, implementation, and deployment results of a building occupant-level energy-footprinting system. Our goals are to provide a real-time view of individual occupants' energy impacts and respond intelligently to reduce consumption; and allows building managers to better manage resources and set energy-related policies. Finally, with the network infrastructure and services in the cloud, we encourage third party applications to build on top of this infrastructure.

Occupant-level energy-footprinting is challenging primarily because of the following two reasons. First, we need a way to fairly divide the energy consumption of a shared resource, such as HVAC or refrigerator in a public kitchenette. Second, some public/shared appliances do not exhibit real-time relationship between usages and the associated energy consumption (e.g., taking out a can of soda does not immediately trigger the fridge to enter the cooling operation).

To address these challenges, we first propose an estimator-based power model for several large classes of appliances and spaces. This model allows us to better tie appliance usage with energy consumption. Second, we present several ways to correlate energy usages of shared resources to the responsible occupant. Third, we present several ways to visualize and feedback energy-footprints; and lastly, a trigger-based mechanism to automatically reduce energy.

1.1 Contributions

This paper makes the following three contributions.

First, we design an energy usage model for several large classes of appliances and spaces, both exclusive and share (c.f. Section 2). The model decomposes the power usage

Buildsys'12, November 6, 2012, Toronto, ON, Canada.

into base and per-use power, with each parameter derived from historical data. This approach enables the appliances with delay characteristics to be accounted accurately in real-time.

Second, we built a complete system on top of the proposed energy usage model (c.f. Section 3). The system includes various front-end sensors and data analysis engine on the back-end servers. And, the system has several ways to analyze and visualize individual energy consumption. For example, individual occupants can view their total energy-footprints growth over time. They can also view the energy usages with respect to each appliance or resource. For building managers, the system can visualize the base and per-use energy usage for each application or resource.

Third, the paper presents our experience and results from one week-long deployment (c.f. Section 4). One observation is that our trigger-based mechanism allows non-technical users to program intelligent actions and encourage energy saving through automatic device actuations or real-time reminders. Specifically, the intuitive user interface on mobile phone and web portal simplifies the task of connecting physical and virtual sensors/actuators with triggers. Our deployment results show an energy reduction of up to 40.8% in the case of fridge with the reminder trigger.

2 Appliance Modeling and Attribution

There are several classes of appliances in a commercial building; some are dedicated to particular occupants, such as desktop computers, and some are shared, such as HVAC or refrigerators in kitchenettes. For dedicated resources, attribution is usually straight-forward (even though our system also supports time-sharing between multiple occupants for dedicated resources), but for shared resources, how to accurately and fairly allocate their energy consumptions is yet unsolved. This is an interesting problem primarily because for some classes of appliances, we cannot simply take the instantaneous power consumption and compute its energy over the duration of its user's access time. For example, taking a can of soda from a shared fridge may last a few seconds, but the energy consumption correlated to this action does not occur until some time later when the compressor kicks in (since fridges operate in a hysteresis region).

In this section, we first introduce a simple yet effective way to model both dedicated and shared appliances. The model separates "base" power/energy and "per-use" power/energy. We then show how to compute and update the parameters of this model, followed by how to compute individual occupant energy consumptions based on these parameters.

2.1 Background

Energy-attribution may seem like a trivial problem at first, given that we already have technology to measure and report in real-time power consumptions. One could, for example, directly meter the power of an appliance and attribute it to its owner; or in shared appliance cases, use some way to detect human access (more on this in later sections) and divide energy consumption by usage time. This approach works fine for *state-less* and *memory-less* devices such as lights or TV. Here we define the two terms as follows:

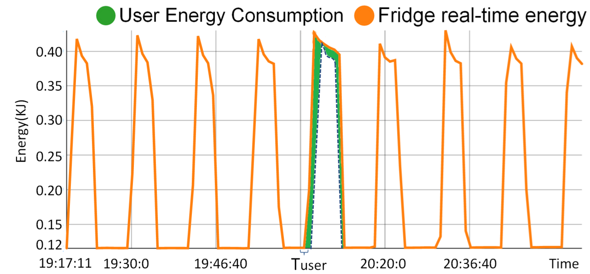


Figure 1. Fridge exhibits the memory property

- **State-less** – Total energy consumed is simply the sum of energy consumed from users' active usage
- **Memory-less** – Energy consumed directly and fully reflects the current usage in time

However, for devices that satisfy at least one of the conditions, it is unclear how to attribute energy. For example, fridge is stateful, as it requires some *base* power to maintain constant temperature (even without any active user). Fridge also has memory, as the addition of an active user does not immediate increase its power consumption until some time later.

Figure 1 shows the energy trace of a typical fridge in a public kitchenette for about 90 minutes. For this device, the compressor needs to run periodically to cool the interior, regardless of actual uses. The result is a relatively large *base* energy expenditure. At time T_{user}, a user opened the fridge for less than a minute. This leads to a slightly longer compressor cycle that follows after. However, the exactly contribution of that user's last access cannot be measured directly, but can be estimated if we know how much energy it would otherwise consume if the user did not use the fridge, based on historical data. We label the per-use power in green, to roughly indicate the difference. In fact, the majority of the energy expenditure did not occur until the user closed the fridge door.

This example highlights the challenge that our work takes the first step in solving. Although the actual energy usage may not occur until some time in the future, the user feedback should still happen in real time. Therefore, we need a model that can estimate the necessary data. Specifically, Nevertheless, we need some way to attribute the "base" power to the occupants (even if they do not directly benefit from it), and the additional "per-use" power required to heat or cool that last user.

To address this problem, we propose an appliance model that considers the following two parameters:

- **Base** power – The operational power that an appliance must consume to be able to function, regardless of usage
- **Per-use** power – The additional power consumed for a single use, which reflects the power consumption due to a particular user

2.2 Appliance Base Models

Our model makes the following two assumptions.

1. The energy consumption of a device is equal to base

energy consumption plus the sum of all "per-use" power by its users

2. There is a positive relationship between device energy consumption and the device usage time

Suppose the current time is T_c and a time window T_w before T_c, we have the following equation.

$$E = P * T + B \tag{1}$$

E is the device's real energy usage during the time window (but may be measured with a delay ΔT from the current T_w, as described in more detail in Section 2.3.1), P is the per-use power, T is the total device usage time during the time window, and B is the base energy.

For every time window T_w, E and T are known while P and B are unknown. Since T_w is a running time window, we can arrive at a linear system of equations, with the number of rows defined by the size of our window and the update rate. We choose to use more equations as we approach T_c since recent history is usually more indicative of the near future. With simple regression analysis, we can compute the parameters P and B.

After updating the parameters, we can use equation 2 to allocate the energy of device k to user i. P and B are the parameters we computed above. T'_w is the energy allocation time window. $T_{i,k}$ is the device usage time of user i during the time window. $E_{i,k}$ is the energy allocated to the user i during this allocation window. N is the number of users who shared the device. We note that N is defined over a time window as well, such as the current hour, day, or even month, per policy.

$$E_{i,k} = P * T_{i,k} + \frac{B}{N}\frac{T'_w}{T_w} \tag{2}$$

In the energy allocation process, we update parameters as time goes on and allocate energy consumption of each device to each user using the latest parameters. Therefore, for each user, we can compute the real-time energy consumption (or energy footprint) by summing all individual usages over all devices M, as below.

$$E_i = \sum_{k=1}^{M} E_{i,k} \tag{3}$$

2.3 Discussions

2.3.1 Parameter Estimations

The previous section mentions that, at each update time, we can compute P and B by solving a linear system of equations (of the form Equation 1) with regression analysis. The objective is to maximize the regression line approximation to past data points, or the R^2 value. However, for appliances with the memory property, the choices of timing parameters can influence the modeling results.

In our experiments, we find that if one uses a device (with the memory property) during a time window, the associated energy consumption is usually delayed till the next time window. In other words, E of Equation 1 represents the device's real energy usage due to the user's usages in T_w, but the metering hardware may not observe the change until time $T_w + \Delta T$. In other words, we may need to advance the energy measurement window by ΔT to capture the energy consumption changes. We can determine the optimal value of ΔT by trying to maximizing R^2. Or, we increase ΔT until R^2 is maximized. In fact, we can also classify the appliances as having the memory property if ΔT is positive.

T_u is the parameter update interval, and its optimal value depends on the frequency of the appliance's active usages. Since appliance's usage pattern varies over time, we adopt a dynamic approach to estimate T_u as follows. We double T_u if the R^2 values from two consecutive parameter updates do not differ by more than a threshold. Otherwise, we decrease T_u by one unit. In our experiments, T_u varied between 5 and 10 minutes. parameters, we choose more equations closer to the current time horizon T_c.

T_w can be viewed as a constraint on the training data set for Equation 1, and it depends on the appliance's behavior. Ideally, T_w should be large enough to cover a wide range of data. However, it should not be too large to avoid accepting data points not collected in the same state as the current one. We use the R^2 value to evaluate how well the regression line approximates the points within T_w. Section 4 uses this technique to pick a T_w of 60 minutes.

In Equation 2, the device usage time $T_{i,k}$ is measured in different ways. For example, we use a light sensor to detect door openings for the fridge, and correlate with an indoor localization device to determine the user ID; for the coffee-machine, we use the real-time energy record to determine $T_{i,k}$. Please refer to Section 3 for more details.

2.3.2 Energy Consumption Estimations

In Section 2.2, Equation 2 takes in the most recent P and B to estimate the current energy consumption. However, our experience suggests that this simple approach does not always work well, especially for stateful appliances. Considering the running example of fridges, fridges have a distinct state of cooling and duty-cycling states (c.f. Section 4). After transitioning to a new state, if the system does not update model parameters in time, all energy consumption estimations after the transition can have a large error margin. To address this problem, the system includes both *single-pass* and *double-pass* strategies.

- Single-pass strategy: Use the most recent P and B to calculate the current $E_{i,k}$. We note that T'_w is usually very small to increase the frequency of model parameter updates, especially for stateful appliances. However, the tradeoff is the increase in computational overhead from frequent model parameter updates. The double-pass strategy relaxes this requirement.
- Double-pass strategy: Double-pass follows the same procedure as single-pass, except for the additional pass on correcting potential energy consumption estimation errors. Specifically, as the system updates P and B at the end of each time window, it backtracks all energy consumption estimation in this window and recalculates. As a result, T'_w is usually larger than T_u. Although there is a delay before errors are corrected, this strategy eventually gives a more accurate footprint.

3 System Architectures and Implementation

We now build on the modeling results from the previous section, and present a complete real-world system towards real-time energy apportionment.

3.1 Architecture

Figure 2 depicts the system architecture, and it highlights four components: *resource accounting*, *localization inference*, *data analysis*, and *event triggering*.

Both the resource accounting module and localization inference module sit at the lower layer of the architecture. The former generates device-centric data of the raw energy consumption and the operational status (e.g., the fridge door is opened or closed); the latter generates human-centric data such as the human's relative location to appliances. These data subsequently flows to the back-end servers where the data analysis module sits.

The data analysis module tries to answer the following two questions. First, who the potential users of a particular appliance are; Second, how much energy each user is currently consuming. The system currently infers the potential users from the relative distance between the appliance and human subjects. The second question is non-trivial as appliances can have variable delay components in the energy consumption formula. Our system approaches this problem with modeling (c.f. Section 2), and it updates the parameters of each device with each round of incoming data streams. We note that one improvement of our solution over previous work is the combination of on-line estimation and off-line deep analysis of personalized energy consumption apportionment.

Finally, the event triggering module consumes the results from the data analysis module. To increase the public awareness on energy consumption, this module first triggers user-specific web-based/smartphone-based UI to be updated with new results. In addition, this module takes the steps towards energy policy enforcement with user-defined triggers. An example is turning off the television as the user's energy consumption approaches the predefined threshold.

3.2 Implementation

3.2.1 Resource Accounting Module

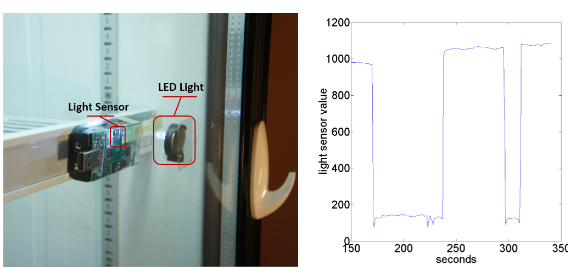


Figure 3. ***Left*** **- A virtual sensor infers the fridge door position from light-intensity data.** ***Right*** **- Spikes in sensor readings indicate door open/close events**

Depending on the appliance of focus, this module can encompass different sets of *physical* and *virtual* sensors. To illustrate, we use fridge as the running example. Our system currently use the physical sensors of energy sensing and temperature monitoring; these two sensors provide visibility into the operating dynamics of the fridge. Knowing precisely when the fridge door is opened introduces some complexities, as there is no single physical sensor that can directly report this information. Our system builds a virtual sensor on top of the light-intensity sensor (c.f. Figure 3). When the fridge door is closed, the LED source is sufficiently close to the light-intensity sensor to register high values. On the other hand, the light-intensity sensor registers low values when the door is opened.

The network backbone provides bidirectional traffic flows between sensors and back-end servers. It carries upstream sensor data that are generated either reactively (e.g., location updates) or periodically (e.g., energy metering). In addition, the network backbone transports downstream commands from the back-end servers for querying and actuating sensors.

The network backbone consists of three parts: a low-power 802.15.4 networking fabric for the sensors, a fast Ethernet fabric for the back-end servers, and gateways that link both networking fabrics. The current system uses the widely-available Ethernet for the back-end server networking fabric. The entire system adopts IPv6 to leverage the vast address space for assigning all devices with unique and globally addressable addresses. Moreover, recent standardization work from the 6LowPan working group makes it possible to deploy IPv6 networks on resource-constrained devices.

3.2.2 Localization Module

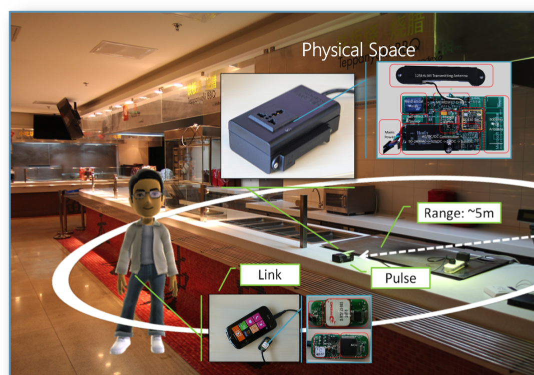


Figure 4. ***Pulse*** **creates a virtual zone with magnetic beacons, which can be detected by** ***Link***

Localization solutions can be categorized by the sensing granularity and implementation differences. Since the most appliances are operational only from arm-reach distance, proximity-sensing solutions are suitable for our purpose.

Common proximity-sensing solutions include NFC-like techniques such as taking photo of the QR code on appliances. However, these solutions require users' involvement. We opt the magnetic-inductance beaconing mechanism. Jiang et al. demonstrated the suitability of this technology in creating consistent and adjustable virtual zones [11].

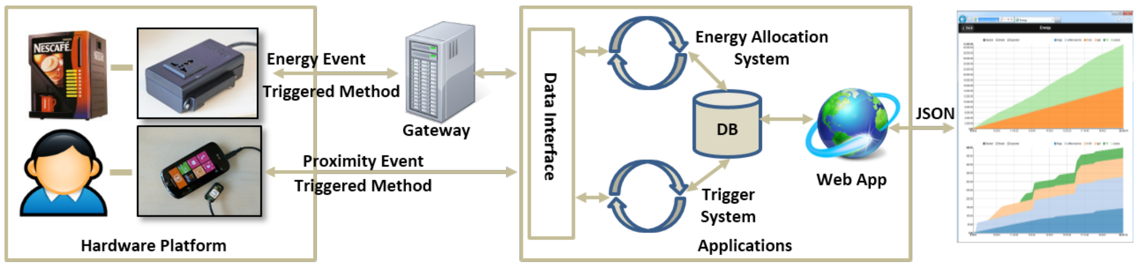


Figure 2. System architecture

Appliances have unique beaconing devices that transmit magnetic signals tuned to 125 kHz (c.f. Figure 4). And, human subjects carry the magnetic sensor plugged into the smartphone's headphone jack for power and data channel.

3.2.3 Data Analysis and Event Triggering Module

Our back-end servers run open-standard protocols: JSON for data representation and HTTP for transport. Similar to XML, JSON messages are text-based, human-readable, self-describing, and language-independent. However, with a smaller grammar, JSON incurs a lower computational and space overhead. These characteristics are suitable for resource-constrained sensors.

We use a set of open-source packages: Tornado Web Server [3], mongoDB database [2], ruby-on-rails. With a non-blocking design, Tornado can scale to thousands of simultaneous standing connections from sensors. mongoDB is a scalable NoSQL database with native support for JSON-style documents. We implement the front-end web portal with the help of ruby-on-rails framework.

For the energy consumption delay component on certain appliances, the data analysis module implements both the single and double-pass algorithm. The former attributes the current energy consumption to active users, and the latter uses the energy consumption aggregated in an appliance-specific window. These two strategies have the trade-off between real-timeness and accuracy of personalized energy accounting. Our deployment experience motivates a hybrid design to maintain the sweet spot (c.f. Section 4).

4 Deployment and Evaluation

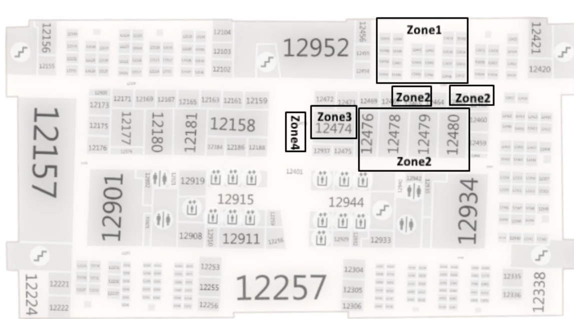


Figure 5. Floor plan of the deployment

To evaluate our system in the real-world, we deployed the complete system covering key parts of a floor inside an office building (c.f. Figure 5). This section presents the results and experience from a week-long segment of the deployment.

4.1 Deployment Setup

Figure 5 shows the four areas of focus: cubicles (zone 1), meeting rooms (zone 2), kitchen (zone 3), and rest area (zone 4). The discussion below describes the device and appliance instrumentation, according to these zones.

Cubicles We accounted the energy consumption of five sets of desktop workstations and peripherals; each set is instrumented with an energy accounting device that functions as pass-through plug similar to ACme [10]. The back-end server received one energy reading per minute. In addition, the work area of each computing set has a proximity-sensing beacon for detecting user presence.

Meeting rooms Since lighting equipment is a major energy consumer in this case, we placed light-intensity sensors to monitor the lighting operational status in six meeting rooms. In addition, a proximity-sensing beacon sits at the center of each room for detecting room occupancy.

Kitchen We instrumented the coffee machine and the fridge with the same energy accounting devices described above. Both appliances also have a proximity-sensing beacon to infer active users. As mentioned in Section 3, the fridge also has a virtual sensor built on top of a LED source and a light-intensity sensor to infer the door position.

Rest area The rest area has a wall-mounted television. We deploy a virtual sensor built on the light-intensity sensor to infer the television operational status. A proximity-sensing beacon is placed directly below the television.

4.2 Data Analysis and Visualization

Section 3 describes the two strategies of the data analysis module: instantaneous and window-based analysis. The former trades accuracy for real-timeness, and vice versa for the latter. Figure 7 depicts the accuracy difference with energy traces from the fridge. Our deployment experience suggests that both variations alone can not fully satisfy users, and motivates a hybrid design. Specifically, we estimate with the instantaneous approach on-the-fly, and then correct the errors over time with the window-based approach.

Figure 6 shows the web portal. The personalized dashboard provides links to device management functionalities (e.g., assigning device ownership and creating triggers) and real-time data visualization tools. We now present the set of visualization tools with the most user visits throughout the deployment.

Device-View The device-view provides snapshots of the per-user energy consumptions for the targeted device. For shared appliances, this view summarizes the energy consumption distribution among users. Figure 8 shows the device-view

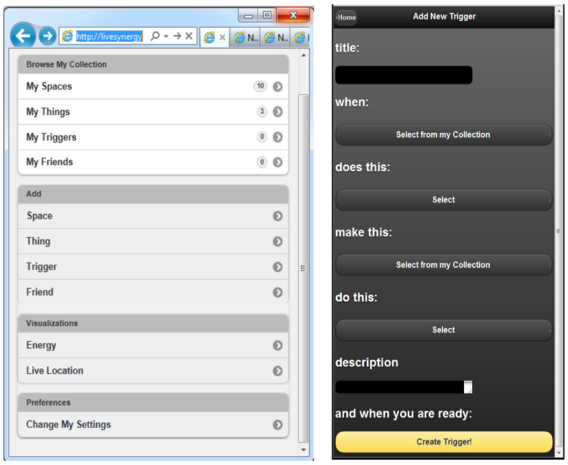


Figure 6. Web portal. *Left* - Personalized dashboard. *Right* - Trigger creation page

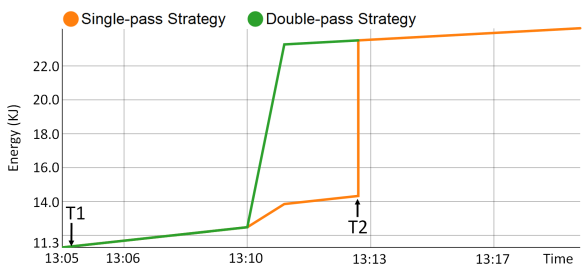
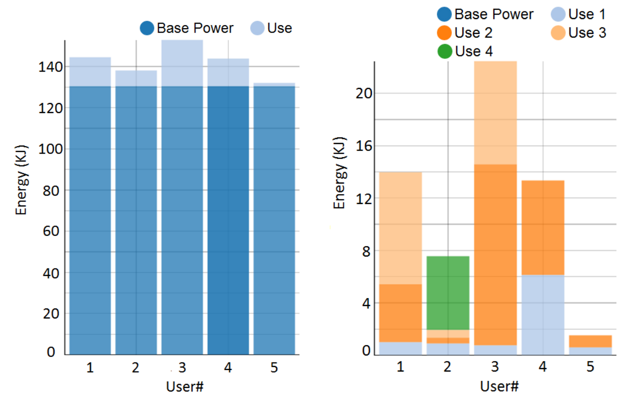


Figure 7. Instantaneous and window-based analysis with the energy trace from the fridge. The two curves show the accuracy difference of the two strategies



(a) Overview (b) Detailed per-use view

Figure 8. Device-view for the fridge

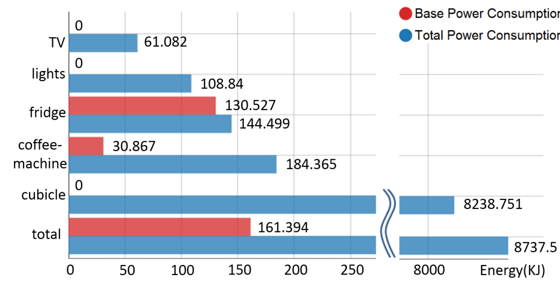


Figure 9. User-view of total energy consumption for each device the user has interacted with

visualization of the fridge. The web portal first shows Figure 8(a), which presents the base power and aggregated per-use power for each user. For the fridge owner, this subfigure reveals how users collectively use the resource. The web portal can also show Figure 8(b) that drills down to the per-use power. Our deployment experience suggests that users prefer this subfigure as it reveals their resource usage patterns.

User-View The user-view visualizes the per-device energy consumption for the targeted user. Figure 9 shows the users' total and base energy consumption in real time, according to the devices they have interacted with. The difference between them is the user's active energy consumption. Figure 10 shows the users' energy consumption over time. And, many users found Figure 10 helpful in determining unattended devices that consumes more energy than they should. Figure 10(a) gives the overview of all appliances monitored, and it shows HVAC and cubicle as the two dominating energy consumers in this case. Then, Figure 10(b) delves into the breakdown of the other appliances.

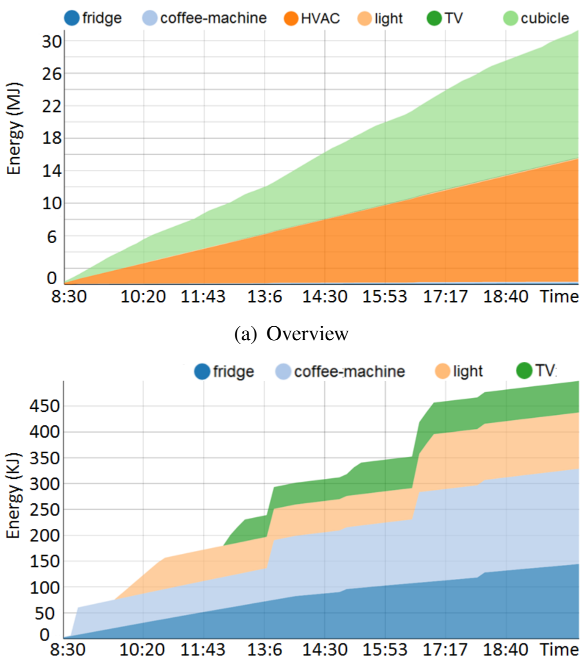
4.3 Sensitivity Analysis

The real-time estimation and visualization feedback depend on the model introduced in Section 2. This section evaluates the effectiveness of the model with the fridge's data sets. Compared to other appliances, the fridge exhibits relatively complicated usage patterns (i.e., users can actuate the fridge by adding and removing objects at different times for variable length of time) and behavior (i.e., there are delays between users' actuations and fridge's reactions).

Figure 11 plots the fridge energy consumption as observed by the energy meter. An interesting observation is the distinct phases characterized by the the frequencies of duty cycles. In the morning, the fridge is in the cooling process and consumes energy to cool the beverages that staffs put in. Then, it enters the duty-cycling state to maintain the interior temperature. This data set allows us to derive the model parameters, P and B, as explained in Section 2.

We first look at the impact of device usage time on P and B. Figure 12 illustrates with fitting curves for both states from the data sets collected. Each data point represents the observed energy consumption for different length of device usage period by users. We note the slope of the curve describes P (per-use power) and the y-axis intersection represents B (base energy). Figure 12 shows that the base energy is significant higher in the cooling state, and the per-use energy is higher in the duty-cycling state. This observation implies that the energy consumption attributed to individuals' current actuations changes according to the fridge current state. Specifically, when the fridge is already cooling the interior (due to room-temperature beverages put in by the staffs), then the additional loads introduced by users are relatively small.

Figure 12 supports our model design decision of separat-



(a) Overview

(b) Zoomed-in view

Figure 10. User-view of energy consumption for each device over time

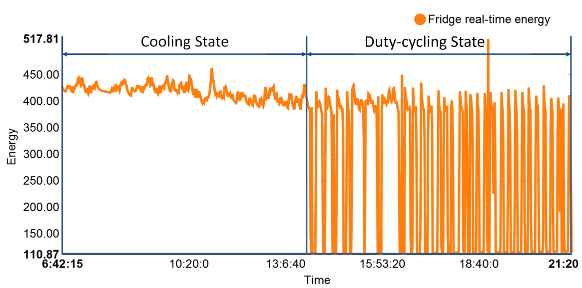


Figure 11. Fridge energy consumption and running state

ing P and B. We now look at how well the model captures the observed fridge behavior. Figure 13 shows both the per-use power and base power with respect to the time. The base power curve captures the morning cooling state from room-temperature beverages being put in. The fridge consumed energy to lower the interior temperature. The per-use power dominated in the afternoon after the beverages were cooled. And, the fridge duty-cycled to counter the temperature increase due to individual uses.

Finally, we look at the choice of T_w on the accuracy of the appliance model. Ideally, T_w should be large enough to cover a wide range of data, but not too large to avoid accepting data points collected in a different state. Our choice of T_w is guided by the R^2 value of the data points within that window, as a high R^2 value maximizes the goodness of fit of the model. Figure 14 shows the results from the fridge's dataset, and we picked $T_w = 60$ for the high R^2 value.

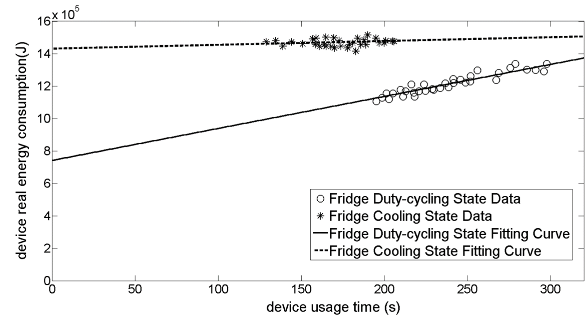


Figure 12. Curve Fitting for different states of the fridge. The slope of the curve describes per-use power and the y-axis intersection represents base energy

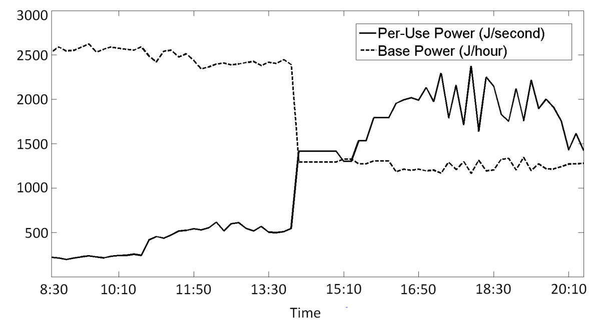


Figure 13. The base power dominates during the day, as the fridge works to cool the interior filled with warm beverages. After that, the per-use power due to user usages dominates the fridge's energy consumption

4.4 Triggers

Triggers take energy accounting to the next level by allowing user-definable policies and event-based actions. For example, users appreciate the use of triggers for reminding them to turn off appliances when they go to bed. We first present the most popular triggers created by the users.

1. If the fridge door has been opened for a long time, then the user receives a SMS alert.
2. If users leave the cubicle without turning off the monitor, then they receive a SMS alert.
3. If users leave the cubicle, then the certain devices are turned off automatically.
4. If users have been consuming more energy than the predefined threshold, then they receive a SMS alert.
5. The system generates energy usage report according to predefined time settings.

	Avg pwr consumption (KJ)		
	Without triggers	**With triggers**	**Energy savings (%)**
Fridge	685.2	405.5	40.8
Computers	1019.7	372.9	63.4

Table 1. The impact of recurring and real-time reminders crated by triggers on reducing energy waste

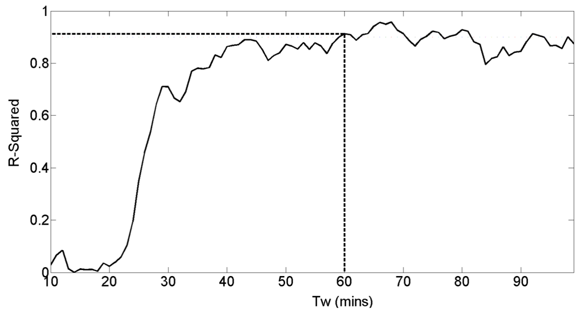


Figure 14. The impact of T_w model parameter on the goodness of fit. We picked $T_w = 60$ in our experiments for the high R^2 value

Table 1 shows the improvement in energy saving that triggers can provide in our deployment. As Jiang et al. [12] mentioned, a single notice to users may wear off over time. On the other hand, triggers provide recurring and real-time reminders to users. In the case of fridge, the user receives an SMS alert when the fridge door has been opened longer than a threshold. The average per-use power consumption drops from 685.2 KJ to 405.5 KJ, or 40.8% energy reduction. A similar trigger was created for computers, and we observed an energy reduction of 63.4%.

5 Related Work

To enable accurate and real-time energy footprinting inside shared spaces, we need ways to monitoring energy usages of various resources in real-time, and ways to correlate that usage to occupants.

The research community has recently made significant progress in building energy monitoring. For example, in [14] and [10], the authors use wired and wireless networked plug-load meters to directly measure appliance energy consumptions. This approach results in accurate energy measurement, but may be costly if deployed for every appliance in the building. Non-intrusive load monitoring (NILM) is a set of techniques for disaggregating energy consumption from an aggregated source [7, 16]. This approach is cost-effective, but often requires calibration and may not be as accurate as direct measurements. In contrast, our work uses a combination of plug-load monitoring and secondary indicators to infer power states.

Associating energy usage to occupants accurately and in real-time is challenging. Previous works such as [6] have focused on using various radio based technologies for localization, which can be used to correlate energy to users. However, RFID and NFC require active user input, which is not convenient. Bluetooth [8] and Wi-Fi based systems are attractive due to their ubiquitous nature, but they are not sufficiently accurate. Jiang et al. demonstrated the suitability of using magnetic fields for creating consistent and adjustable virtual zones in physical space [11], which can be used to mark the active area of an appliance or space. We adopt this approach in our work. In [9], the authors introduced the case for apportionment inside buildings and investigating the differences between a number of possible policies. We build on top of their work, and propose a more complete model that accounts for large classes of devices, including those that have states and memory. In addition, instead of asking users to log usages manually, our system automatically determines usages and attribute energy expenditure to the appropriate users. Furthermore, we improve the allocation accuracy by considering the device usage time.

To reduce the energy consumption and save energy, Mankoff et al. explore how social networks can motivate users to reduce their ecological footprints [15]. HomeMaestro [1] is a prototype of a home automation system, which uses triggers to address the numerous issues plaguing current home automation systems. Our work shares similar ideas, but focuses on energy saving actions.

6 Conclusion

This paper tackles a critical step in motivating energy consumption reduction: accurate accounting of personal energy footprint. Our system incorporates front-end sensors with back-end data analysis and modeling to accommodate a wide range of appliances with different characteristics. Our deployment demonstrate an energy reduction up to 63.4% with triggers. Looking forward, we are planning for a building-wide deployment.

7 References

[1] Homemaestro. http://shaunsalzberg.com/medialab/homemaestro.

[2] mongodb. http://www.mongodb.org.

[3] Tornado web server. http://www.tornadoweb.org.

[4] Energy outlook 2010. energy information administration. http://www.eia.doe.gov/oiaf/ieo/index.html, 2010.

[5] G. Bellala, M. Marwah, M. Arlitt, G. Lyon, and C. E. Bash. Towards an understanding of campus-scale power consumption. In *BuildSys*, 2011.

[6] D. H and D. Fox. Mapping and localization with rfid technology. In *In International Conference on Robotics and Automation*, 2003.

[7] G. Hart. Residential energy monitoring and computerized surveillance via utility power flows. In *Technology and Society Magazine*, 1989.

[8] S. Hay and R. Harle. Bluetooth tracking without discoverability. In *LoCA 2009: Proceedings of the 4th International Symposium on Location and Context Awareness*, 2009.

[9] S. Hay and A. C. Rice. The case for apportionment. In *BuildSys '09: Proceedings of The First ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, 2009.

[10] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity ac metering network. In *IPSN/SPOTS*, 2009.

[11] X. Jiang, C.-J. M. Liang, K. Chen, B. Zhang, J. Hsu, J. Liu, B. Cao, and F. Zhao. Design and evaluation of a wireless magnetic-based proximity detection platform for indoor applications. In *IPSN/SPOTS*, 2012.

[12] X. Jiang, M. Van Ly, J. Taneja, P. Dutta, and D. Culler. Experiences with a high-fidelity wireless building energy auditing network. In *SenSys*, 2009.

[13] Y. Kim, T. Schmid, M. B. Srivastava, and Y. Wang. Challenges in resource monitoring for residential spaces. In *BuildSys*, 2009.

[14] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *IPSN/SPOTS*, 2007.

[15] J. Mankoff, D. Matthews, S. R. Fussell, and M. Johnson. Leveraging social networks to motivate individuals to reduce their ecological footprints. In *HICSS*, 2007.

[16] A. Rowe, M. Berges, and R. Rajkumar. Contactless sensing of appliance state transitions through variations in electromagnetic fields. In *BuildSys*, 2010.