

**A High-Fidelity Energy Monitoring and Feedback Architecture for Reducing
Electrical Consumption in Buildings**

by

Xiaofan Jiang

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

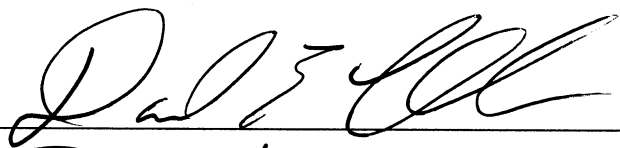

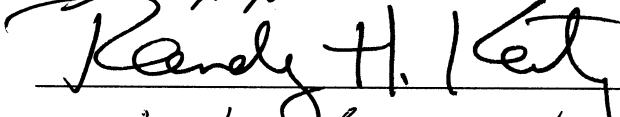

University of California, Berkeley

Committee in charge:

Professor David E. Culler, Chair
Professor David M. Auslander
Professor Randy H. Katz
Professor Seth R. Sanders

Fall 2010

The dissertation of Xiaofan Jiang, titled A High-Fidelity Energy Monitoring and Feedback Architecture for Reducing Electrical Consumption in Buildings, is approved:

Chair		Date	<u>9/7/2010</u>
		Date	<u>9/7/2010</u>
		Date	<u>08 September 2010</u>
		Date	<u>9/7/2010</u>

University of California, Berkeley

**A High-Fidelity Energy Monitoring and Feedback Architecture for Reducing
Electrical Consumption in Buildings**

Copyright 2010
by
Xiaofan Jiang

Abstract

A High-Fidelity Energy Monitoring and Feedback Architecture for Reducing Electrical Consumption in Buildings

by

Xiaofan Jiang

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor David E. Culler, Chair

Existing solutions in commercial building energy monitoring are insufficient in identifying energy waste or for guiding improvement. This is because they only provide usage statistics in aggregate, both spatially and temporally. To significantly and sustainably reduce energy usage in buildings, we need an architecture and a system implementation that provide high-fidelity real-time visibility into each component of the building.

We propose a three-tiered architecture consisting of *sensing*, *data delivery and representation*, and *applications and services*. We show that this layering allows us to cleanly abstract the low-level details of the myriads of disparate monitoring instruments and protocols, provide an uniform data representation interface, and enable innovation in portable building applications. This thesis further explores each layer in detail and present design decisions and findings.

Building on top of this architecture, we propose an application process flow for energy data analysis and visualization, substantiated by a real deployment. This process consists of three parts: first, to understand and instrument the load tree; second, to conduct data analysis, modeling, and disaggregation of energy usage statistics; and third, combined with meta-data, to re-aggregate individual load usages into actionable representations for visualization and feedback to the occupants.

Finally, we evaluate the proposed architecture and process flow with a diverse class of building applications, visualizations, and deployments.

Dedicated to my parents, Guocheng Jiang and Wei Ning,
for their continuous support and encouragement throughout my life.

Contents

List of Figures	v
List of Tables	x
Acknowledgements	xi
1 Introduction	1
1.1 Problem Statement	2
1.2 Architecture Overview	3
1.3 Roadmap	5
2 Background	7
2.1 Energy Flows in Buildings	7
2.1.1 Electric Flows	8
2.1.2 Climate Plants	10
2.2 Current Building Energy Monitoring and Management Solutions	10
2.2.1 Building Management Systems	11
2.2.2 Plug-load Monitoring	13
2.2.3 Power Disaggregation	14
2.3 Sensor Networks	15
2.3.1 Hardware Platforms	15
2.3.2 Networking	16
2.4 Building Energy Modeling Software	17
2.5 Summary	18
3 System Architecture	20
3.1 Building as a System	20
3.2 System Architecture Overview	23
4 Sensing and Measurement	27
4.1 Plug-load Energy Monitoring and Control	28
4.1.1 Problem Overview	28

4.1.2	ACme Hardware Design	29
4.1.3	Embedded Software	37
4.1.4	Evaluation	38
4.1.5	Micro Benchmarking	39
4.1.6	Privacy Implications	42
4.1.7	Enabled Research	42
4.2	Multi-modal Sensing	43
4.3	Summary	46
5	Data Delivery and Representation	47
5.1	Networking	48
5.1.1	Routing	48
5.1.2	Edge Router	49
5.1.3	Transport	49
5.1.4	Network Performance	50
5.2	Uniform Representation of Physical Data	51
5.2.1	Design Considerations	52
5.2.2	ACme sMAP Service	53
5.3	Summary	56
6	Applications and Services	57
6.1	ACme Application	57
6.2	Database and Metadata	58
6.3	Summary	60
7	Analysis and Modeling	61
7.1	Structure of Energy Flows	61
7.2	Modeling and Disaggregation	63
7.2.1	Additivity	63
7.2.2	Multi-resolution	66
7.2.3	Empirical Model	67
7.2.4	Appliance Signature Analysis	68
7.3	Summary	70
8	Feedback	72
8.1	Re-aggregation	72
8.1.1	Functional Re-aggregation	73
8.1.2	Spatial Re-aggregation	74
8.1.3	Individual re-aggregation	76
8.1.4	Electrical Re-aggregation	79
8.2	Visualization	81

8.3	Google PowerMeter	84
8.4	Mobile Applications	85
8.5	Summary	86
9	RadLab as a Green Building Testbed	88
9.1	Network	88
9.2	Usage Profiling	89
9.3	Deployment Model	90
9.4	Results	91
9.5	Improvements	92
10	Ongoing and Future Research	97
10.1	Deployments	97
10.2	Building-to-Grid	99
10.3	Online Energy-Climate Model	100
10.4	None-Intrusive Load Monitoring	101
11	Conclusion	102
	Bibliography	105

List of Figures

1.1	Aggregate energy usage statistics are insufficient for identifying waste or helping to reduce consumption.	2
1.2	System architecture and application process flow.	3
2.1	Energy end-use flow maps for the commercial buildings. Image source: [29].	8
2.2	Example of an electrical distribution diagram inside a typical commercial building.	9
2.3	Example of climate plants inside a commercial building.	10
2.4	SCADA is a supervisory system where a central host controls set-points and remote terminal units (RTUs) perform local decisions. Image source:[82].	11
2.5	Broadwin WebAccess HMI displaying air flows parameters inside the computer science building at Berkeley.	13
2.6	Evolution of early mote platforms developed at UC Berkeley. Source: [68].	16
2.7	Ultrasound transceiver sensor board for measuring distancing using ultrasonic time-of-flight (left); <i>SPOT</i> micro-power meter for high-fidelity instrumentation of low power devices at scale (right).	17
3.1	Existing buildings are often missing the feedback loop between the electrical load tree and the occupants.	21
3.2	Building as a system, is composed of three subsystems – “human behavior”, “climate HVAC”, and “electrical load tree”.	22
3.3	System architecture and the process flow of the ACme application.	24
4.1	ACme consists of five primary components: current-to-voltage conversion, energy metering, AC/DC power supply, microcontroller and radio, and solid state relay.	29
4.2	ACme-A uses shunt resistor as I-V conversion, direct rectification as AC/DC power supply, and ADE7753 as the energy metering IC.	30
4.3	ACme-B uses an in-line Hall Effect sensor as I-V conversion, a step-down transformer followed by a bridge rectifier as the AC/DC power supply, and performs energy calculation in software using the microcontroller.	31

4.4	Current carrying trace has a minimum width of 120mil and uses polygon fills as heat sinks.	37
4.5	ACmeX2 is more compact and efficient but does not include relay.	38
4.6	Power trace of a Polycomm telephone shows that ACme-A is able to resolve low power appliances down to 40mW.	40
4.7	Power trace of a coffee maker shows that ACme-A is able to measure appliances with high power pulses.	40
4.8	ACme reveals interesting characteristics of devices such as a laptop. We can observe the power states, charging curve, and CPU usage.	41
4.9	ACme reveals both the compressor cycles (100W at around 15 minutes intervals) and defrost cycles (450W a few times a day) in a refrigerator.	41
4.10	Figure (A) shows the output of one axis of an accelerometer attached to a large motor driving air conditioning equipment over an hour. Figure (B) shows the output of a simple vibration sensor attached to the same motor over the same hour. Each sensor indicates when the motor is operating and allows for real-time estimation of HVAC energy consumption.	44
4.11	Only 4 light sensors are needed to cover 26 light bulbs since they are controlled together by 4 sets of switches and motion sensors.	45
4.12	Light sensor readings can be easily converted to power using thresholding.	45
5.1	Sensor nodes are connected to the Internet through IP routers. ACme nodes are the sensors and edge routers are Linux-class devices.	49
5.2	This edge router integrates a Meraki Mini and Epic Mote using a custom carrier board and routes IP packets between the sensor network and LAN.	50
5.3	A snapshot of the network topology.	51
5.4	Various meters and sensors that measure physical data, but communicate through different protocols and over different physical mediums. For example, the Dent panel meter uses Modbus protocol over RS485 while ACme meters use TCP and UDP over IPv6 on wireless radios, even-though both are communicating the same type of information - quantity of energy.	52
5.5	IPv6-based sensors such as ACme's implement sMAP directly while legacy devices such as Modbus based branch meters implement sMAP via proxies. Applications interface with any metering resource uniformly via sMAP. In this example, a visualization application is used to graph real-time energy readings from both ACme's and from branch meters.	53
5.6	A fragment of the sMAP RESTful interface implemented by ACme.	54
5.7	A typical ACme sMAP resource URI.	54
5.8	Formatting and parameter resources.	55
5.9	Setting up a per-minute periodic report for the <i>reading</i> resource from ACme 326 to a data repository.	55

6.1	The ACme application architecture.	58
6.2	Reading from an ACme.	59
6.3	Actuating an ACme.	60
7.1	Snapshot of a small portion of the load tree in a computer science building. Voltage is stepped down from the building substation into floor-level electrical panels which in turn distribute power into either AC outlets in the low-voltage case or lights and major equipments in the high-voltage case.	62
7.2	The power that flows through power strip A is the sum of the power consumed by the laptop, the 24" LCD, and the sub-power-strip. We can calculate the power at C1 by simply subtracting M2 and M3 from M1.	64
7.3	The power profile of a 24" LCD can be calculated by subtracting both the laptop metered at M2 and the sub-power-strip metered at M3 from the power strip A, metered at M1. At 1.5hr, the desktop and two 19" LCDs are turned on, using an additional 200W seen by M3 and M1 for about 20 minutes. At 2.5hr, the laptop was removed from the workspace, reducing M2 to 0W. The 24" LCD drops into standby mode after being disconnected from the laptop, drawing around 10W via subtraction.	65
7.4	Sampling rate top to bottom: 4Hz, minute sampling, hourly sampling. (A) reveals the turn-on transient; (B) reveals the compressor and defrost modes of the refrigerator; and (C) reveals the human influence.	66
7.5	Result of calculating an improved empirical model using proportional scaling from a measured aggregate. By using upstream power measurements in addition to generic models of appliance behavior by type ("core power models"), models for specific appliances can be augmented to incorporate device-specific measurements. Here we look at a 24" LCD, a 17" LCD, and two laptops. . .	68
7.6	Laptop energy consumption disaggregation at the sub-component level. . . .	69
7.7	Water dispenser daily energy consumption over the same day of the week for three weeks. Notice the general trend in power consumption in (B) as well as in (C).	70
8.1	A portion of the load tree is decomposed into individual appliances, then re-aggregated into four functional classes - laptop, desktop, LCD, and light. .	74
8.2	Stacked graph of aggregate power consumption grouped by functional class over a day.	75
8.3	Spatial re-aggregation of the load tree. Three rooms (Room-A, Room-B, and Room-C) contain several plug loads (laptop, desktop, LCD screens) and light fixtures. Directly measuring the energy used by a particular space may not be possible if power arrives via multiple load trees, like wall outlets and built-in lighting.	76

8.4	The spatial energy consumption across several offices in an office building. The empirically measured average power draw of several loads is averaged over space using a square smoothing kernel with rounded corners. This figure highlights that the occupants in room 487 have a higher average power demand than those in neighboring offices.	77
8.5	Individual re-aggregation of the load tree. Three building occupants use a mix of electrical loads. Some loads, e.g. laptops or desktops, are dedicated to one user while other loads, e.g. lights or a refrigerator are shared by some subset of the occupants. Directly measuring an occupant's fractional usage may not be possible so approximation techniques are needed.	78
8.6	The distribution of per-occupant average power across a group of nearby occupants. The overall average draw (across all occupants in the monitored area) is 132 W. Each bar includes the fractional contribution from the shared lights and refrigerator (38.5 W), as well as the dedicated contribution from each user's personal loads like laptops, desktop, and monitors.	79
8.7	Per-circuit loads in our office deployment. Loads can be migrated within each grouping simply by moving to a different socket on the same outlet. This data could guide a building manager to improve load balancing among circuits.	80
8.8	A web portal for monitoring of individual energy usages in real-time.	81
8.9	A web based visualization displaying the real-time power usages grouped by appliances types.	82
8.10	A live Sankey diagram showing the energy flows in the building electrical load tree.	83
8.11	Google PowerMeter displaying real-time power usage of a refrigerator measured by an ACme. It provides simple historical statistics and future usage predictions.	85
8.12	An Android application that scans a bar code associated with an appliance monitored by an ACme and graphs its power consumption in real-time.	86
9.1	Location and connectivity graph of AC meters and light sensors. There are a total of 38 AC meters, 28 of which are spatially distinct, and 6 light sensors. The vibration sensor, not shown here, is attached to the HVAC infrastructure in the basement of the building.	89
9.2	The type and number of plug-loads currently monitored using ACme nodes.	90
9.3	Power traces from various appliances under measurement.	91
9.4	Aggregated energy usage by appliance type normalized by number of appliances in an enclosed environment.	93
9.5	Aggregate trace of measured LCDs in the laboratory over a four-week period. The white rectangles represent the work week, shaded vertical strips highlight the weekends, and the solid horizontal line represents the average consumption over the entire duration.	94

9.6	This is a trace of a student’s desktop machine running Windows XP before and after installing power management software. The machine was reconfigured on this day to drop into full sleep mode after 15 minutes of idleness, reducing the idle power from 119.0 Watts to 2.4 Watts.	95
9.7	Estimated saving of the laboratory energy consumption if all of the desktops incorporated energy management software.	96
10.1	ACme deployments and related research.	98
10.2	Using historical load traces and simulated supply, the LoCal simulator predicts future load and performs trades on a virtual market consisting of two types of goods: base power and variable power. The right side of the graph shows how the prediction (green is purchased base power and blue is variable power) matches the actual load (shown in red).	99
10.3	An online energy-climate model that combines real-time sensor feeds with architectural model of the building, enabling dynamic optimizing of energy and comfort.	100

List of Tables

4.1	Properties of ACme-A and ACme-B.	34
4.2	ACme node interface	38
4.3	Summary of ACme-A	39
4.4	Summary of ACmeX2	39
8.1	Survey of densely instrumented region, depicted in Figure 4.11.	73
8.2	Comparison of sMAP and Google PowerMeter	84
9.1	Work-week average aggregate power for measured LCDs, along with week-by-week percentage changes.	93

Acknowledgments

With profound gratitude, I want to thank my advisor and mentor David Culler, who has guided me each step of the way, and taught me what it means to be a researcher. I have grown significantly since I met David, both personally and academically, and I am extremely grateful for his guidance.

I want to thank Professor Randy Katz, Professor Scott Shenker, Professor Seth Sanders, Professor Ion Stoica, Professor Kris Pister, and Professor Shankar Sastry for their advise and support throughout various stages of my academic career.

I want to thank Kamin Whitehouse, Alec Woo, Joseph Polastre, Sarah Bergbreiter, Cory Sharp, Philip Levis, and Robert Szewczyk, who mentored me during my undergraduate and early graduate years, and introduced me to the field of research.

And I want to thank my collaborators, colleagues, and friends, who have provided countless technical, intellectual, and personal help with various aspects of my work. These include, but are not limited to, Stephen Dawson-Haggerty, Prabal Dutta, Jay Taneja, Jorge Ortiz, Jaein Jeong, Arsalan Tavakoli, Ken Lutz, Mike He, Evan Reutzler, Jeff Hsu, Minh Van Ly, Zhangxi Tan, Albert Goto, Andrew Krioukov, Prashanth Mohan, Scott McNally, David Zats, Gilman Tolle, Jonathan Hui, Feng Zhao, Wendy Qiu, and of course, my dissertation committee.

This work was funded in part by the National Science Foundation Graduate Fellowship, the NSF LoCal grant, the NSF Action Webs grant, the NSF Lawrence Berkeley National Laboratory MELS grant, and the NSF CRI grant.

Chapter 1

Introduction

Our insatiable thirst for energy is the societal problem of our age. Electricity usage in residential and commercial buildings consists of 72% of the entire US electrical power expenditure [78], and a third of that is wasted [77]. In addition, because electricity is generated from fossil fuels, it represents a significant portion of the carbon footprint of the occupants. As a comparison, a typical Pacific Gas & Electric (PG&E) customer uses about 540kWh of electricity *per month*, while the Computer Science Division building at UC Berkeley consumes approximately 12MWh of electricity *per day*.

Systems researchers have a great deal to offer to solving this problem. With the rapid advancement in power-efficient and inexpensive micro-controllers, sensors and actuators are becoming increasingly ubiquitous and connected. Even the simplest sensor costing a few dollars can easily be equipped with a network interface and a CPU capable of running a small operating system. The past decade of wireless sensor network research has shown us the potential of intelligent networks of connected devices. We are now in a position to utilize these advancements in technology and the lessons learned from sensor networks research, and apply them in commercial buildings.

Efforts to bring wireless monitoring to buildings have emerged in the past few years in both academia and industry. Most of them consist of feeding metering data back to the user, relying on the users' ability to interpret the data and modify their behavior. However, we have yet to take a systems approach to really understand the underlying phenomenon we want to observe and answer the basic questions of what to measure, where to measure, when to measure, and how to make sense of the data. Only then, can we begin to answer the key questions: where does energy go in a building, and how to create more actionable solutions for reduction? This dissertation introduces an architecture and implementation of a building energy monitoring system, that (1) orchestrates a network of sensors and actuators in real-time, (2) analyzes multi-modal sensor data, and (3) creates models and actionable visualizations based on the fundamental structure of the building.

1.1 Problem Statement

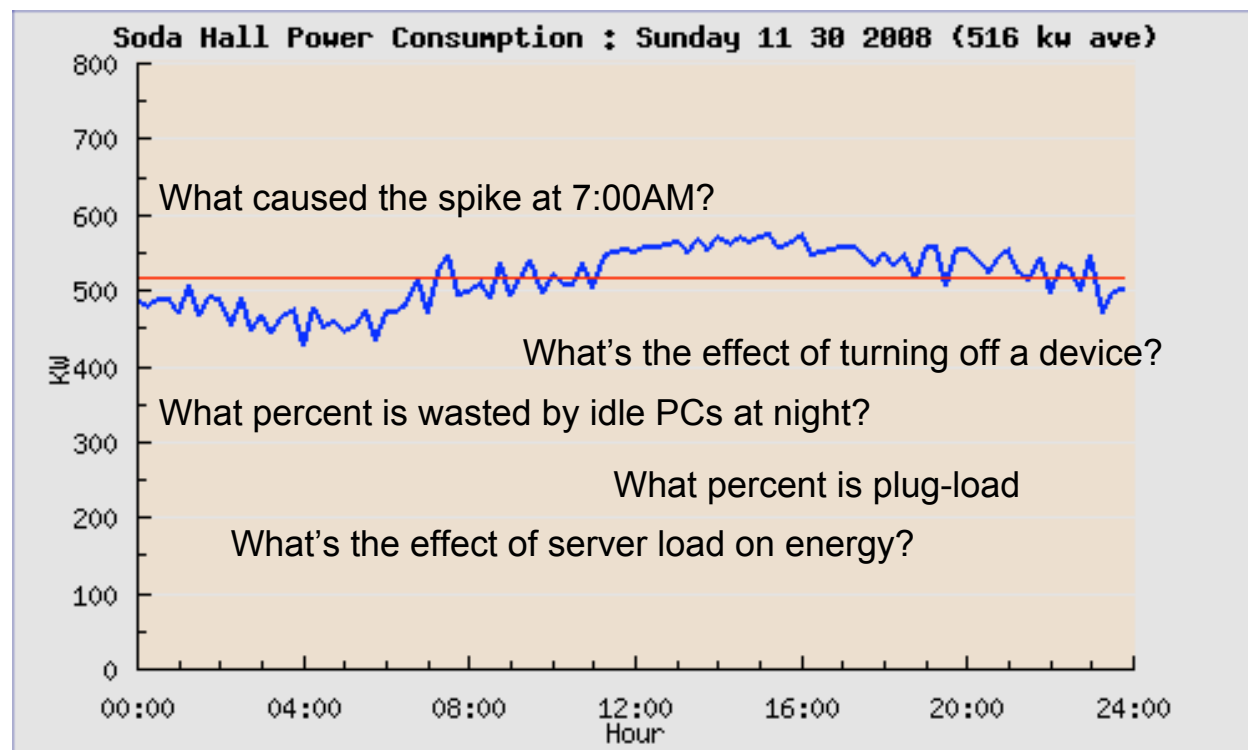


Figure 1.1: Aggregate energy usage statistics are insufficient for identifying waste or helping to reduce consumption.

Research has shown that providing feedback on electrical usage to occupants can reduce energy consumption by 5-15% [36][65][31]. However, existing building monitoring infrastructures are insufficient to identify energy waste or guide improvement because they only provide usage statistics in aggregate, both spatially and temporally. For example, Figure 1.1 shows the power consumption graph of the Computer Science building at UC Berkeley for one day. While it shows the overall envelope, daily average, and power variations, it provides very little information that can actually be exploited for reducing consumption – e.g., we don't know what caused the rise at 7:00AM, what's the effect of a particular action on power usage, what percent is wasted by idle PCs at night, or what percent is plug-load.

The hypothesis of my thesis is that *a high-fidelity energy monitoring architecture that provides fine-grained real-time visibility into building energy consumption enables significant and sustainable energy reduction by feeding back actionable views of energy uses to occupants.*

1.2 Architecture Overview

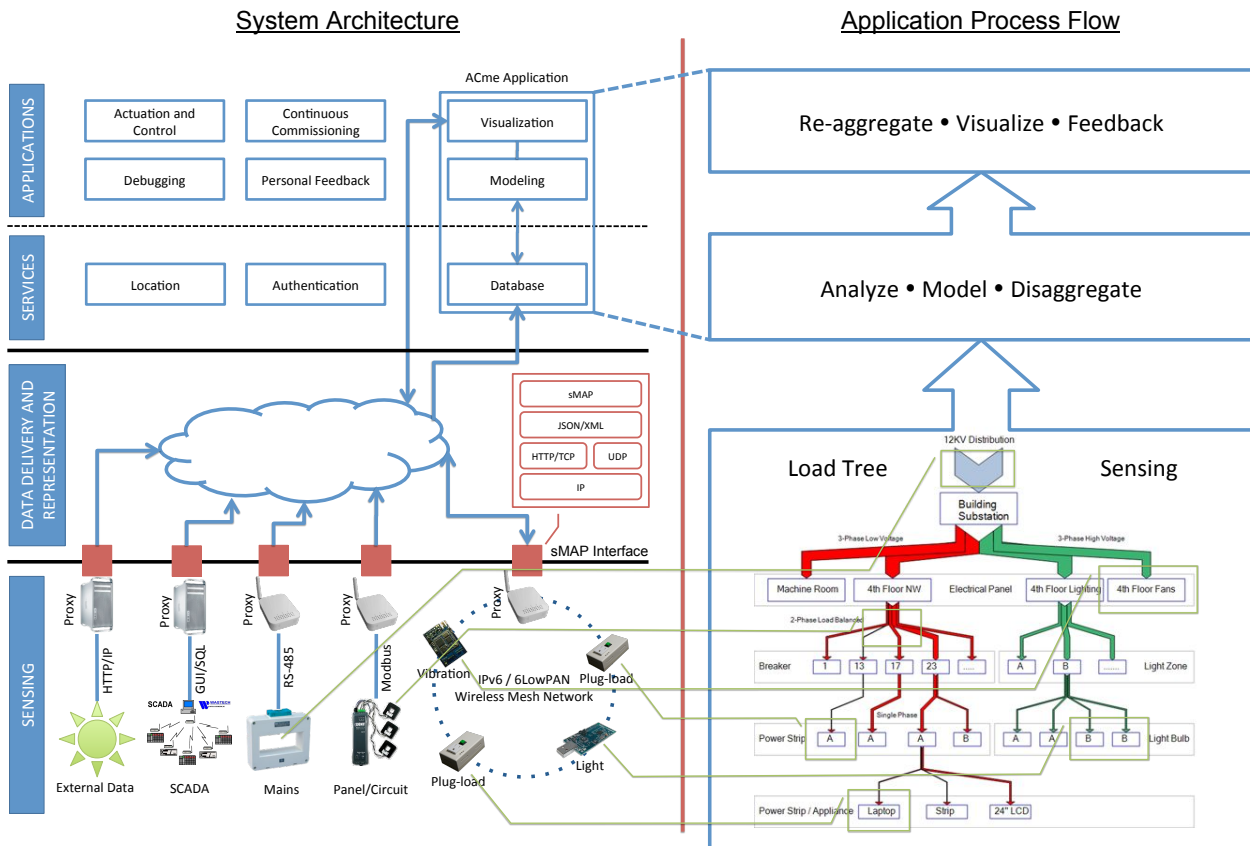


Figure 1.2: System architecture and application process flow.

Our solution consists of several parts, including tools for directly monitoring power usage, a three-tiered architecture that decouples data sources from applications, and a process flow for creating actionable views of usage statistics that lead to reduction.

This architecture is composed of three layers – “sensing”, “data delivery and representation”, and “service and applications”, as seen on the left of Figure 1.2.

To understand where and how energy is distributed and consumed, we must first measure it. The *sensing* layer is therefore fundamental to this architecture. In complex environments such as a commercial building, diverse classes of sensors, meters, and actuators are spread throughout the building, monitoring thousands of sense points at various granularities.

With such a diverse set of data sources, for applications to communicate with all of them becomes labor-some, if not impossible. The *data delivery and representation* layer functions as a narrow waist, decoupling portable application and services from the sensing layer below. In this architecture, sensors, meters, actuators, and other data sources communicate over IP

and use a standard application profile for uniform access.

Sitting on top of this uniform data representation layer are portable *services and applications*. Different applications can utilize the data layer and service layer below in innovative ways, without being hampered by the low-level details of disparate monitoring instruments and protocols.

Alongside this architecture, we also propose an application process flow, shown at the right side of Figure 1.2, for creating models using data obtained through this architecture, and generating actionable feedback that leads to energy reduction. We begin by understanding the underlying electrical distribution graph of the building, a tree-shaped network of electrical flow (load tree). Ideally we would like to instrument every point in load tree; however, this is typically cost prohibitive. Instead, we place appropriate types of sensors at strategic points in the load tree and create models for the most important and/or prevalent loads. These models of loads, such as appliances, lights, and HVAC systems, allow us to infer consumption of unmetered loads. Once we disaggregate power usage to the end-uses, we then re-aggregate according to different classifications, such as function, location, and ownership. Finally we feed back this data to the occupants and building managers via various types of visualizations for reduction.

The ultimate validation of this work would be a significant reduction in building-wide energy usage. Ideally we would like to conduct a user study at scale and observe the resulting improvements. However, while our architecture is designed to support a wide range of building applications for this purpose, it is impractical to conduct a comprehensive behavior study. Instead, we evaluate our solutions based on the following metrics and criteria that are more constrained:

- How reliable and accessible is the building energy data provided by this architecture?
 - How accurately does the power meter resolve power variations of appliances?
 - How easy is it to instrument a large number of plug-loads and how accessible is the energy data?
 - How scalable is this architecture with respect to sensors and meters?
 - To what extent can existing and legacy building instruments be incorporated in this system?
- How much effort does it take for third party developers to develop portable building applications using this architecture?
 - How interoperable is this architecture with external devices and applications?
 - How expressive or restrictive is the interface for physical information?
 - How cross-platform are applications built on top of this architecture?
- How actionable are the visualizations in facilitating energy saving?

- How much additional coverage can we gain by analyzing and modeling raw sensing data?
- What is the range of visualizations that can be generated following the proposed process flow?

1.3 Roadmap

In Chapter 2, we begin by understanding how energy is distributed inside a building in its various shapes and forms, followed by an ontology of existing building monitoring and control solutions. We take a critical look at some sensor network platforms and systems relevant to buildings, and present a survey of existing building energy modeling techniques and software.

In Chapter 3, we treat the building as a system to better understand the inputs and outputs of this large complex system with multiple sub-systems. Next, we present the design considerations and overview of a high-fidelity energy monitoring and feedback architecture for buildings, consisting of three layers – “sensing”, “data delivery and representation”, and “applications and services”. The subsequent chapters describe each layer of this architecture in more detail, grounded by the design, implementation, and evaluation of a system based on this architecture.

In Chapter 4, we explore in more depth the first layer in our architecture – *sensing*. We first describe the design, implementation, and evaluation of ACme – a wireless IPv6-based plug-load meter we designed to measure electrical usages of appliances at the plug-load level. We then explore other types of sensors, such as light sensors and vibration sensors, for inferring electrical consumption when direct measurement is inconvenient or costly.

In Chapter 5, we first describe an IPv6-based ad-hoc network that provides wireless connectivity to ACme nodes. We then describe a simple RESTful web-service-based approach, called sMAP, that uniformly represents physical data for a diverse class of sensors, meters, and actuators. sMAP provides a horizontal layering, decoupling the physical sensors, meters, and actuators from applications above.

In Chapter 6, we use the data acquisition portion of the ACme application as a demonstration of the RESTful interactions between the sensing layer and the services and applications layer, similar to any N-tier application. We defer the data analysis, modeling, and feedback of this data in Chapter 7 and Chapter 8.

In Chapter 7, we describe strategies that use multi-modal data collected throughout our network and the structure of the load tree to construct appliance electricity consumption models, enabling us to infer consumption of unmetered devices.

In Chapter 8, we demonstrate that by re-aggregating energy data according to different classifications, such as function, location, and ownership, we can reveal useful information and provide better feedback for reduction. Next, we show various ways for visualizing energy data, including an online web portal for tracking individual loads, a Google PowerMeter

application that interfaces with our system for tracking personal energy footprint, and an Android application that visualizes and controls ACme loads directly on mobile phones.

In Chapter 9, we describe a deployment based on our architecture, consisting of 50 ACme's, a few light sensors and a vibration sensor. We show that using this system, we are able to visualize high-fidelity energy usage in real-time and reduce overall consumption.

In Chapter 10, we describe some of the ongoing deployments and future research extending from this thesis.

In Chapter 11, we conclude with key results and contributions.

Chapter 2

Background

Commercial buildings have evolved in the past few decades into complicated systems with thousands of sensors and actuators that continuously monitor and control the micro-climate inside the building. Building monitoring, control, and management systems have become a multi-billion dollar industry over the years. Electronically-controlled and computerized building management systems became popular as early as the late 1960's and early 1970's, with many well known companies such as Honeywell and Siemens. The focus of these commercial systems has been on providing comfort and safety to the occupants. As energy is becoming more scarce and expensive, there is renewed interest in saving energy in buildings. Additionally, we are starting to see many opportunities where wirelessly networked devices can help us better understand and management building energy consumption. In this chapter, we outline key results and places for improvements. We begin by understanding how energy is distributed inside a building in its various shapes and forms in Section 2.1, followed by an ontology of existing building monitoring and control solutions in Section 2.2. We take a critical look at some sensor network platforms and systems that are relevant to buildings in Section 2.3, followed by a survey of existing building energy modeling techniques and software in Section 2.4.

2.1 Energy Flows in Buildings

Energy flows into buildings primarily in the forms of electricity and natural gas. The US Environmental Protection Agency (EPA) and US Department of Energy (DOE) have commissioned multiple studies to trace the end use of energy in building back to the original energy sources, as shown in Figure 2.1, taken from the 2006 PNNL study [29]. According to this study, commercial buildings consume 8.42 quadrillion btu (abbreviated as quads hereafter) of energy in end-use, 14.51 quads in natural resources (including a small amount of nuclear and hydroelectric), and loses 6.25 quads during conversion and transmission of electricity. Of the 14.51 quads of raw natural resources consumed, roughly 62% is in the

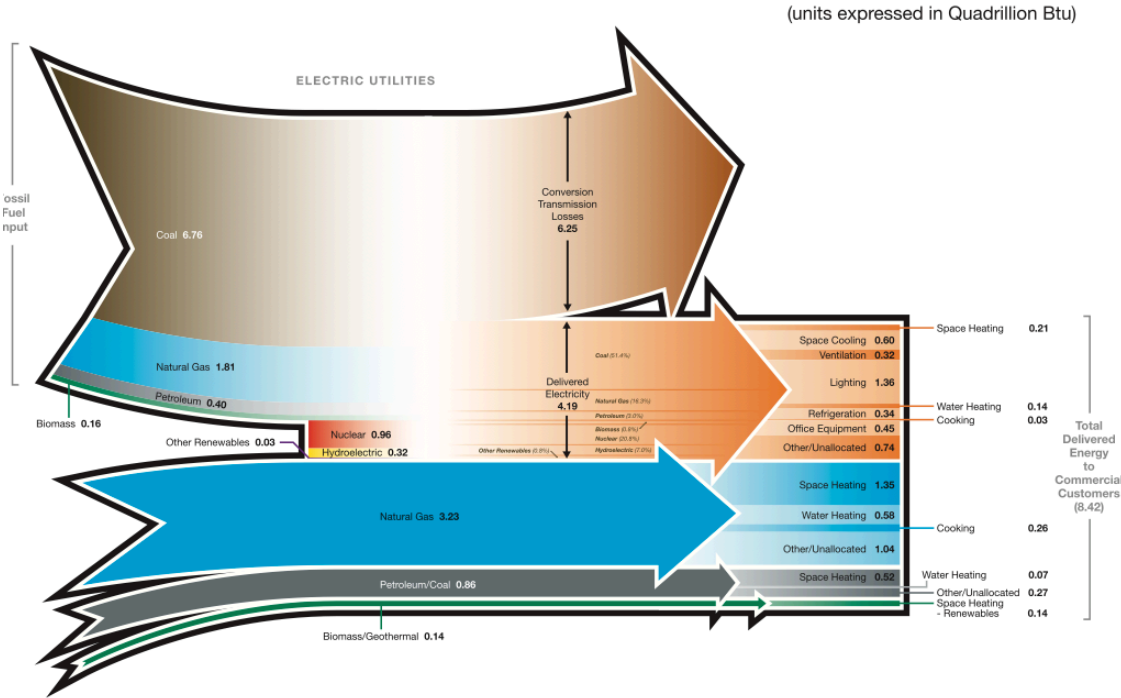


Figure 2.1: Energy end-use flow maps for the commercial buildings. Image source: [29].

form of electricity and 29% in natural gas. Lighting and cooling make up the majority of the electrical load while heating is the major gas load. This figure provides us with an understanding of the scale of energy consumption and loss in commercial buildings and a breakdown of end-use. While such graphs help establish the case for the need to reduce energy consumption in buildings and identify major areas for concentrating efforts, they do not indicate what actions to take. For that we need to really understand where energy is used, consumed, and wasted, which requires diving inside the building and understand how these energy is distributed to the end-uses. In the rest of this thesis, we focus primarily on electrical energy, which represents the majority of commercial building energy usages.

2.1.1 Electric Flows

The primary form of energy flowing into buildings is electrical. Figure 2.2 shows an example of an electrical distribution diagram inside a typical commercial building. In this example, a high voltage distribution line (12kV) from the neighborhood substation connects the building substation to the grid. Once inside the building, the building substation steps down the voltage into three phase high voltage and three phase low voltage, distributed throughout the building. High voltage power (480V, 3 phase) is routed to the HVAC systems such as pumps, fans, and chillers, and into high voltage distributions panels on each floor

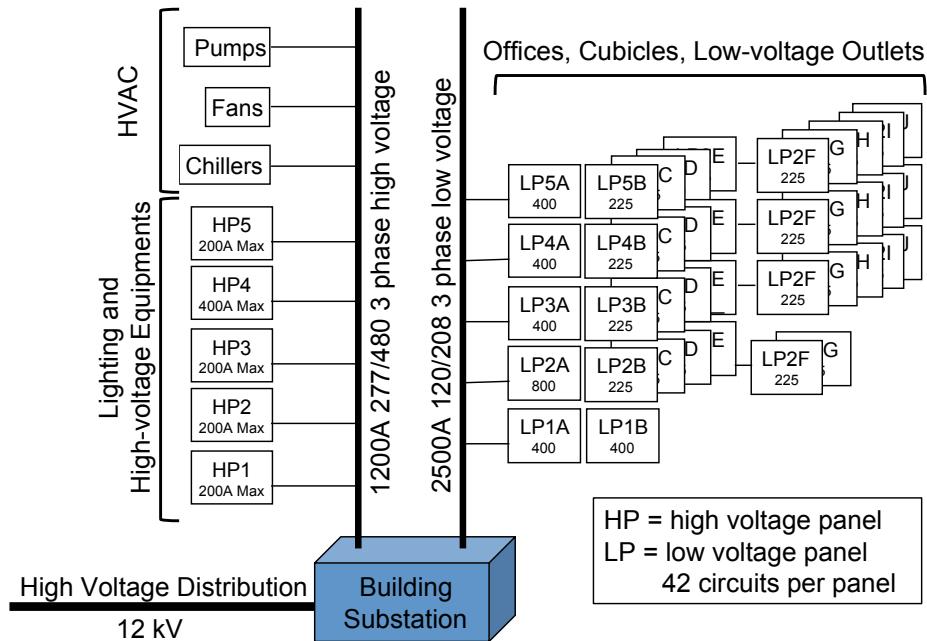


Figure 2.2: Example of an electrical distribution diagram inside a typical commercial building.

(labeled as HPx) to power lighting loads on the ceilings. Certain equipments also require high voltage, such as elevators. On the right side of this diagram, low voltage power is distributed throughout the building into low voltage panels (LPxx) on each floor. One floor often have multiple panels, either directed off the substation transformer or from another panel. Each panel usually contains 42 (single phase) *circuits*, which distributes power into all the low voltage loads spread throughout the building, often in the form of electrical outlets. Because the electrical distribution diagram can usually be viewed as an acyclic graph where a *parent node* is connected to multiple *children nodes*, as in the case of this example, it is also referred to as the *load tree*.

The *load tree* is the starting point for monitoring energy usage with high-fidelity and in real-time, and is central to our architecture. We can readily see that the load tree is a complicated structure with thousands of loads attached to its leaves. While directly measuring each load provides us with the most detailed view, it is impractical and costly. We often give up this level of fidelity by measuring at some aggregation point, such as at the panels or even at the building level, trading fidelity with coverage. We describe this trade-off and explore ways to improve fidelity in more detail in Chapter 7.

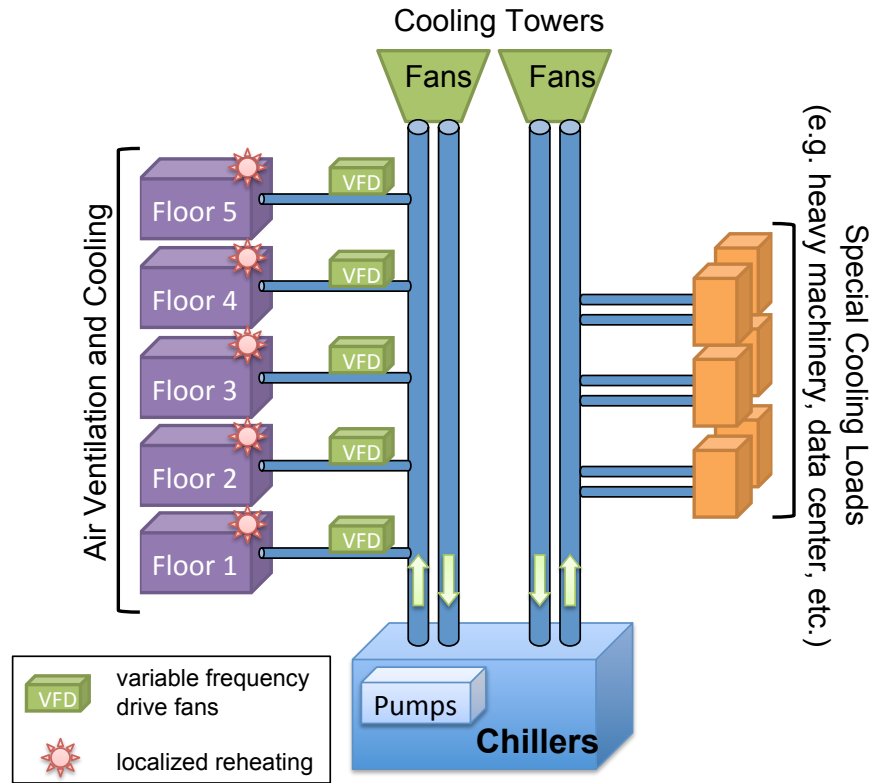


Figure 2.3: Example of climate plants inside a commercial building.

2.1.2 Climate Plants

One of the major loads on the electrical load tree is to service the climate plant, which includes chillers, pumps, air conditioning units, air handling units, fans, etc., as shown in Figure 2.3. In this example building, chillers cool the air using chilled water; cold air is delivered throughout the building with fans. The amount of cooling is controlled by vents and variable frequency drives (VFD), and is reheated locally where needed. This is not a very efficient HVAC system because the operating point of the air conditioning units is determined by the needs of the worst heat load. This example shows that there are many opportunities for savings in the HVAC system, if we are able to observe and control it.

2.2 Current Building Energy Monitoring and Management Solutions

Various forms of building energy monitoring and management solutions have emerged over the years. The set of solutions and products range from whole-building management

systems that cost millions of dollars to single plug-load meters that can be found in local stores for tens of dollars, and from established companies to university research projects, or even hobbyists. In this section, we survey some of the prominent works and products in this area, and analyze their advantages and drawbacks with respect to what we need in a monitoring and feedback system for reducing energy consumption.

2.2.1 Building Management Systems

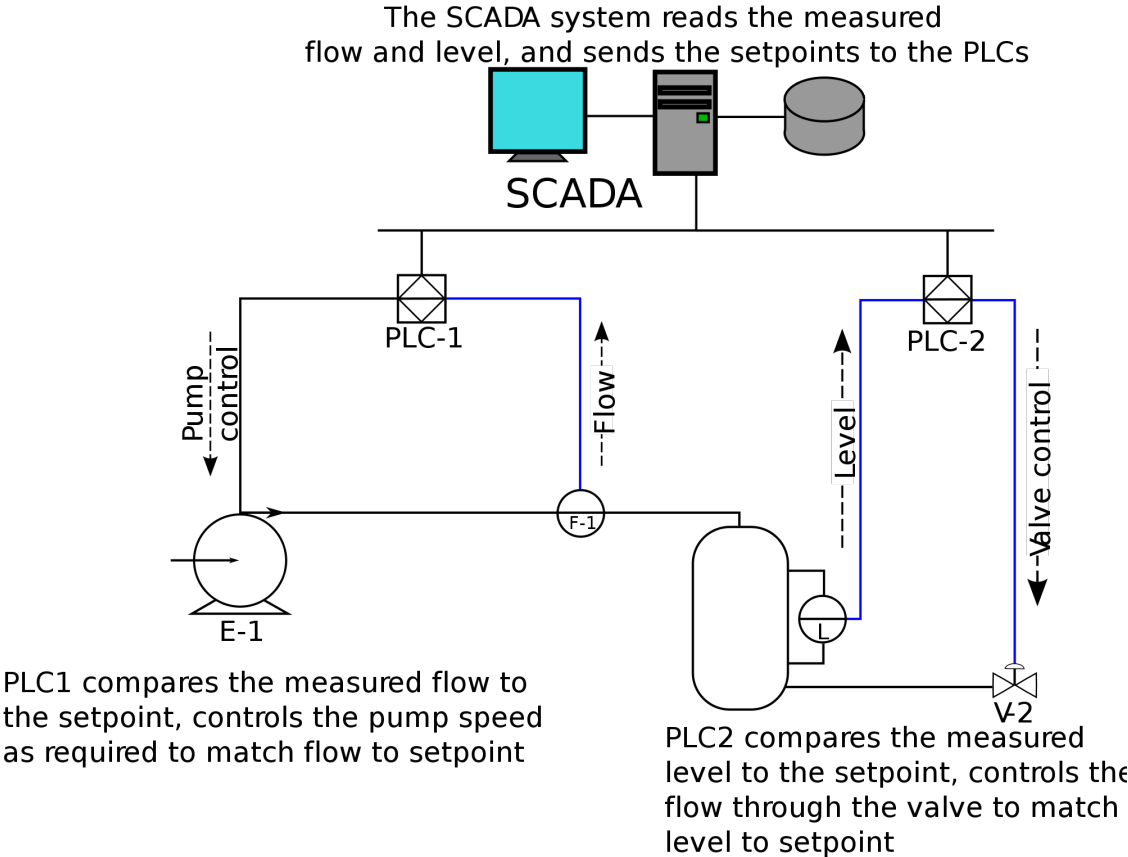


Figure 2.4: SCADA is a supervisory system where a central host controls set-points and remote terminal units (RTUs) perform local decisions. Image source:[82].

Computer-aided building management systems, such as Building Management System (BMS) [80] and Supervisory Control and Data Acquisition (SCADA) [82], are widely used today in commercial buildings to control physical processes. BMS is a software-hardware system often used in large buildings to controls and monitors the buildings mechanical and electrical equipment. Similarly, SCADA is a supervisory system where a central host controls set-points and remote terminal units (RTUs) perform local decisions, as shown in Figure 2.4.

BMS and SCADA are similar in architecture, and both have their respective ecosystems of hardware and software vendors [12][18][4][5][17], and a mixture of open protocols and proprietary protocols at various levels in the stack.

For example, BACnet [72] is an ISO standard protocol for communication widely used in BMS systems. BACnet defines a number of services used to communicate between building monitoring and control devices. It defines a set of services used for device and object discovery, including *I-am*, *who-has*, *I-have*, and etc. It also defines services for data sharing such as *read-property* and *write-property*, which operate on a set of 30 objects describing various sensor readings and parameters. BACnet further defines and communicates over a number of link layers, including ARCNET, Ethernet, BACnet/IP, Point-To-Point over RS-232, Master-Slave/Token-Passing over RS-485, and LonTalk.

Modbus [75] is a serial communication protocol often used to connect a supervisory computer with a RTU in SCADA systems. Amongst different Modbus versions, Modbus RTU is most common, and uses a compact binary representation of the data. Modbus defines a payload format for reading and writing virtual device “registers,” and is typically running directly on a link-layer such as RS-485 serial link or Ethernet. Messages are framed and separated by idle periods. Each frame consists of start/address/function/data/CRC-check/end.

For the rest of this chapter, we focus on SCADA since it is used in many buildings at UC Berkeley and representative of commonly used building management systems.

The SCADA architecture reduces the complexity of RTUs local to the physical sensors and actuators and minimizes communication overhead. It is a good solution in the age when computation and network bandwidth are expensive. However, it also has its drawbacks as a result. In addition, SCADA is climate-centric and is designed to optimize for comfort, not energy consumption. As a result, SCADA is not well suited for identifying energy savings or inefficiencies. We identify some of the problems with SCADA below:

- The main control loop in a SCADA system, and the only part involving humans, is through the Human-Computer-Interface (HMI), where the operator or building manager sets parameters and set-points. This is a system where the occupants, the primary load to the building, are not in the control loop except through thermostats. This system does not model the building according to the load tree, but instead focuses on local thermal loops. It is comfort-centric instead of energy centric.
- Sensors and actuators (RTUs) are tightly integrated inside vertical stovepipe solutions, communicating using legacy protocols such as MODBUS [75], and without an open interface for external communication. While some newer ones use SONET [74] and SDH [73] or even open protocols such as IP, the fundamental design remains the same – distributed RTUs communicate with a supervisory host via simple set-points, and data is not made available outside of the local SCADA system. This design approach prevents data from being consumed programmatically and impedes the proliferation of *portable building applications*.

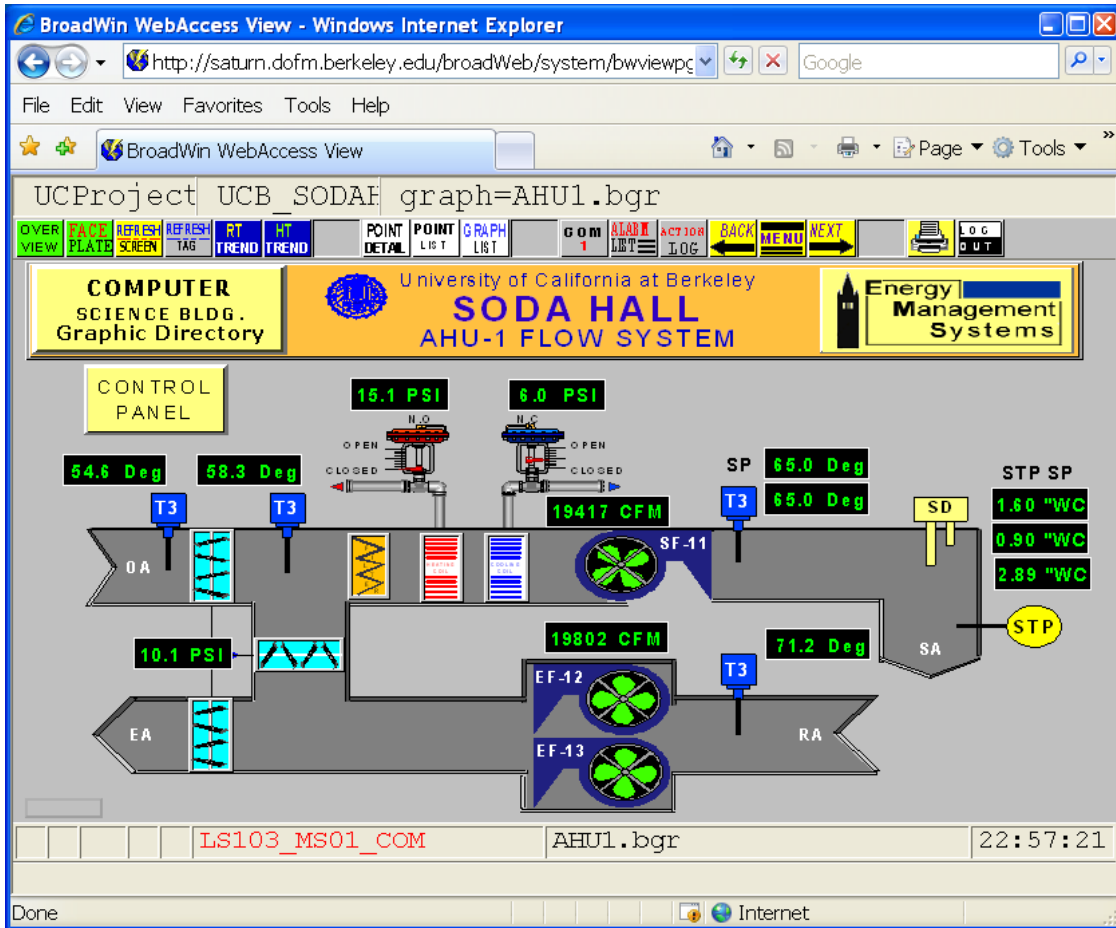


Figure 2.5: Broadwin WebAccess HMI displaying air flows parameters inside the computer science building at Berkeley.

- Often the only interface exported by SCADA is the HMI, which is often GUI based. For example, Figure 2.5 is a web-based HMI by Broadwin [4] SCADA system to display the air flows in the computer science building at Berkeley. While GUI is the right level of abstraction for humans, the information bandwidth is severely limited and response time is very slow; it is inadequate as an interface for other programs. Although data is available from a database, there is no easy way to directly communicate with the underlying sensors and actuators in real-time.

2.2.2 Plug-load Monitoring

With the recent increase in awareness of energy conservation, several commercial products have appeared on the market which measure single outlet energy consumption, commonly

referred to as plug-load meters [13]. These are helpful for point measurements, but still require manual measurement. Such an approach does not address the deployment challenges at scale. At the same time, there has been a tremendous amount of research and industrial effort in recent years that has made significant strides toward providing greater visibility. The MIT *Plug* [59] power meter platform provides high-fidelity apparent power measurements, which is useful for profiling a load over short and long time scales. Multi-modal sensing has also been explored in the literature [28]. In addition, industries have invested a significant amount of work towards improving building energy monitoring. Several startups, such as Tendril [21], Greenbox [8], and EnergyHub [9], have introduced ZigBee Home Profile-based wireless energy monitoring solutions. These products take a bottom-up approach by providing detailed power measurements of selected individual loads. While this approach is useful in observing a few loads at high fidelity, it is neither practical nor cost-effective when full coverage of tens or hundreds of appliances is desired. The area of wireless sensor networks has also made substantial progress in this application space. For example, Sentiilla [19] offers a data center energy monitoring solution that uses wireless plug-load meters and interoperates with other types of sensors. Arch Rock [3] offers a sub-monitoring solution for commercial buildings that uses wireless branch level meters. Kim et. al. have developed methods to infer power usage using non-intrusive means such as magnetic sensors [54], and proposed a framework to profile personal resource consumption using a combination of resource monitoring and activity monitoring [53].

2.2.3 Power Disaggregation

An alternative to extensive sub-metering is to place the sensing instrument at the root of the *load tree*, and use algorithms to increase visibility by disambiguating an aggregated load from the top down. For example, many utility companies have introduced AMI programs that provide near-real-time visibility into the aggregate energy consumption of homes. Some utilities are partnering with aggregators such as the Google PowerMeter [7] project and the Microsoft Hohm [15] project to provide a rich visual feedback of user energy usage at the household level. Some utilities have incorporated “bill disaggregation” web applications that break down users’ monthly bills by disaggregating the different types of loads from their aggregated energy traces. This type of approach was originally proposed by Hart in 1992 [44]. He proposed disaggregating individual electrical loads based on real and reactive power measurements. The approach is feasible for a small number of loads that have distinguishable differences in power factor. Norford et al. improved this method with event detection to help disambiguate appliances with similar reactive and real power signatures. More complex algorithms have been developed and have shown improvements [56], [40], [69], [27], [57], [58]. However, this approach is generally less effective in an office environment in which many loads are based on switched power supplies, such as desktops, laptops, and LCD screens. Patel has demonstrated techniques for detecting and classifying electrical loads by using a single plug-in sensor to monitor noises on the power line [66]. This approach is promising

but requires a training period and not yet feasible in large commercial environments where similar loads are densely located.

2.3 Sensor Networks

Sensor network research has made great progress over the past decade in creating low-power wireless embedded devices, systems, and networking technology, and this technology has been applied to numerous studies of real-world environments. Many of these, including habitat and climate monitoring demand extremely low energy consumption, long lifetime on batteries, low sample rates, and reliable ad-hoc networking in wide open spaces. In our work, we focus on a large scale wireless sensor network for a very different kind of application—high fidelity, building-wide, electricity monitoring.

2.3.1 Hardware Platforms

Hardware platforms for wireless sensor networks have evolved from early experimental hardware used by a few universities to relatively mature products that can be purchased individually or as part of larger commercial solutions. However, the basic tradeoff between resources and lifetime has not changed for the better half of this decade. Increased hardware resources such as CPU speed, radio bandwidth, and memory are desirable for various applications that require such resources. Unfortunately, they come at the cost of higher energy consumption, reduced lifetime, and bigger form factor.

Figure 2.6 shows the evolution of some of the platforms developed at Berkeley. In comparison to modern computers, this class of devices has very limited resources, on the order of a few kilobytes of RAM and a few megahertz of processor.

In addition to the main processing and communication unit, mote platforms are usually augmented by application-specific sensing boards that perform the actual sensing functionalities required by the application. For example, on the left of Figure 2.7 is an ultrasound ranging board I developed for the Mica and Mica2 platforms that uses ultrasound transceivers to measure distance [79]. This sensor board is attached to the Mica mote via a 51-pin connector; the CPU on the Mica mote communicates with the ultrasound board via GPIO pins and an I2C bus. The ultrasound transceiver triggers an interrupt when it receives a ultrasound signal which is used to calculate distance using time-of-flight measurements. In working towards using sensor networks for high-fidelity power measurement of low-power devices at scale, I developed the *SPOT* platform [52], as shown on the right of Figure 2.7. *SPOT* is a sensor board for the MicaZ platform that uses a combination of analog and digital techniques for metering power at the micro-amp resolution and with a dynamic range of 45000:1. This work provided us with valuable experience in designing sensor network platforms for monitoring electrical energy.

However, as wireless sensor networks become more mature and applications more diverse,








Mote Type Year	<i>WeC</i> 1998	<i>René</i> 1999	<i>René 2</i> 2000	<i>Dot</i> 2000	<i>Mica</i> 2001	<i>Mica2Dot</i> 2002	<i>Mica 2</i> 2002	<i>Telos</i> 2004	
									
Microcontroller									
Type	AT90LS8535		ATmega163		ATmega128			TI MSP430	
Program memory (KB)	8		16		128			60	
RAM (KB)	0.5		1		4			2	
Active Power (mW)	15		15		8		33	3	
Sleep Power (μ W)	45		45		75		75	6	
Wakeup Time (μ s)	1000		36		180		180	6	
Nonvolatile storage									
Chip	24LC256				AT45DB041B			ST M24M01S	
Connection type	I ² C				SPI			I ² C	
Size (KB)	32				512			128	
Communication									
Radio	TR1000				TR1000	CC1000		CC2420	
Data rate (kbps)	10				40	38.4		250	
Modulation type	OOK				ASK	FSK		O-QPSK	
Receive Power (mW)	9				12	29		38	
Transmit Power at 0dBm (mW)	36				36	42		35	
Power Consumption									
Minimum Operation (V)	2.7		2.7		2.7			1.8	
Total Active Power (mW)	24		27		44	89		41	
Programming and Sensor Interface									
Expansion	none	51-pin	51-pin	none	51-pin	19-pin	51-pin	10-pin	
Communication	IEEE 1284 (programming) and RS232 (requires additional hardware)							USB	
Integrated Sensors	no	no	no	yes	no	no	no	yes	

Figure 2.6: Evolution of early mote platforms developed at UC Berkeley. Source: [68].

the tradeoffs are no longer single-dimensional. For example, we are beginning to recognize the importance of shorting turn-a-round time between hardware cycles, leading to a platform design methodology that separates “expert modules” that can be easily recomposed from “carrier boards” that are application specific [38]. In the building monitoring area, we are starting to see many opportunities for energy scavenging, such as directly powered by AC power lines instead of relying on batteries. At the same time, we are confronted with new issues such as safety, form factor, and cost. We address some of these new challenges in Section 4.1.

2.3.2 Networking

Because control loops require communication with individual devices, collection-only protocols such as MultiHopLQI or CTP [84, 42] were not viable options. CentRoute [71] provides communication with single routers in the network, but this arouses concerns that its static routing decisions may not be sufficiently agile in poor RF environments. Existing open and standardized point-to-point routing protocols for wireless networks such as AODV and DYMO are not appropriate for this class of networks due to high control overheads

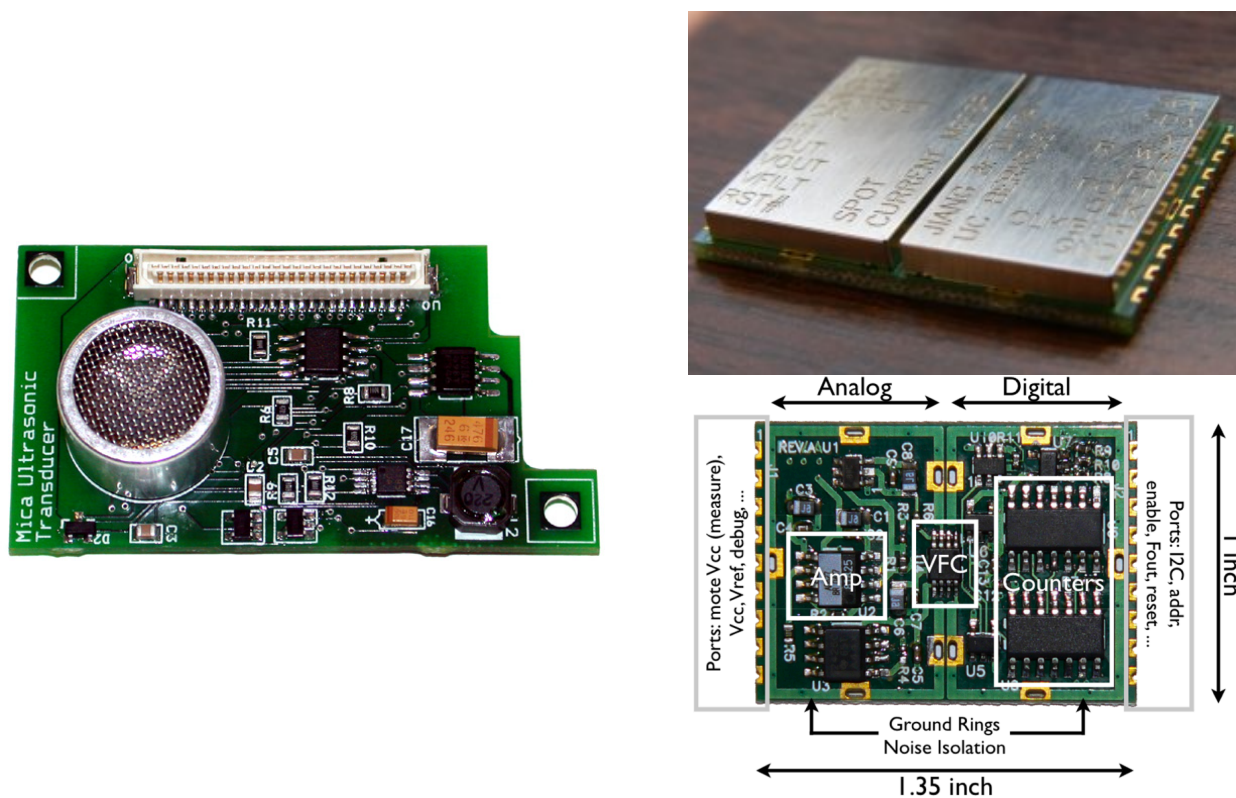


Figure 2.7: Ultrasound transceiver sensor board for measuring distancing using ultrasonic time-of-flight (left); *SPOT* micro-power meter for high-fidelity instrumentation of low power devices at scale (right).

or large routing state [67, 30]. While S4 appears promising, actual use of it would require the design and implementation of a location service [60]. Proprietary stacks such as TSMP, WirelessHART, and PhyNet [20, 23, 11] also seem worthwhile but are not widely available.

2.4 Building Energy Modeling Software

Building modeling software is widely used by architects, civil engineers, and building managers both during the construction of buildings and post-construction analysis. Models range from simple architectural schematics with room layouts and building forms to detailed modeling of glare, glass specifications, and insulation material R-values; and from HVAC models that help simulating system responses to building loads to whole building energy analysis on energy costs and carbon footprints.

Current building modeling tools can be divided into the following types [24], each with different uses and target audiences.

- Daylighting / lighting modeling
- Computational fluid dynamics (CFD)
- Building component analysis
- HVAC analysis
- Building thermal analysis
- Whole building energy simulation programs (BESP)

Among these, whole building energy simulation (BESP) is perhaps most relevant to this work. There are a number of free and commercially available BESP software, such as DOE2 [6], EnergyPlus [76], TRNSYS [22], Ecotect [25], and Energy10 [63]. DOE2/eQuest is well validated with research and has a good interface for detailed modeling; EnergyPlus has more advanced modeling capabilities with a modular programming structure but currently lacks a good graphical interface; Ecotect has a reasonable 3D modeling interface with nice graphical results viewer, and can be exported to powerful simulation tools such as Radiance and EnergyPlus. As energy becomes a focal point in the building CAD communities, several companies have offered energy simulations based on 3D architectural renderings, such as Green Building Studio [26] offered by Autodesk.

While building energy modeling software is moving in the right direction, it is still missing some critical components. Current modeling software uses static models with pre-determined parameters instead of using real-time sensing data as an input to continuously improve modeling parameters. They also do not model appliances or end-uses, nor occupants themselves, which have significant impact on the overall building energy. Furthermore, while the simulation results are useful for architects and building managers, they provide little useful information that can be easily acted upon by occupants. Through this work, we hope to create an architecture that can enable these types of dynamic simulation of whole building energy in real-time.

2.5 Summary

Buildings are large complex systems comprised of thousands of sensors and actuators, however, it is difficult to access these data sources because existing building management systems hide them behind vertical stacks with legacy and proprietary protocols. Furthermore, because current systems optimize for comfort instead of energy efficiency, little emphasis is placed on directly instrumenting the electrical load tree. As a result, occupants do not have visibility into the building's energy usage, and cannot act to reduce their consumption.

In addition, direct monitoring of plug-load power provides visibility into the end-uses, and is valuable to both occupants and building managers. While there are several commercial plug-load meters, they either do not provide adequate measurement fidelity, or are

not scalable in large deployments. Recent research efforts have shown initial results in indirect monitoring of appliance power usages, but are not yet feasible in large commercial environments.

Finally, while there are a number of building modeling applications that simulate various aspects of building's climate and energy characteristics, the fidelity of their models is at the room granularity. They do not model appliances or end-uses, and therefore do not provide sufficient fidelity.

Chapter 3

System Architecture

With a better understanding of energy flows in buildings and an overview of existing monitoring and management solutions in the previous chapter, we now take a more systems view of the building to better understand how we might approach this problem using computer science techniques. We decompose the building into three interconnecting sub-systems – “human-behavior”, “climate HVAC”, and “electrical load tree” – and investigate their relationships in Section 3.1.

In section 3.2, we present the design considerations and overview of a high-fidelity energy monitoring and feedback architecture for buildings, consisting of three layers – sensing, data delivery and representation, and applications and services. The subsequent chapters describe each layer of our architecture in more detail, supported by the design, implementation, and evaluation of a system based on this architecture.

3.1 Building as a System

A building is fundamentally a space that controls environment for the comfort of its occupants and provides the infrastructure to support their activities. To provide these two functions, a typical building has a physical plant, responsible for generating the internal climate, and an electrical load tree, responsible for routing electricity. To better understand the building as a system, we decompose it into three interlocking views: environment and activity, climate plant, and electrical load tree, as shown in Figure 3.1. The environment and activity view describes building occupancy and occupant activities, the fundamental *load* or input to the building as a system. The climate plant view describes the physical plant inside a building, which often includes heating, cooling, ventilation, water and wastewater, and is a function of the environment and activity view. To power the physical plant, a large amount of electricity is typically required, which must be supplied by the electrical load tree. The occupants also directly impact the electrical load tree through appliances and plug-loads. In this decomposition, the load tree is a central component, but it is often the least visible in

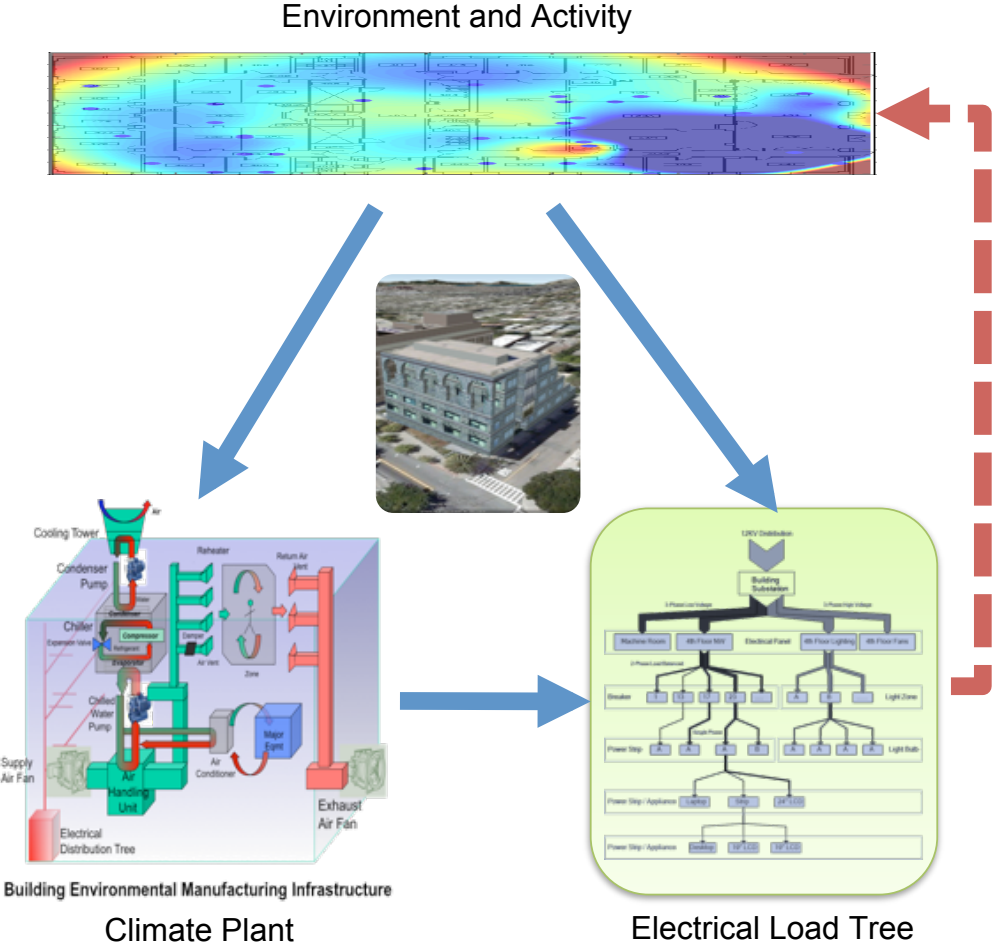


Figure 3.1: Existing buildings are often missing the feedback loop between the electrical load tree and the occupants.

today’s buildings. Occupants rarely have the ability to observe the electrical load tree, and cannot act based on its status.

We can formalize this system by qualifying the three components separately as individual systems with inputs and outputs, as shown in Figure 3.2. Using this energy-centric view of the building system, we hope to gain an insight into how energy is distributed, used, and wasted within a building. The first part of this system is labeled as “Human Behavior”. This describes the human-generated load, on, and external to, the building’s infrastructure. The input to this system fluctuates according to various external stimuli such as occupancy, seasonal changes, work schedule, weather, and etc. The output of this system, $S(t)$, is a direct input to the climate and HVAC of the building, including heat, air flow, and water usage. The output also impacts the electrical load tree through direct electricity usages such

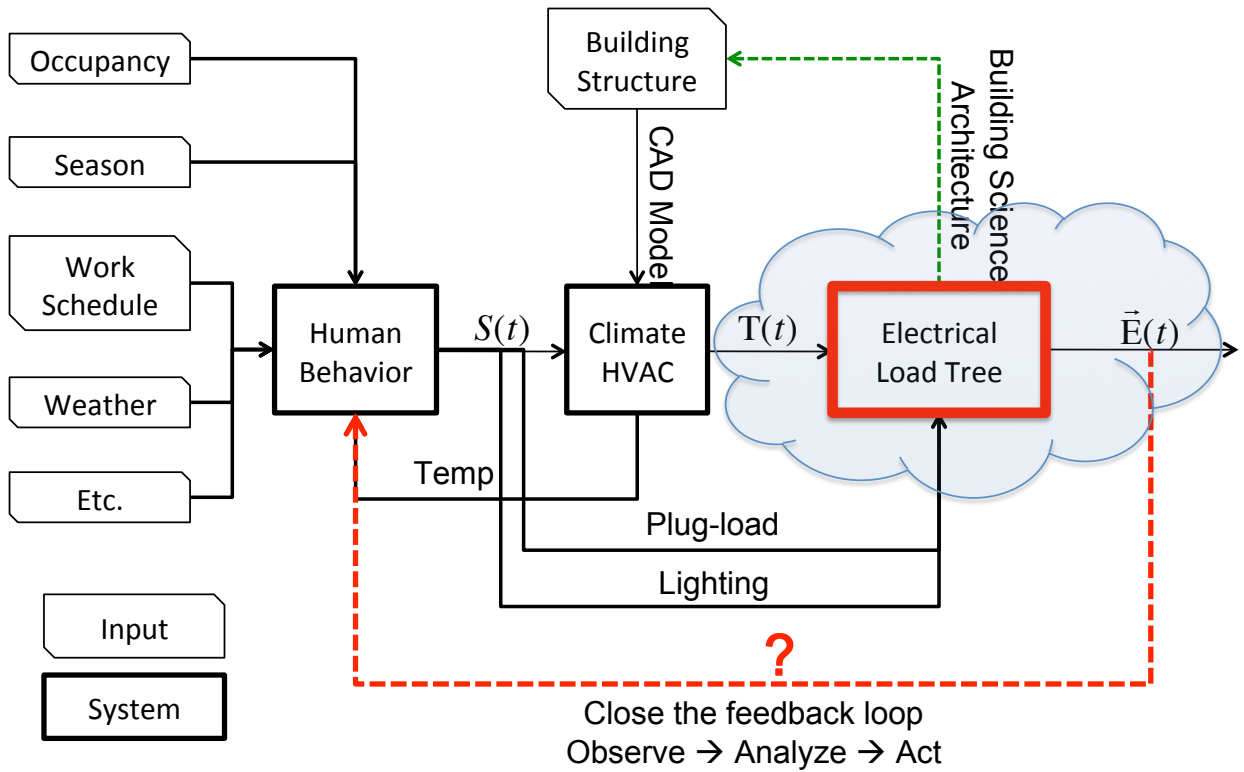


Figure 3.2: Building as a system, is composed of three subsystems – “human behavior”, “climate HVAC”, and “electrical load tree”.

as elevator, lighting, and plug-loads. The system labeled “Climate HVAC” is not only a function of the outputs of “Human Behavior” but also of the structure of the building itself. And in many cases, optimizing the building structure can have a significant impact on the “Climate HVAC” system, as seen in [49]. For the purpose of this thesis, we assume the building structure is not something we can control. The output, $T(t)$, of this system, is the load, and input to the “Electrical Load Tree”. In many buildings, $T(t)$ is the largest energy use.

As seen in this figure, the electrical load tree is perhaps the most important system inside a building. It is directly impacted by all electrical uses, either directly or via the “Climate HVAC” system. Observing the electrical load tree is therefore essential in understand where energy is consumed and wasted, and is a necessary step in reducing consumption. However, in today’s buildings, it is often the least observable. The only feedback buildings have to occupants is through thermostats, as signified by the feedback loop from “Climate HVAC” to “Human Behavior”, labeled “Temp”. However, there is little or no feedback from “Electrical Load Tree” to “Human Behavior”, resulting in an open-loop system, vulnerable to instabilities in the input.

Current efforts such as demand response [64] try to create a direct feedback, but only kicks a few times during a year. We argue that to truly reduce energy consumption, we should, in addition to *peak-shaving* once or twice in the summer, focus on reducing the *baseline continuously* by treating the building as a holistic system. We propose an architecture for reducing energy consumption in buildings that starts by providing the facilities for directly observing the load tree, followed by applying tools to analyze, model, and simulate the end-use, and ultimately leading to the realization of energy savings, either through feedback to the occupants or automatic methods.

3.2 System Architecture Overview

Learning from the short-comings of existing building management systems (Section 2.2.1), and borrowing from traditional software tiering, we propose a three-tiered architecture for monitoring and reduction of energy in buildings with following properties:

- Abstracting heterogeneous data sources using an uniform data representation layer, and enabling open¹ and programmatic access by portable applications.
- Instrumenting the whole building as a real-time distributed system centering around the electrical load tree, instead of human-based control of set-points.
- Enabling occupants to reduce energy voluntarily by coupling them in the control loop and providing them with visualizations of energy usage.

This architecture is composed of three layers – sensing (Chapter 4), data delivery and representation (Chapter 5), and service and applications (Sections 6.1 and 6.2), as shown on the left side of Figure 3.3.

To reduce energy in buildings, we must first understand where and how energy is distributed and consumed. Empirical measurement of energy is therefore paramount in this architecture, and is the first layer in our architecture – **sensing**. In complex environments such as a commercial building, diverse classes of sensors, meters, and actuators are spread throughout the building, monitoring thousands of sense points at various granularities. We identify several prominent classes of devices that belong to this layer in the bottom-left of Figure 3.3.

In this figure, a busbar Rogowski coil (a flexible type of current transformer) is used to monitor the *mains* power at the building base-station level, using RS-485 as the link protocol for communication. At the *panel* level, current transformer based circuit meters are used for monitoring individual breakers in panels, and communicate using Modbus over Ethernet. Wireless building management systems are also becoming increasingly common,

¹Open does not imply insecure. There are many encryption protocols such as SSL/TLS, that enable secure communication on top of open standards.

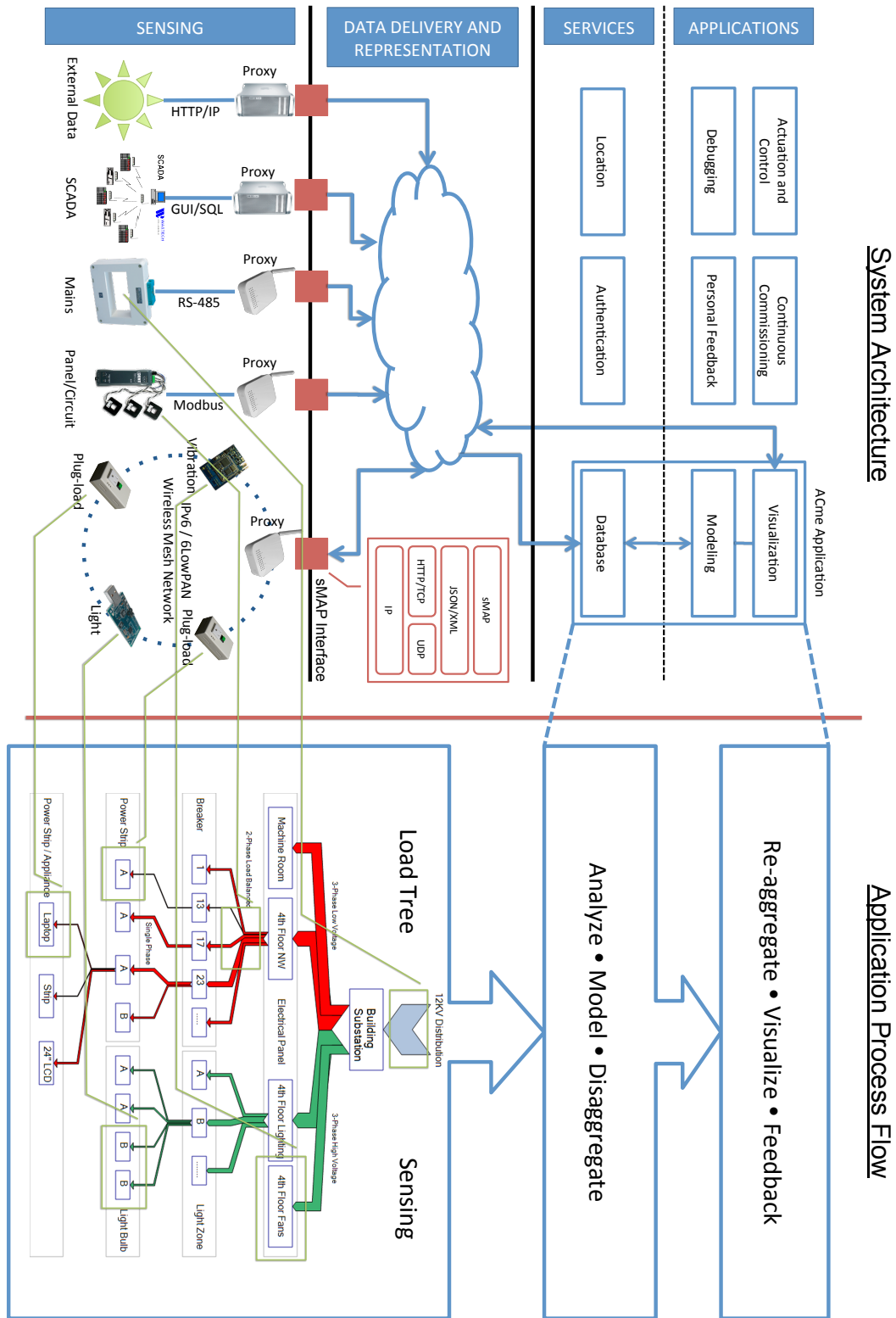


Figure 3.3: System architecture and the process flow of the ACme application.

based on WirelessHART, ISA 100.11a, or the Zigbee/HomePlug Smart Energy Profile 2.0 (SEP2) [83, 48, 70]. With intelligent and connected devices becoming increasingly cheaper and ubiquitous, *smart AC plug-load meters*, *light sensors*, or even *vibration sensors* could be used to monitor various physical phenomena inside the building that help in inferring energy usage (some of these sensors and meters are described in more detail in Chapter 4). Furthermore, sensing is not limited to physical sensors. *External data* such as weather patterns, occupancy fluctuations, and etc. are all valuable information, and is treated simply as data sources in the sensing layer. In addition, we can also abstract *SCADA data* simply as another data source, and communicate with it through GUI or SQL.

Regardless of where these data come from or how they communicate internally, they can be easily exported via the use of proxies with network interfaces to provide IP connectivity. In this architecture, the **sensing** layer includes any device or data source that provide physical information and has the ability to communicate with external devices over IP, either directly or through proxies.

However, with such a diverse set of data sources, each speaking a different language, for applications to communicate with all of them is extremely complicated and difficult, if not impossible. As a result, a three layered approach is not only natural, but necessary, for enabling innovation in building applications without been hampered by the low-level details of the myriads of disparate monitoring instruments and protocols. The **data delivery and representation** layer functions as a narrow-waist, decoupling portable application and services from the *sensing* layer below. In our vision, sensors, meters, actuators, and other data sources will communicate over IP and use a standard application profile called sMAP – Simple Measuring and Actuation Profile. sMAP provides uniform access to a large subset of devices typically found in a building. This layer is the hardware abstraction layer; below it, code may be device-specific. For newer devices such as IP-based sensors, sMAP may be integrated directly; for legacy devices such as Modbus based meters, sMAP can be implemented on a proxy, as shown as red boxes in Figure 3.3. A typical sMAP proxy acts as a bridge or gateway. On one side, it implements the device-specific code for communicating with a particular sensor; on the other side, it implements sMAP profile using JSON/XML as the data interchange format, and communicate using HTTP over IP. We describe this layer in more detail in Chapter 5.

Sitting on top of this uniform data representation layer are services and applications that are portable. **Services** are REST-ful web services that communicate with data sources in the sensing layer via sMAP and provide added functionalities to applications; and **applications** can directly communicate with data sources via sMAP and with multiple services. For example, database is a commonly used service that stores historical data from data source below and provide them to multiple applications above. Different applications can utilize these data layer and service layer in innovative ways. Building managers may wish to use an actuation and control application for maintaining the building; occupants may use a personal feedback and energy visualization software running on their mobile phones for tracking their personal energy usage; and application developers can utilize these data in a debugging

environment before publishing their building application.

In the next three chapters, we define the specifics for each of the three layers in this architecture, and demonstrate its feasibility by providing an empirical basis for evaluating its effectiveness. Based on this architecture, we implement and deploy a real building monitoring system and develop an energy feedback application, as shown on the right side of Figure 3.3. This application is an instantiation of our architecture for evaluating its effectiveness to enable building applications. In this application, we present the process flow from understanding and measurement of the load tree to the data analysis, modeling, and disaggregation of energy usage (Chapter 7), and finally re-aggregating into actionable representations for visualization and feedback to the occupants (Chapter 8).

Chapter 4

Sensing and Measurement

“If you cannot measure it, you cannot improve it.” - Lord Kelvin

Indeed, without empirical measurement of the physical world, it is impossible to know where energy is distributed, consumed, and wasted, let alone reducing it.

The premise of our architecture is the existence of a pervasive networked monitoring infrastructure that connects humans and applications with the energy pulse of the building in real-time. This sensing tier is defined as the collection of devices of data sources that provide physical information and has the ability to communicate with external devices, either directly or through proxies. This sensing is composed of a wide range of data sources, forming what we call the *observation space*, which consists of three dimensions - *what*, *where*, and *when*.

“What” refers to the type of physical phenomenon been measured, such as electricity, water flow, and light intensity. In addition to directly measuring electrical power, several research have shown that motion, sound, location, and vibration can all be used to infer electrical energy [28]. “Where” refers to the location of the observation point, in particular, with respect to the load tree (explained in more detail in Section 7.1) of the building. A coarse point of observation is at the root of the tree – the whole building consumption. However, at this granularity, while we have the entire envelope, it lacks detailed information of how energy is used within this envelope. By further instrumenting deep into the load tree, we arrive at locations such as electrical distribution panels for floors, power distribution units (PDUs) for server rooms, and eventually single plug-loads or even individual resistors or capacitors inside an appliance. “When” refers to the frequency of measurements and relative phases between them. For example, the home PG&E bill is one sample point per month, with a resolution of kWh. Whole building usage is often at per-minute granularity, and reported in 15 minute batches. In contrast, modern meters and scopes can often measure at Mhz frequency. In our system, a combination of different sensors and meters are used, each at a different point in the observation space, motivating the need for an architecture for obtaining time-correlated data in real-time.

While many sensor feeds already exist in modern buildings at different granularities, an

important type of meter is often missing from current buildings – plug-load meters that directly monitor electrical usages of appliances at the leaves of the load tree. Existing plug-load meters are usually unconnected and do not scale, as discussed in Section 2.2.2. To address this problem, we designed a wireless IPv6-based plug-load meter – called *ACme*. We describe the design, implementation, and evaluation of *ACme* in Section 4.1.

As mentioned before, in some situations, it is inconvenient or costly to directly measure electrical power. Instead, we can infer energy consumption by using sensors with different *modes*, such as vibration and light. We describe some of these sensors and methods for obtaining time-correlated data in Section 4.2.

4.1 Plug-load Energy Monitoring and Control

ACme is a wireless sensor and actuator device for monitoring AC energy usage and controlling AC devices in a building environment. The *ACme* node integrates an Epic [38] core module with a dedicated energy metering IC to provide real, reactive, and apparent power measurements, with optional control of an attached load. *ACme* uses the Berkeley *blip* [32] IPv6/6LoWPAN stack to provide IPv6 connectivity. *ACme* nodes automatically join the IPv6 subnet after being plugged in, and begin interactions with applications. We evaluate our system in a green building deployment with 49 nodes spread over several floors of a Computer Science Building and present energy consumption data from this deployment.

4.1.1 Problem Overview

Electricity consumption in typical commercial buildings is divided between HVAC, lighting, and plug-loads. The Soda Hall building at Berkeley consumes approximately 12MWh of electricity per day. Preliminary data show that the plug-load usage represents approximately 20%, or 2.4MWh, of the total usage and much of this is used by desktop systems. In addition, these loads generate heat that must be removed by the HVAC system. Desktop usage is also highly correlated with lighting loads. My goal is to build an interactive, near real-time monitoring capability of numerous small loads, as well as the fewer large ones, so that occupants can understand their electricity usage patterns and adjust their behavior to reduce their energy footprint.

The AC metering application drives a different set of technical challenges and pushes sensor network research in some new directions. Although ultra low power operation on batteries is not required, since the embedded devices are mains-powered, they must consume little enough energy that they do not adversely affect the standby monitoring load. Moreover, since they are monitoring the very flow of electricity that they utilize, so the design of the AC-DC power supply and the power metering is inter-related and rather subtle. The design of the embedded device must address a family of thermal, noise isolation, and safety concerns that few sensor networks have examined. We use this challenge to exercise the recently

published Epic [38] design methodology of expert modules and application-specific “glue” to support the progression from prototype to pilot to production. While the network need not operate with a low radio duty cycle, it needs to be robust even though the devices are deployed in highly RF challenged settings – behind refrigerators, under metal desks, in metal cabinets, on the microwave, and so on.

Moreover, the collection of devices is not a distributed instrument deployed by a single authority, i.e., a macroscope; rather, it is a network deployed bottom up by individuals for a variety of reasons in a variety of settings. The application and usage can be quite heterogeneous, although all the devices cooperate at the network layer to route traffic for one another.

To evaluate our design, we apply the technology to the problem of excessive energy usage in our Computer Science Building. We made ACme nodes available to a relatively large number of students, faculty, and staff, and encouraged them to monitor their workstations, laptops, and other electronics. Collectively the data collection occurs over our ad-hoc mesh network. We have also instrumented other devices with significant power draw in common areas, and made complete power traces from these devices publicly available via the database and web interface. Please refer to Section 9 for more details.

4.1.2 ACme Hardware Design

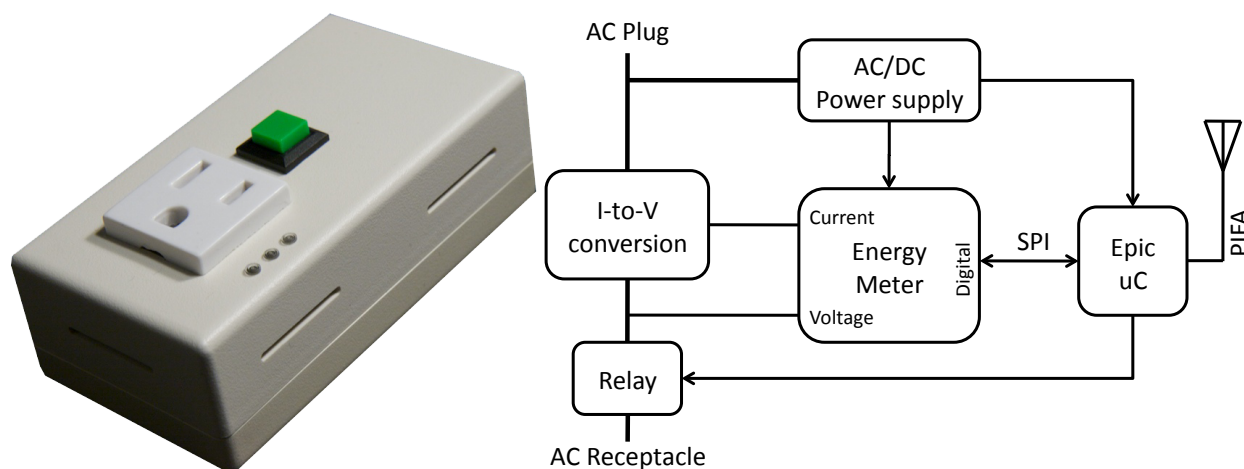


Figure 4.1: ACme consists of five primary components: current-to-voltage conversion, energy metering, AC/DC power supply, microcontroller and radio, and solid state relay.

The power measurement and control is performed by a device consisting of five primary components: current-to-voltage conversion, energy metering, AC/DC power supply, microcontroller with radio, and solid state AC relay, as shown in Figure 4.1.

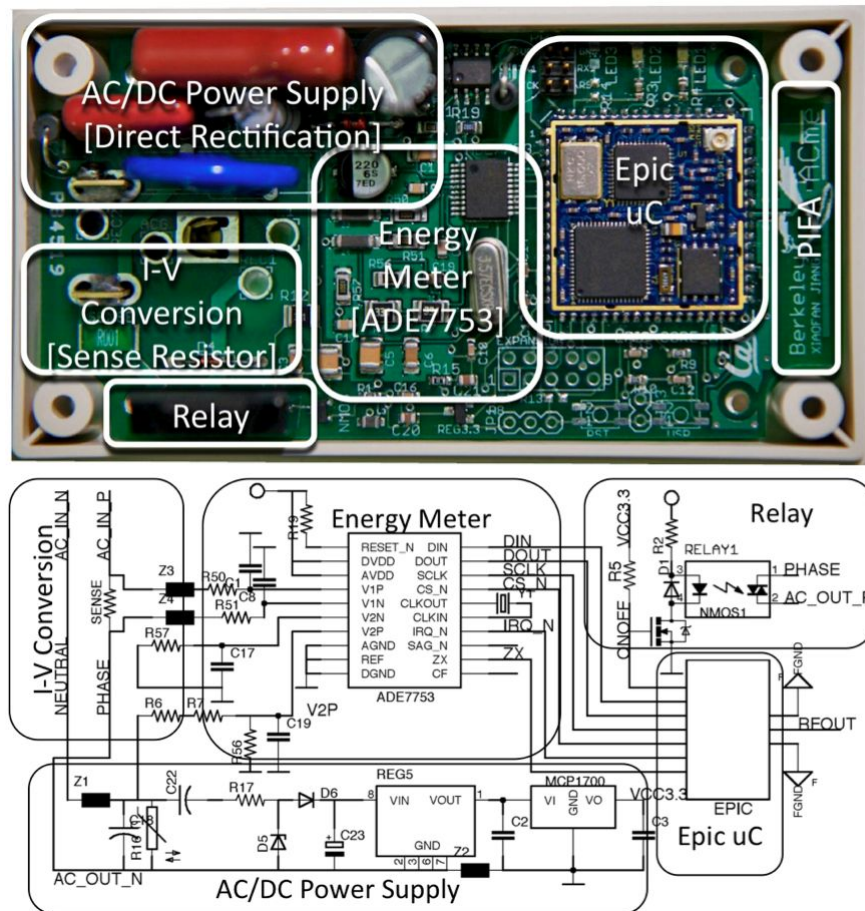


Figure 4.2: ACme-A uses shunt resistor as I-V conversion, direct rectification as AC/DC power supply, and ADE7753 as the energy metering IC.

The current-to-voltage converter converts the current used by the AC appliance to voltage, as described in Section 4.1.2. This signal is amplified and filtered before multiplying with the AC voltage to obtain power. To obtain real, reactive, and apparent power measurements, a dedicated IC is usually used to perform the necessary analog-digital conversions and calculations. The result is output over a digital bus to the Epic microcontroller. *ACme* also takes advantage of the available AC to power its own operation; this requires an AC/DC power supply. To give *ACme* the ability to control the AC device, a solid state relay is used. Finally, a microcontroller is used to communicate with the energy metering chip via SPI, and to the network via radio.

There are several design choices for each component, each with its own advantages and disadvantages. Design decisions for the different components of *ACme* are not independent. For example, a particular choice of current-to-voltage conversion requires a compatible choice

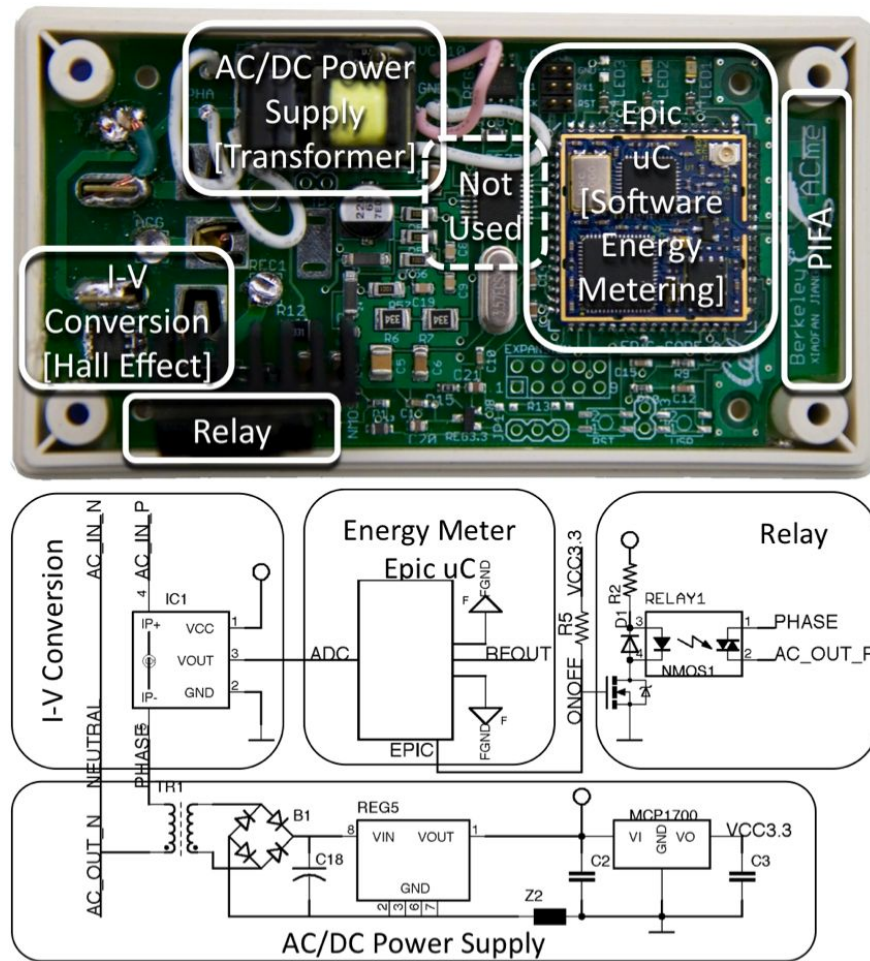


Figure 4.3: ACme-B uses an in-line Hall Effect sensor as I-V conversion, a step-down transformer followed by a bridge rectifier as the AC/DC power supply, and performs energy calculation in software using the microcontroller.

of power supply and energy meter. We find that there are only a few internally consistent combinations, each appropriate for a distinct class of applications. To explore the available design space, we built two meters, shown in Figure 4.2 and 4.3. We compare the two designs and evaluate their effectiveness for our application in Section 4.1.2.

Current-to-Voltage Conversion

There are multiple ways to convert current to voltage. The design choices we make in I-V conversion constrain our choice of power supply and energy meter. We evaluate their differences and tradeoffs with respect to their corresponding power supply and energy meter.

A **shunt resistor** is the most common choice for inexpensive AC meters used in 2-

wire single phase systems. It is used in the ACme-A design as shown in Figure 4.2. A precision resistor in the range of $m\Omega$ is typically used in series with the AC load. The small voltage drop across the resistor is proportional to the current, and is first amplified, and then multiplied by the voltage, to produce power measurements.

Shunt resistors are inexpensive and small in size, but they have drawbacks. The voltage drop across the shunt may effect the AC device if the load has a high power draw. Also, the resistive heating results in thermal design concerns as well as measurement issues, since the resistance of the shunt is proportional to its temperature.

The power supply typically must establish a virtual ground at one end of the shunt resistor because most energy meter ICs are not fully differential. This implies that direct rectification should be used in the power supply. The safety of this design is less than ideal because a virtual ground is established at AC mains, coupling the rest of the metering circuit to high voltage. This method is also inefficient because it requires “dumping” energy to ground.

Another common method of converting current to voltage is to use a **current transformer**. There are two common versions of current transformers: a traditional in-line version, and non-contacting clamp-on version.

In-line current transformers are a popular choice for moderately priced AC meters used in 3-wire distribution systems. The current transformer performs vector addition of Phase A and Phase B via two primary turns of opposite polarity [1] and transfers the current to the secondary via magnetic induction. The current on the secondary is converted to voltage via a “burden” resistor.

Current transformers cleanly decouple the low voltage DC circuit from the high voltage AC input, which make them safer than a shunt resistor. Using a current transformer in the I-V conversion stage allows us to use a step-down transformer followed by low-voltage rectification as the power supply, which is more efficient than direct AC rectification. However, current transformers are more expensive than shunt resistors and also require phase compensation. They are also generally large in size and weight; the smallest current transformer readily available is relatively large compared to the rest of the ACme board. For this reason, we decided against using current transformer as the I-V conversion element in ACme.

The second form of current transformers are clamp-on meters, in which case the entire sensor is the secondary of the inductive coupling. These sensors have the obvious benefit of non-intrusive measurement. However, they also have two flaws: (1) the wires in the AC line need to be physically separated so that the sensor can attach to the phase wire, and (2) the meter itself must be powered by a separate power supply, which means either batteries or a separate AC-DC converter is needed. The clamp-on option is attractive for branch level metering, but not ideal for receptacle level.

A third method of converting current to voltage is the **Hall Effect sensor**. These devices use the Hall Effect [10] to measure current and can be either clamp-on (non-contacting) or in-line. The clamp-on form factor is not considered here for reasons above.

In-line Hall Effect sensors intercept the AC current and couple it with an internally

calibrated Hall Effect element. This approach is compact and precise. More importantly, the high voltage AC input is electrically isolated from the low voltage output inside the in-line Hall Effect sensor, providing an electric isolation of kilovolts. This makes it possible to use an efficient step-down transformer as the power supply. The step-down transformer also establishes a ground at a safe, low voltage. ACme-B is designed using this approach, and is shown in Figure 4.3.

Energy Metering

Energy metering is the process of calculating the power and energy from the current and voltage. This process can be done in either software or hardware. The two methods have different tradeoffs and are appropriate for different applications.

In the software solution, a single wire connects the output from the I-V conversion to the microcontroller's ADC. This pushes measurement into the microcontroller, which must sample the signals, multiply, and accumulate in software. While this choice avoids the need for a dedicated meter IC, the microcontroller is kept busy performing the sampling, and the results are less precise due to lower sampling rates. In the case of ACme-B, we cannot connect the AC wires to the microcontroller directly to obtain the voltage because a transformer is used as the power supply. We assume a constant RMS voltage in converting current to power. This is acceptable for applications which monitor only apparent power.

There are many commercial ICs that perform energy measurement in hardware. For example, Microchip's MCP3905 supports real power measurement using two ADC channels, one for current and one for voltage. The output is a pulse whose frequency is proportional to the power. However, it does not support energy accumulation, requiring another chip or the microprocessor for computing energy. The analog pulse output requires either constant sampling using ADC or triggering an interrupt on every pulse, further burdening the microcontroller. Analog Devices's ADE7753 provides real, reactive, and apparent power calculations. It internally integrates power to produce energy, provides extensive filtering, and includes a temperature sensor. ADE7753 stores power and energy measurements in registers, and communicates with the microcontroller via the SPI bus. This simplifies the data acquisition process and allows the microcontroller to easily configure parameters in the ADE7753. ACme-A (Figure 4.2) uses this chip.

AC/DC Power Supply

The two typical approaches to DC power supply design are direct rectification or step-down transformer followed by rectification. They are each suitable for a particular type of I-V conversion; the choice is determined by how the ground reference is established, in addition to efficiency, size, and cost considerations. We built both designs to better understand the relationship and tradeoffs between components. ACme-A uses direct rectification while ACme-B uses a transformer.

There are several methods for rectifying an AC signal. We chose a simple half wave rectifier. Direct rectification of the high main voltage avoids the use of a transformer but requires high voltage capacitors. Care needs to be taken when using this method because the ground is at high voltage. This will lead to shorts or even fire if another device that uses earth ground is connected while powered on, as might occur when using a scope to debug the circuit or connecting to a PC.

Traditional step-down transformers followed by a bridge rectifier are more efficient than direct rectification since inductive coupling wastes little power and a bridge rectifier at low voltage is efficient. By using a current transformer or an in-line Hall effect sensor for the I-V conversion, the low-voltage part of the circuit is safely isolated from the high-voltage input, sometimes referred to as “Galvanic isolation”. Unfortunately, typical current transformers are too large for our application. Instead, we use a Hall-effect ASIC sensor in the design of ACme-B.

Comparison of ACme-A and ACme-B

	ACme-A	ACme-B
Measurements	real, reactive, apparent	apparent
Energy accum.	hardware	software
CPU load	low	high
Cost	medium	low
Idle power	1W	0.1W

Table 4.1: Properties of ACme-A and ACme-B.

ACme-A, shown in Figure 4.2 uses the combination of a $1m\Omega$ precision sensing resistor, direct AC-to-DC rectification, and the ADE7753 energy meter IC. ACme-B, shown in Figure 4.3 uses the combination of in-line Hall Effect sensor, transformer-based power supply, and software-based energy metering. Table 4.1 shows the basic differences between them. As we can see, ACme-A is a more capable energy meter at the expense of efficiency. ACme-B, on the other hand, reduces cost and complexity at the expense of lower power measurement fidelity and higher CPU load. We believe that the full range of high-fidelity energy measurements is important in order to support various types of energy related applications that this platform is intended to enable. The 1W idle power is at the high end of acceptable overhead. For the rest of the paper, we focus on ACme-A, but the observations carry over to ACme-B.

Control

The appropriate form of control for many AC devices is a simple microcontroller controlled switch. This allows the embedded software, and consequently remote application, to have full control over the state of the AC device. It is useful for many applications in which we want to close the sensing-actuation loop. We chose a solid-state-relay over traditional

electro-mechanical relays due to its smaller size, higher switching speed, higher EMI immunity, and a near-infinite switching lifetime. Our choice, the Sharp S216SE1, provides a high isolation voltage due to opto-coupling. The S216SE1 has a rated maximum current of 15A (1800W).

First, while the solid state relay gives ACme the ability to control appliances, it produces considerable heat, limiting the maximum current rating of ACme. Also, ACme should maintain the state of the relay during reboot and reprogramming. Ideally this would be implemented in hardware using a bistable switch; this version of ACme simply ensures that the plug is turned on during reboots, maintaining the same semantic as a pass-through plug. We learned this lesson the hard way, accidentally power-cycling all devices under measurement after reprogramming the network over the air!

Microcontroller and Radio

ACme uses the Epic Core [38] as its microcontroller and radio because Epic exports all the functionalities of a typical “mote” but in a small and easily-composable package. Epic is a single-side-leadless module that can be easily incorporated into a circuit board. In future iterations of ACme production, Epic Core can be inlined into the ACme layout to further reduce cost and assembly time. Epic Core uses the TI MSP430F16 microcontroller and the CC2420 radio, which are both supported by TinyOS.

To enable high speed sampling of energy, we connect the SPI bus of the ADE7753 to USART1 of MSP430, since USART0 is already shared by the radio and the flash. This configuration allows high speed samples to stream into flash. ADE7753’s IRQ and zero-crossing outputs are connected to provide wake-up interrupt and dimmer functionality. Finally, a PIFA board antenna provides good radio range and minimizes cost.

Mechanical

Usability is a primary concern for ACme. We use a plug-through design which minimizes intrusiveness by acting as a small in-line power adaptor. The male plug of the appliance is inserted into the female receptacle of ACme, and the male side of ACme is plugged into the AC outlet. A small form factor is important because users will not accept large, research-grade devices. We chose a UL approved ABS plastic enclosure the size of a regular power adaptor with a dimension of 10x5.6x4cm.

One of the applications of energy monitoring is to alert users of abnormalities or excessive energy use. To natively support these types of applications and for conventional system diagnostics, we expose the red, green, and blue LEDs to the outside via three light-pipes. Users are also able to control various aspects of ACme via the external push-button.

Thermal

Because the relay and AC/DC power supply produce considerable heat under heavy load, we machined the enclosures with vents on all sides in addition to putting a heat sink on the relay. However, real users often place ACme's under desks or behind appliances where ventilation is minimal. Due to the lack of air convection, even with a heat sink, the solid-state relay still produces significant heat build-up under prolonged high wattage loads such as microwaves and heaters. As a result, in a later iteration of ACme's, we have removed the solid-state relay in favor of an electro-mechanical relay.

In addition to heat generated by relays and power supplies, at 15 amps of current, the current-carrying PCB trace itself produces significant heat, and is proportional to the trace resistance, as characterized by its geometry. In designing the latest revision of ACme (ACmeX2), we use the following equations to calculate the minimum trace width for external copper layer at low frequencies:

$$\begin{aligned} Area[mils^2] &= (Current[Amps]/(k * (Temp_{Rise}[^{\circ}C])^b))^{1/c} \\ Width[mils] &= Area[mils^2]/(Thickness[oz] * 1.378[mils/oz]) \\ &\text{where } k = 0.048, b = 0.44, c = 0.725 \end{aligned}$$

At 15A current, 2oz copper weight, and 40°C rise from 25°C ambient temperature, the minimum trace width is calculated to be 107mil. Using this as a guideline, we designed the current path by constraining the minimum trace width to 120mil, shortened the routing distance, and used large polygon fills on both sides of the board as heat sinks, as shown in Figure 4.4. This design significantly reduces heat dissipated and require no heat-sink nor enclosure vents. We placed ACmeX2 under a load at 1800W and measured the internal temperature to reach a maximum of 70°C while outside remains close to room temperature.

ACmeX2

We deployed around 100 ACme-A in various environments for about a year and generated a large amount of usage data. In addition, we received valuable feedbacks from real users from different scenarios. As a result, we redesigned ACme combining the advantages of both ACme-A and ACme-B to result in the following improvements:

- Reduced size significantly to fit inside floor boxes.
- Removed relay to address privacy concerns and to reduce size.
- Updated power supply module to reduce cost and size.
- Resigned current path to reduce heat dissipation.
- Use Hall-effect sensor for non-intrusive sensing similar to ACme-B

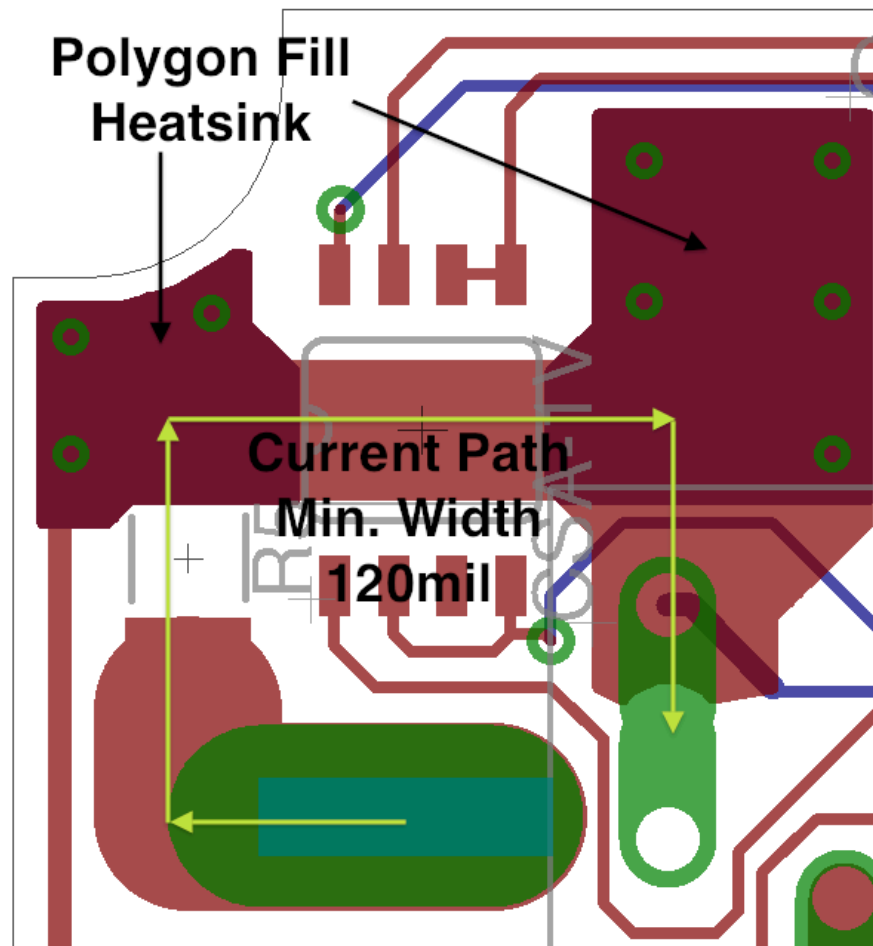


Figure 4.4: Current carrying trace has a minimum width of 120mil and uses polygon fills as heat sinks.

- Use ADE7753 for accurate energy measurement similar to ACme-A.

We have named it ACmeX2 and are in the process of manufacturing around 1000 units to be used in the Lawrence Berkeley National Laboratory (LBNL) and Cory Hall deployments.

4.1.3 Embedded Software

The ACme node software provides a simple interface over the network, shown in Table 4.2, and is built around an ASCII shell component running on UDP port 2000. The shell allows us to remotely adjust sample parameters, debug the network connection, and access an over-the-air programming facility on individual meters. Data is reported to an external host using a separate UDP port in a binary data format.

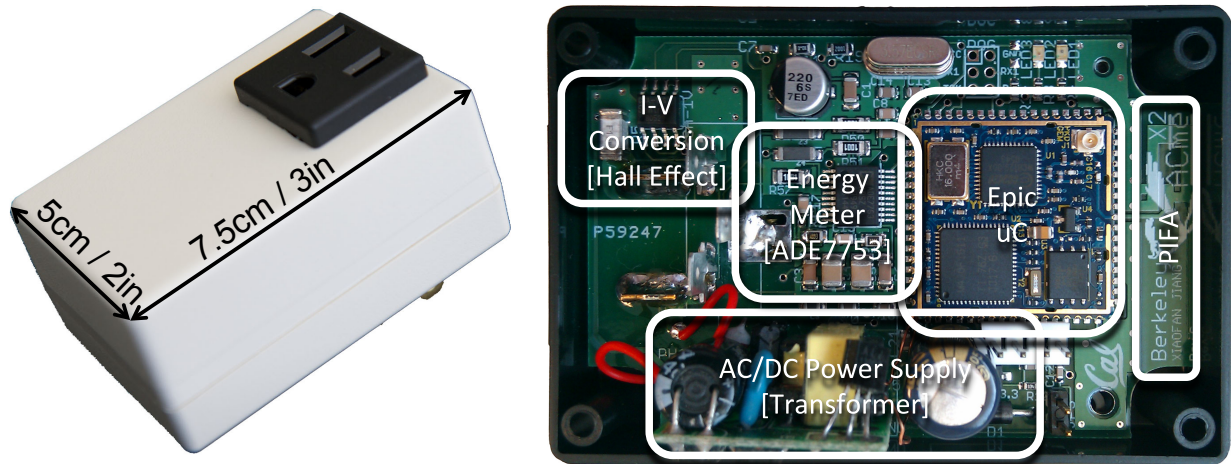


Figure 4.5: ACmeX2 is more compact and efficient but does not include relay.

ACme command	function
<code>read_energy()</code>	read the current energy
<code>read_power()</code>	read current power
<code>report(ip_addr, rate)</code>	send reports
<code>switch(state)</code>	set relay to state

Table 4.2: ACme node interface

The driver for the metering chip and relay is implemented in TinyOS as a standard sampling interface which provides the ability to read the energy consumed by the appliance at regular intervals via a `start(period)` command. The driver can turn on and off the connected AC device using the `set(state)` command and query the current state via a `getState()` command.

In the ACme-B implementation, we sample the AC waveform from the Hall Effect sensor at 6kHz, and track the peaks. The peaks are proportional to power, assuming constant voltage. Power is further accumulated inside the microcontroller to obtain energy.

4.1.4 Evaluation

We evaluate the design of ACme based on how well it enables energy related applications and on its ease of use in home and office environments. As shown in Table 4.3, ACme-A can be used in challenging applications where high-fidelity data is needed for all three types of power. ACme can also provide high speed sampling or long term metering. ACmeX2 further improves upon ACme-A and ACme-B to provide better performance, lower idle power, and smaller form factor, as shown in Table 4.4.

Measurements	W, VA, VAR
Resolution	40mW
Sampling speed	14kHz
Report speed	2.8kHz
Maximum power	1200W
Energy accumulation	6.26min
Radio range	multiple floors
Idle power	1W
Size	10x5.6x4cm

Table 4.3: Summary of ACme-A

Measurements	W, VA, VAR
Resolution	25mW
Sampling speed	14kHz
Report speed	2.8kHz
Maximum power	1800W
Energy accumulation	6.26min
Radio range	multiple floors
Idle power	< 0.5W
Size	7.5x5x4cm

Table 4.4: Summary of ACmeX2

ACme can be easily adopted into home and offices thanks to its plug-through design and small size, comparable to a laptop power adaptor. We evaluate ACme in a building-wide deployment, as described in Section 9.

4.1.5 Micro Benchmarking

To evaluate how ACme performs on real loads, we measure several appliances found in a typical office environment. We find that ACme is able to provide visibility into appliance power consumptions with adequate resolution, dynamic range, and over long periods of time at high temporal fidelity.

In Figure 4.6, an ACme-A is used to monitor a Polycomm conference telephone, which consumes around 2 watts on standby and around 3 watt when active. The steps in the vertical dimension indicates that the maximum resolution is around 40mW.

In Figure 4.7, an ACme is used to monitor a coffee maker, which utilizes a heating element. From this graph, we can see that ACme accurately captures the power spike exhibited by the heating cycles of the coffee maker.

In Figure 4.8, we can observe several interesting modes of a laptop. The exponentially delaying curves at the beginning of each cycle shows the charging characteristic of this laptop right after it plugs into a power adapter – as batteries is being charged, the rate of charging also slows down. The spiky portions shows the power consumption of active elements of the laptop, including CPU, screen, fans, and etc.

In Figure 4.9, an ACme is used to monitor a refrigerator. This graph shows two interesting modes of this refrigerator – the smaller (around 100W) but frequent (15 minutes interval) cycles represent the power consumptions of compressor activity, while the less frequent (a few times per day) but higher power (450W) cycles are caused by the defrost cycles.



Figure 4.6: Power trace of a Polycomm telephone shows that ACme-A is able to resolve low power appliances down to 40mW.

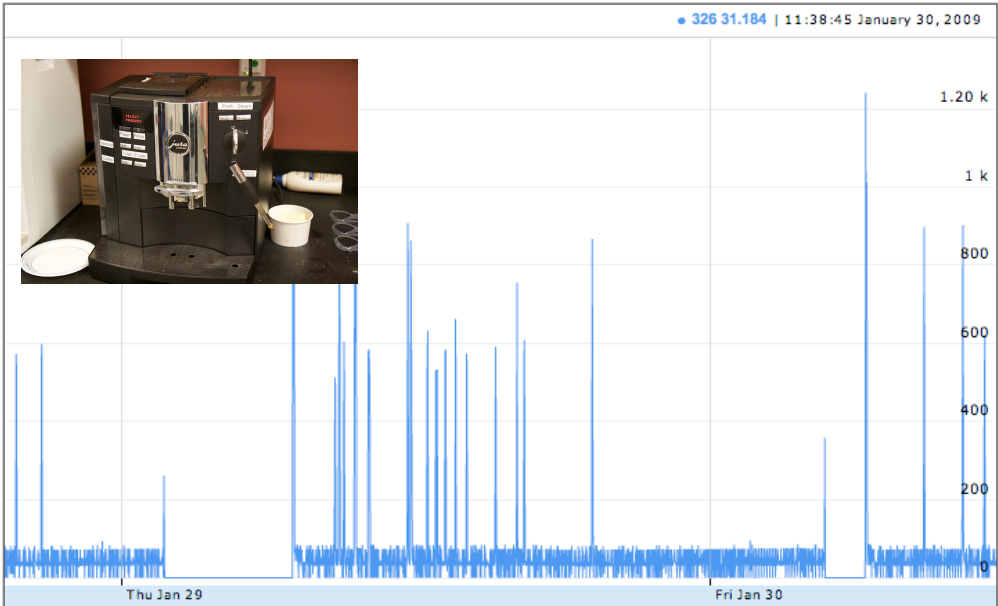


Figure 4.7: Power trace of a coffee maker shows that ACme-A is able to measure appliances with high power pulses.



Figure 4.8: ACme reveals interesting characteristics of devices such as a laptop. We can observe the power states, charging curve, and CPU usage.

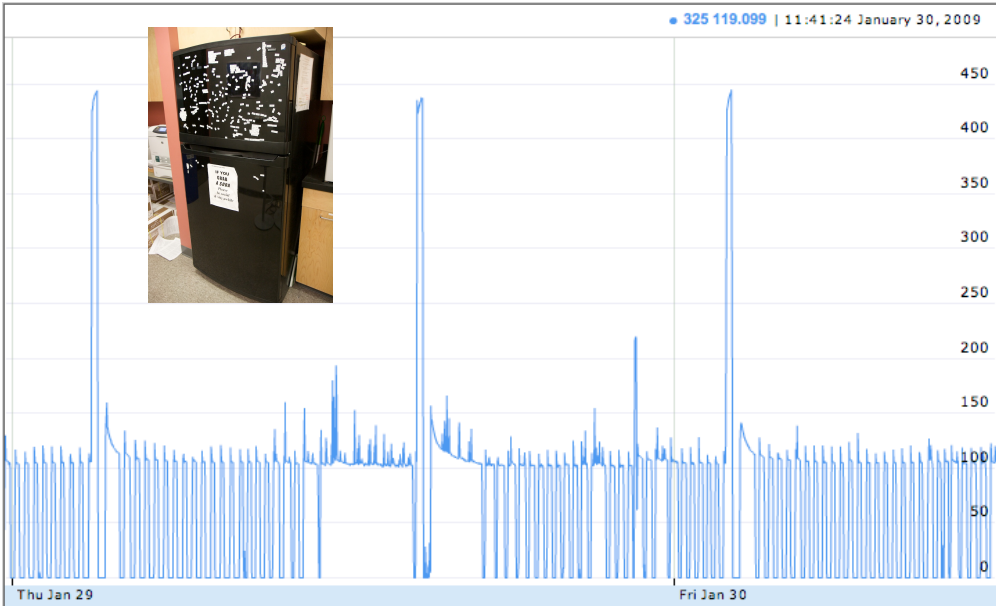


Figure 4.9: ACme reveals both the compressor cycles (100W at around 15 minutes intervals) and defrost cycles (450W a few times a day) in a refrigerator.

4.1.6 Privacy Implications

Privacy issues abound whenever a substantial fraction of devices around people are instrumented and report data to an “eye in the sky.” Even in our own limited deployment, we found it easy to construct privacy-invading queries which were able to reliably determine the lab-occupancy habits of different students in our lab. The power consumed by their work stations and nearby lights are a good predictor whether or not they are there. The students were understandably leery of this once they realized that their advisor might be very interested in that query!

While not addressed in this work, we believe that the use of the internet architecture will actually be a boon when considering the privacy implications of this technology. The fact that the application can be hosted anywhere on the internet means that consumers could choose a variety of providers to host their power algorithms. For instance, their utility might host one, with others hosted by non-profit groups, or locally hosted on computers in the home. This requires robust authentication mechanisms to control the use of the ACme sensors. In general, security in internet-connected sensor networks has not been well addressed. While heavy-handed measures like firewalls, virtual LANs, and proxies can be used, it is important to guard against resource depletion attacks since the resources in question are so meager.

4.1.7 Enabled Research

With the increasing need for energy awareness, we are considering how ACme can be used in future research goals. Many of these goals include human factors and sociological elements which we do not address but can be enabled by ACme.

Personal Accountability: The application we have presented takes an initial step towards the goal of making individuals responsible for the energy they consume. However, we envision much richer data analytics, social features, and control to encourage power reduction. It is well established that making people aware of their consumption causes them to reduce it [37]. In this scenario, we would install ACme devices on a semi-permanent basis and attempt to reduce energy consumption by fostering a culture of conservation.

Energy Auditing: Our hardware, software, and network are designed to enable deployments with little or no planning or preparation. This makes the system ideal for fast energy audits where the system is rapidly deployed across an office building and used to collect data for a short period— from a few days to a week. The data would make occupants more aware of their power footprint and identify the largest opportunities for savings; the process would potentially be conducted iteratively every few months to set goals and assess progress.

Smart Home: Another way to save power in homes and offices is to infer occupancy from various sensors: both the ACme meters we have developed as well as light, humidity, or door-position sensors. An application could use this information to adjust lighting, HVAC, entertainment systems, and any other devices amenable to remote control. Alternatively, residents might carry “smart badges” which would identify them to the home and allow

localization of their positions. The home would respond by developing power profiles to save energy without compromising the home's comfort.

Demand Response: Utilities have a strong incentive to reduce peak demand since that would allow them to reduce dirty “peaking” plants used to meet the last few percent of demand. One proposal is dynamic pricing of power: if price of power increases as generation approaches capacity, energy consumers would be incentivised to reduce their consumption. The combination of Internet connectivity and control make the ACme ideally suited to experiment with “peak-shaving” and “load-shifting” algorithms to reduce the load as generation approaches capacity. For instance, consumers could set up policies to reduce energy consumption when they receive a signal that generation is about to be overloaded, or limit power use to a certain budget. ACme nodes could communicate with each other and make distributed decisions, like an Intelligent Power Switch [45].

4.2 Multi-modal Sensing

One challenge often encountered when trying to instrument a building is that certain consumers of energy are either hard to measure or inaccessible. For example, HVAC electrical energy is converted to other forms of energy in a central location, which is secured and inaccessible. Even if one is to obtain permission to enter the premises, special sensors such as CT clamps are needed to monitor consumption. However, there are multiple ways to obtain equivalent energy measurement without directly measuring consumption.

HVAC systems such as chillers and pumps usually contain relatively high horsepower motors. A close inspection of our building-wide HVAC control room reveals that these systems produce very noticeable vibrations. To demonstrate this, an Epic-based node [38] with both a three-axis accelerometer and a vibration sensor was used to measure the duty-cycling schedule of a motor that drives an air conditioning unit. Multiplying this duty cycle by the ON power produces a real-time power profile. Figure 4.10 (A) shows a trace of one axis of the accelerometer over an hour, and Figure 4.10 (B) shows a trace of a vibration sensor over the same hour. For the particular vibration sensor used, the number of threshold crossings at each motor event varies, but the sensor identifies the beginning of motor events well. Either type of sensor produces a proxy measurement for calculating the electrical energy used by a large motor that is otherwise cumbersome to measure.

Additionally, in most cases, sensors appropriate for a particular form of energy can be used to measure it in its natural unit then convert back to electrical energy using the transfer function along the path of the conversion process. For example, temperature sensors could be used to detect heat flow or AC usage, and flow sensors could be used to detect ventilation usage.

Furthermore, unconventional sensors can be used to infer usage or help in improving the accuracy of existing electrical sensors. For example, proximity sensors carried by individuals can be used to determine the human component to the power profile of refrigerators, or they

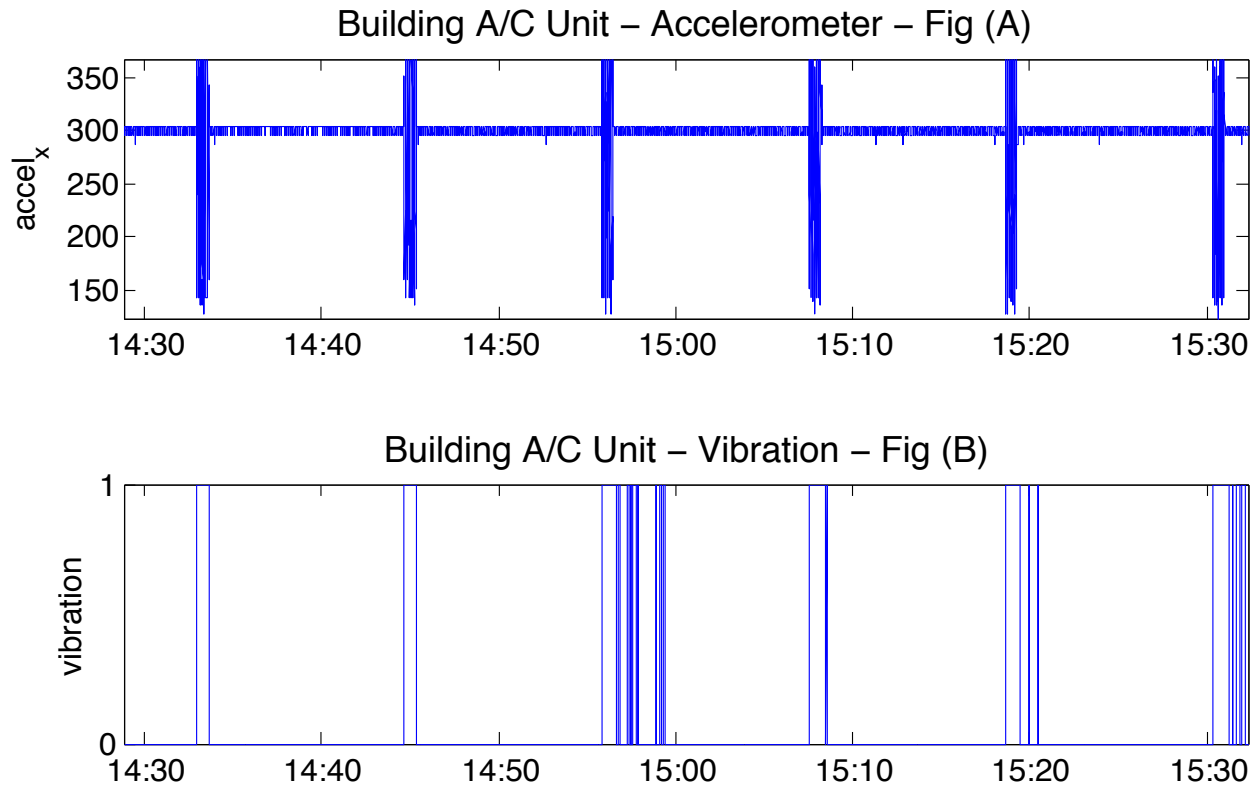


Figure 4.10: Figure (A) shows the output of one axis of an accelerometer attached to a large motor driving air conditioning equipment over an hour. Figure (B) shows the output of a simple vibration sensor attached to the same motor over the same hour. Each sensor indicates when the motor is operating and allows for real-time estimation of HVAC energy consumption.

can be used to allow real-time energy accounting of shared resources such as refrigerators or water heaters, as described in more detail below.

Another shared resource to account for is the energy consumption of overhead lighting. However, to measure it directly, we would need more than ten high-voltage sensors such as CT sensors installed in the electrical panel – this is both cumbersome and costly. Instead, recognizing the fact that all 26 lights fall into 4 light “zones”, controllable only in aggregate, we can simply instrument one light bulb for each of the 4 light zones, as shown in Figure 4.11. We deployed six Telos motes [68] equipped with light sensors and programmed with the same sampling schedule and networking stack as the plug-load meters.

The top graph in Figure 4.12 shows the raw PAR (photosynthetically active radiation) readings while the bottom graph indicates the projected power reading using a simple binary filter with a constant multiplier. As we can see, PAR readings change from near zero to roughly 1085 lumens at around 9AM, indicating that the light has been turned on. The

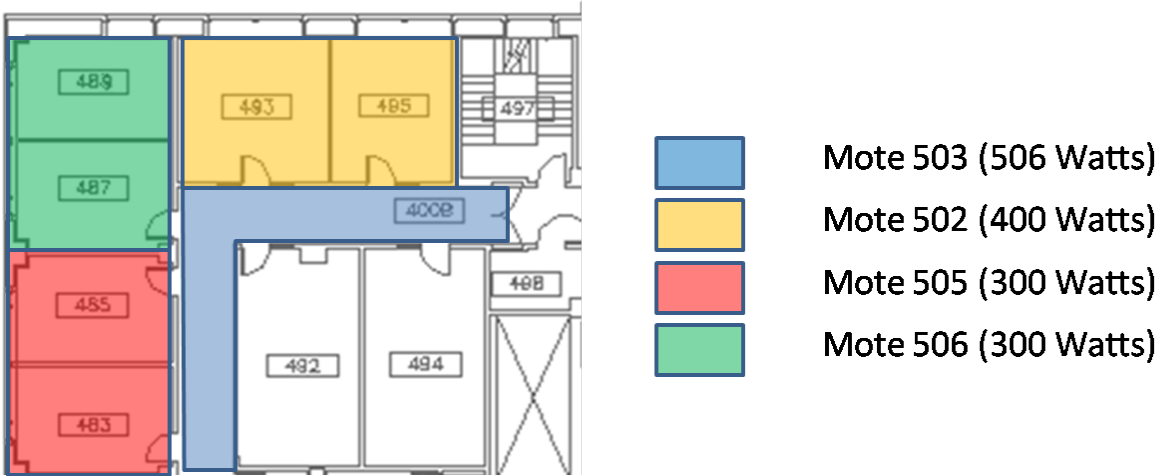


Figure 4.11: Only 4 light sensors are needed to cover 26 light bulbs since they are controlled together by 4 sets of switches and motion sensors.

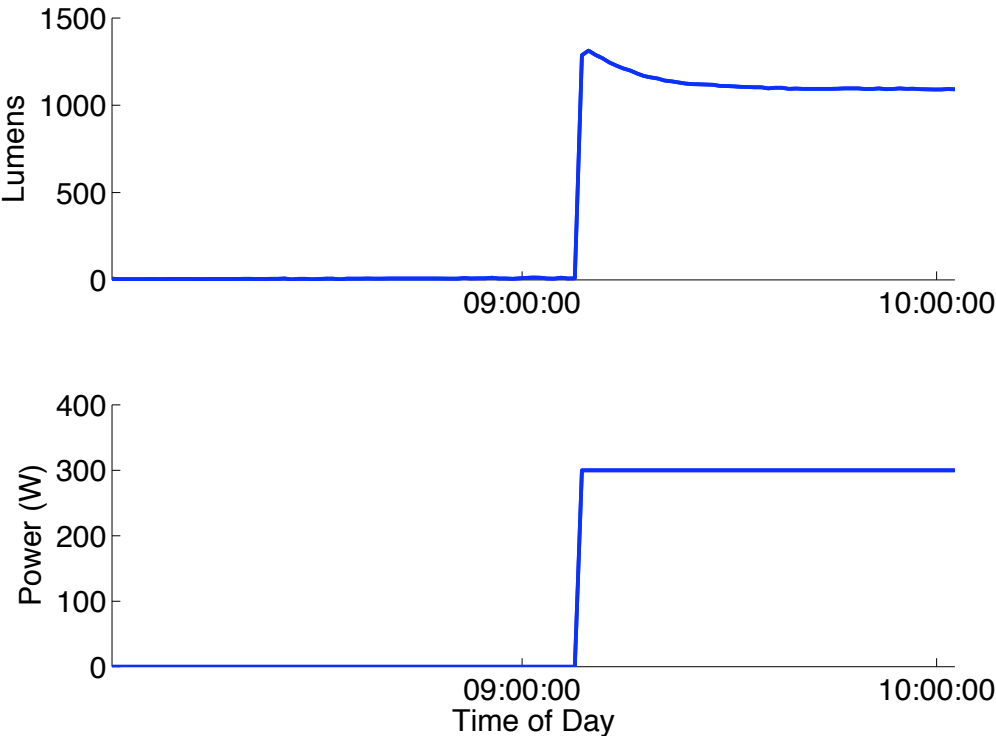


Figure 4.12: Light sensor readings can be easily converted to power using thresholding.

ON/OFF transition is obvious, and can be converted to a 1 or 0 using a threshold. To find the ON power, we simply counted the types of light bulbs and summed their rated power.

4.3 Summary

Combining the advantages of the ACme-A design, which offers higher metering fidelity, with the advantages of ACme-B design, which is more efficient and compact, we arrive at an optimal point in the design space. ACmeX2 is a IPv6-based wireless plug-load meter that provides 1) active, reactive, and apparent power measurements via hardware; 2) high fidelity measurements with milli-watts resolution; 3) the ability to measure appliances up to 1800W without significant heat-up; 4) lower idle power consumption; and 5) compact form-factor.

These characteristics allow ACmeX2 to **accurately** resolve interesting power variations in a wide range of appliances, as demonstrated in Section 4.1.5. Furthermore, because of its compact size and built-in wireless interface, ACmeX2 enables us to **easily instrument** a large number of plug-loads without needing a wired infrastructure. Using an ad-hoc wireless network, ACmes are **scalable** and data is readily **accessible**.

In addition, by using multi-modal sensing, such as using light sensors and vibration sensors, we are able to monitor devices that are normally inaccessible, such as HVAC and high-voltage lighting systems.

In summary, the sensing tier, with the combination of ACme plug-load meters and multi-modal sensors, provide **high-fidelity and accessible** building energy data, both in end-uses and in infrastructure energy usage. In the next section, we integrate with existing building instruments and panel-level meters to provide more complete coverage of the building.

Chapter 5

Data Delivery and Representation

The previous chapter explored various sensing, metering, and actuation options appropriate for energy monitoring and control at different granularities within the building. In particular, we investigate the design and evaluation of a wireless plug-load energy meter - ACme - that provides fine-granularity energy data at the edges of the electrical load tree.

Individually these meters provide isolated data points and is much less interesting than real-time correlated data. A key motivation of our building monitoring architecture is to allow quick deployment of a large number of instrumentation points, such as ACme's, and enable continuous real-time access. At this granularity, requiring connectivity over USB, serial, Ethernet, or other wired channel would not be practical.

In this chapter, we first describe an IPv6-based ad-hoc network that provides wireless connectivity to ACme nodes without the use of either wiring or infrastructure. Our network of ACmes is built on top of the *blip* [32] 6lowpan/IPv6 stack developed by Dawson-Haggerty at Berkeley. This stack provides support for multi-hop IPv6 routing along with TCP and UDP transport protocols, as described in more detail in Section 5.1

While a network of ACme's provides us with detailed electrical usage at the edges of the load tree, it does not give a complete coverage of energy usage inside the building. Fortunately, a commercial building typically contains hundreds if not thousands of sensors and actuators providing temperature, humidity, and electrical measurements at the branch or building scale. But these sense points are often locked inside proprietary protocols and/or sit behind legacy interfaces. To enable simple and efficient communication between these types of data sources and portable applications, including analysis, modeling, and visualization tools, a horizontal abstraction layer is needed. We describe a simple RESTful web-service-based approach that uniformly represents physical data in Section 5.2.

5.1 Networking

Many different approaches to networking a subnet of low-powered devices been presented in the literature. MultiHopLQI or CTP [84, 42] are popular in the sensornets community. However, demand-response control loops will require communication with individual devices and so collection-only protocols are not an option. CentRoute [71] provides communication with single routers in the network, but we are concerned that its static routing decisions would not be sufficiently agile in poor RF environments. Existing open and standardized point-to-point routing protocols for wireless networks such as AODV and DYMO are not appropriate either for this class of networks due to high control overheads or large routing state [67, 30]. While S4 appeared promising, actual use of it would require the design and implementation of a location service [60]. Proprietary stacks such as TSMP, WirelessHART, and PhyNet [20, 23, 11] also appear worthwhile but are not widely available.

Our network is based on `blip` [32], an open-source IPv6 implementation in TinyOS, developed at Berkeley. Key concepts necessary for bringing IPv6 to wireless embedded systems were explored in [47], and so we refer the reader to this reference for a general guide to the challenges and benefits of embracing this network architecture. Importantly, it provides a standard for IPv6 header compression in the form of the `6lowpan` adaptation layer [61], and shows how IPv6 mechanisms like neighbor discovery and extension header processing can be leveraged when using links with extremely small MTUs. In our case, a low-power (duty cycled) link was not necessary since the devices have abundant power available. In this section, we describe design decisions and evaluations specific to our deployment.

Unlike sensor networks with carefully controlled node placements to ensure good connectivity, users may install their ACme meters under their desks, behind metal filing cabinets, or in metal floor boxes, and expect that their data will be visible online. Coverage of these locations by Wi-Fi access points is typically difficult. Instead we utilize the high density of these nodes and multihop routing to obtain coverage even with milliwatt power radios. Their relatively low, packet-per-minute data rates can be serviced by a 802.15.4 mesh.

5.1.1 Routing

The function of the embedded routing protocol is to provide reliable, multi-hop communication within the subnet of ACme nodes. This is a requirement of our design; the sensors and server-hosted applications must be able to communicate. Our routing protocol functions as an intra-subnet routing protocol, responsible for routing packets between endpoints within the same subnet, or delivering them to a gateway if the destination is external. Each sensor node functions as an IP router, and chooses a set of default routes from its neighbors based on router solicitations and router advertisements. Nodes with external connectivity (“edge routers”) advertise a cost of zero, and so the basic routing structure is that of a direct acyclic graph rooted at one or more edge routers.

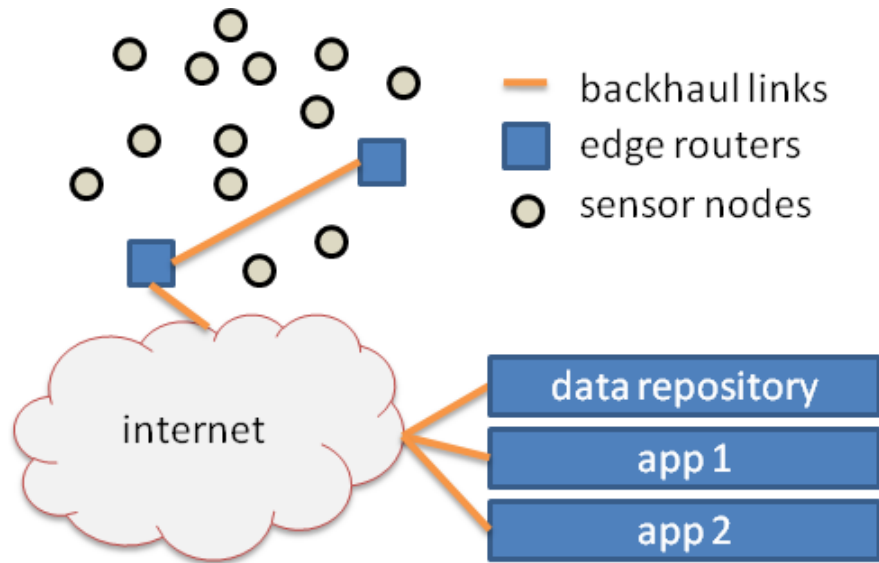


Figure 5.1: Sensor nodes are connected to the Internet through IP routers. ACme nodes are the sensors and edge routers are Linux-class devices.

5.1.2 Edge Router

To provide good network coverage and reliability, we designed and implemented edge routers using two existing Linux-class devices: the Meraki Mini and the OpenMesh Mini-Router [14, 16]. Both of these platforms are built around Atheros system-on-chip products, and run the OpenWRT embedded Linux distribution. Internally, both export a single serial port which we use to add an 802.15.4 radio interface via a user space driver.

The edge routers also act as an Internet router for the subnet assigned to the ACme network. They obtain IPv6 connectivity via a tunnel broker which provides them with a globally routable subnet. Packets from other networks destined to an ACme node arrive at an edge router via the tunnel broker; once they do, they are injected into the network using the same mechanism as internal unicast communication.

5.1.3 Transport

`blip` on the ACme nodes provides TCP and UDP as available transport protocols. Although either would be a reasonable choice for reporting the data, we initially chose UDP since the underlying IP datagram delivery functionality is sufficiently reliable for this application, and removing the end-to-end ACK packets generated by TCP sessions reduces contention. This decision resulted in a very usable system because end-to-end loss is less than 1% for almost all devices and suggests that overhead associated with TCP’s fully acknowledged byte stream may not be necessary. The good reliability results from the local

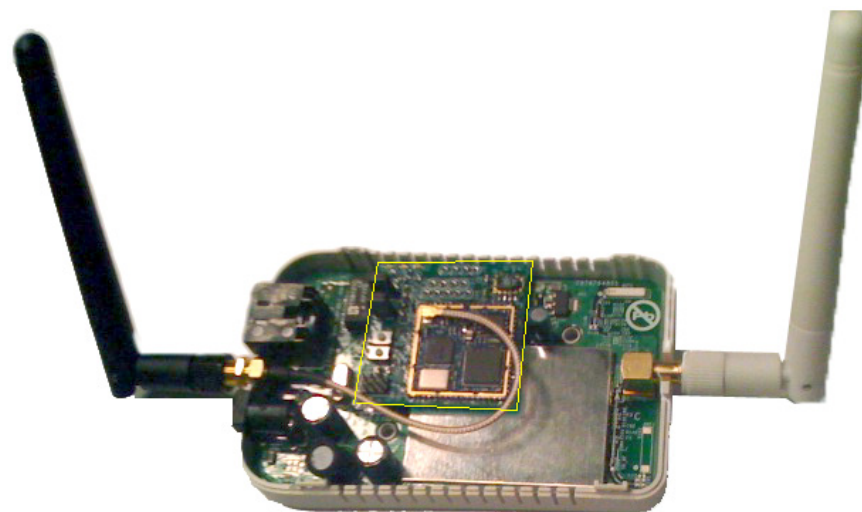


Figure 5.2: This edge router integrates a Meraki Mini and Epic Mote using a custom carrier board and routes IP packets between the sensor network and LAN.

repair methods we employ of multiple next-hops and retransmission. When developing the protocol, we have found that most datagram drops are caused by forwarding queue drops, which are aggravated by poor links since numerous link-layer retries increases queue dwell time. Therefore, the most sensible strategy may be a lower data rate combined with lazy end-to-end NACKs to achieve 100% delivery.

5.1.4 Network Performance

We have deployed a 49-node network across four floors of our Computer Science building. The network uses a single edge router located in our lab to provide routing between the sensors and the LAN, and has been operational for more than a year. Figure 9.3(f) shows a histogram of the data yield from 49 sensors over a three day period. Due to a poor deployment decision, the radios were set to channel 19 resulting in significant interference with existing 802.11 devices and so the network yield was noticeably reduced during busy workdays. For instance, the median yield on Saturday is 99.4%, but drops to 98.4% on Monday. Compared to other reported results concerning the impact of 802.11 on 802.15.4 traffic [62], we observe a relatively modest decline in yield, although our results are not necessarily comparable in different environments.

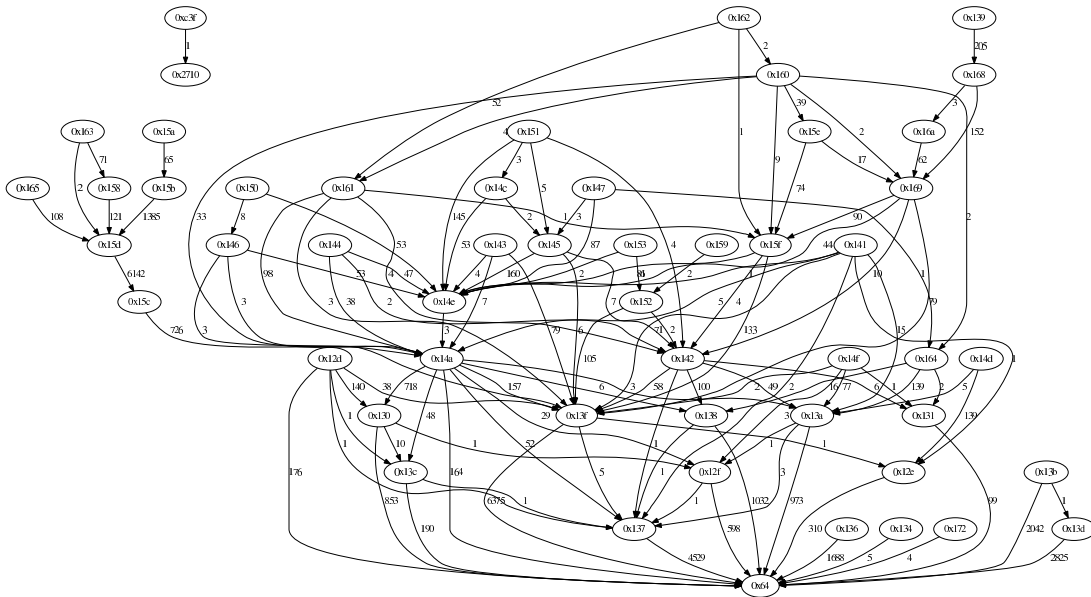


Figure 5.3: A snapshot of the network topology.

5.2 Uniform Representation of Physical Data

In commercial building environments, diverse classes of sensors, meters, and actuators are spread throughout the building, monitoring thousands of sense points at various granularities. In our building scale deployment, a heterogenous set of devices with instinct characteristics are used to measure various energy flows and physical phenomena in the load tree, as shown in Figure 5.4. Several panel-level meters (Dent PowerScout 18 [35]) are deployed to monitor electrical panels for each floor; they speak Modbus protocol over RS485 links. The ACme network is used to measure plug-load power usage, and uses TCP+UDP over IPv6, on top of 802.15.4 wireless links. In addition, environmental sensors such as humidity and light-intensity sensors are used to provide indirect energy measurements (Section 4.2), which use Active Message protocol on top of 802.15.4 links.

To understand how energy is used inside the building, we need time-correlated data in real-time, from all these different types of data sources. However, existing building monitoring solutions usually treat each of them individually as stovepipes, each requiring a different, sometimes proprietary protocol, to communicate. The lack of a unified data representation also hampers the development of portable applications.

To address this problem, we propose a uniform representation of physical information, called *sMAP* - Simple Measuring and Actuation Profile, which provides a horizontal layering, decoupling the myriad of physical sensors, meters, and actuators underneath, from the portable applications above, as shown in Figure 5.5. *sMAP* was motivated by the ACme

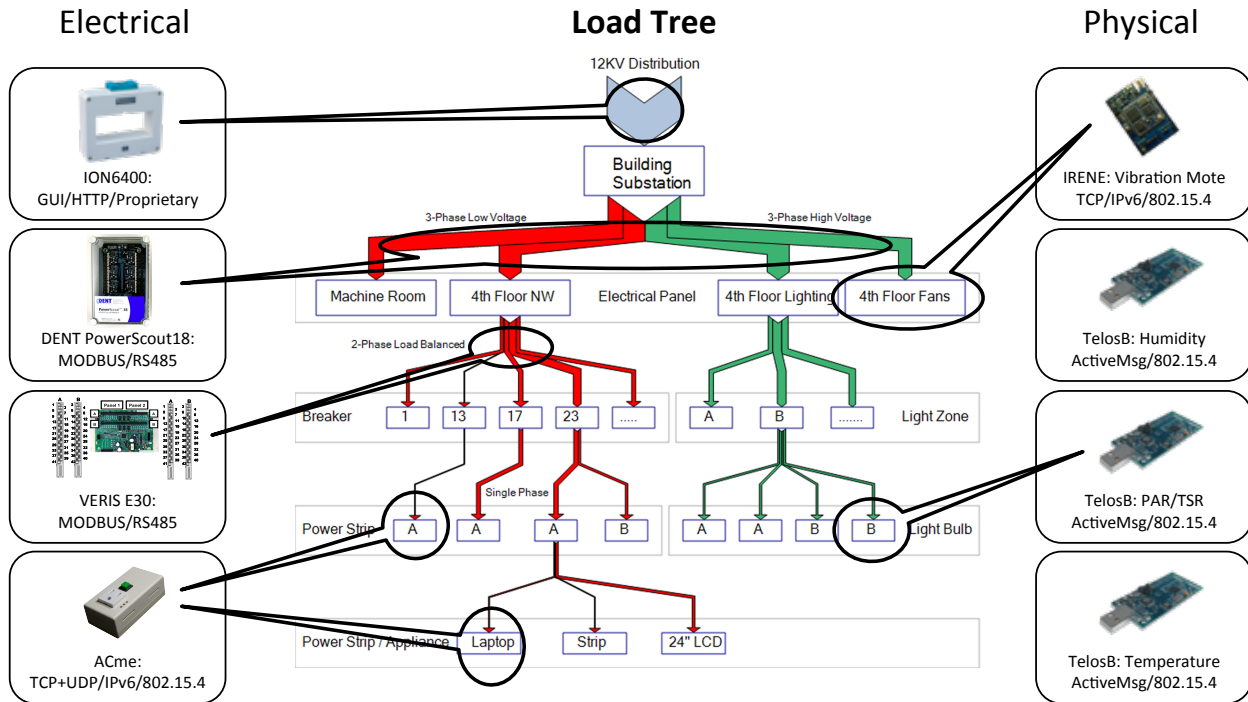


Figure 5.4: Various meters and sensors that measure physical data, but communicate through different protocols and over different physical mediums. For example, the Dent panel meter uses Modbus protocol over RS485 while ACme meters use TCP and UDP over IPv6 on wireless radios, even-though both are communicating the same type of information - quantity of energy.

project and developed in collaboration with Stephen Dawson-Haggerty. In the following section, we describe the high-level design considerations and ACme-specific implementations but refer readers to these papers [50][33] for the complete design, implementation, and evaluation.

5.2.1 Design Considerations

sMAP can be integrated directly into the sensors (for newer devices such as IP-based ACme's) or on a gateway / proxy server (for legacy devices such as Modbus based meters). Using HTTP, we expose sensors, meters, and actuators as resources at standardized URLs and follow the paradigm of Representational State Transfer (REST) [41]. We find that the REST-ful server-client model represents the relationship between physical data feeds and applications that use them quite nicely. In sMAP, objects are represented using JSON. We chose JSON because it already has wide tool support; creating, validating, and sending a JSON object to a RESTful service point is only a few lines of code in most languages.

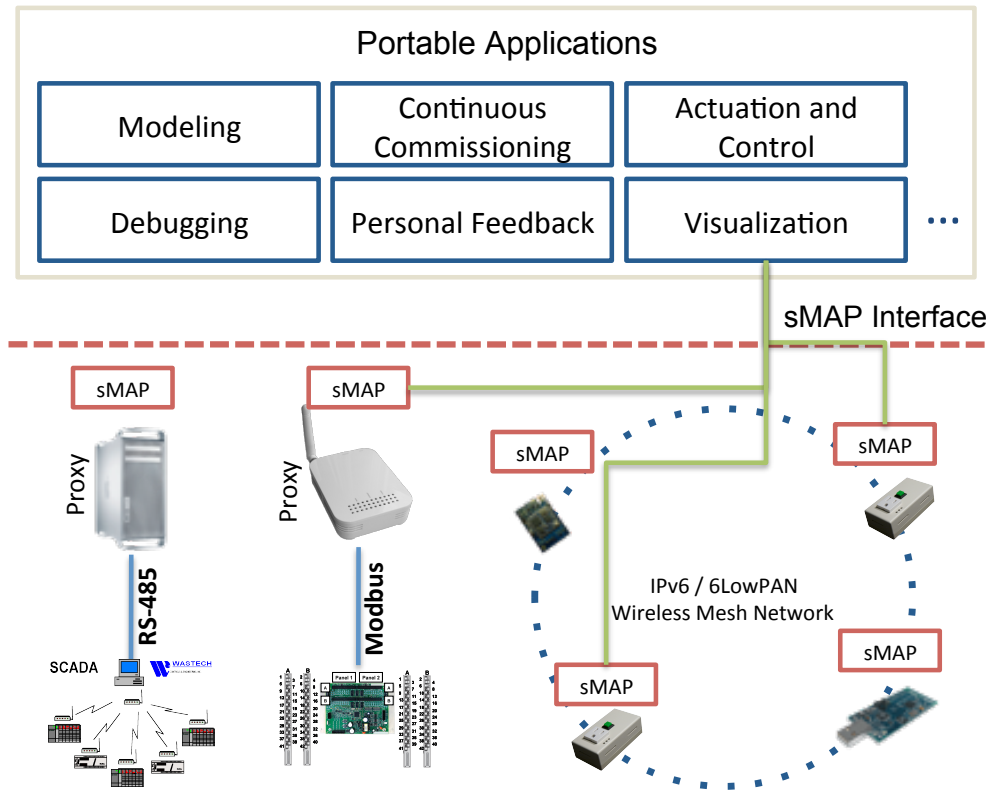


Figure 5.5: IPv6-based sensors such as ACme’s implement sMAP directly while legacy devices such as Modbus based branch meters implement sMAP via proxies. Applications interface with any metering resource uniformly via sMAP. In this example, a visualization application is used to graph real-time energy readings from both ACme’s and from branch meters.

The goal of sMAP is to represent a single device, uniformly providing all the information necessary to query all sensors, meters, and actuators. A *sensor* is a device which instantaneously samples a physical quantity (such as current or light), while a *meter* integrates such a value over time (for instance, energy or flow volume.) An *actuator* is a device which may either convert a digital quantity to an analog one or control a set point; relays, thermostats, and valves are examples of actuators.

5.2.2 ACme sMAP Service

A fragment of the sMAP service implemented by ACme is shown in Figure 5.6; this particular piece represents a meter, abstracting ACme’s energy measurements. Power is represented as a sensor and has a similar representation (not shown). sMAP is hierarchical, and allows a single device to have multiple sense points, each of which instruments a single

/	# list resource under URI root [GET]
/data	# list sense points under resource data [GET]
/[sense_point]	# select a sense points [GET]
/meter	# meters provide this service [GET]
/[channel]	# a particular channel [GET]
/reading	# meter reading [GET]
/format	# calibration and units [GET/POST]
/parameter	# sampling parameter [GET/POST]
/profile	# history of readings [GET]
/report	# create and query periodic reports [GET/POST]
/status	# device-specific status messages [GET]
/context	# instrument serial number or GUID [GET]
/reporting	# for syndication [GET]
/create	# create a new periodic report [POST]
/reports	# show current reports [GET]
/xTswsD4d	# a reporting instance [GET/DELETE]

Figure 5.6: A fragment of the sMAP RESTful interface implemented by ACme.

```

http://326.acme.berkeley.edu/data/default/meter/energy/reading
\___/ \-----/ \_____/ \___/ \___/ \_____/
|      |          |      | |      |
|      | ACme's IP address  sense_pt | channel |
|      |                      meter/sensor  object
transport

```

Figure 5.7: A typical ACme sMAP resource URI.

physical process. In this configuration, every ACme is a distinct sMAP service, therefore only has one *sense_point*. A sense point may also have multiple channels. This is useful because a single instrumented sense point, such as a Hall-effect sensor, may derive multiple related measurements such as current (RMS and peak), instantaneous power, metered energy (over some time interval), and power factor. This would be represented as a single sense point with multiple sensor channels. Since ACme's *meter* *sense_point* only provides energy measurement, it only exports a single channel called energy. A typical URI for reading the energy from an ACme sMAP service is shown in Figure 5.7.

Integrated with the sensor reading itself which is stored in the `reading` resource are all the ancillary facts necessary to interpret the data. `formatting` contains information on what is being measured and how to interpret raw readings, specifying engineering units and calibration constants. `parameter` contains information on the sampling rate in use; this can be used to determine when `reading` will update. Figure 5.8 shows the *formatting* and *parameter* resources provided by ACme. Finally, `profile` is an optional resource which allows a sMAP server to buffer a fixed number of old readings. Please see Section 6.1 for an example.

sMAP also allows periodic reports to be sent via “pushing”; however, very small embedded implementations may not support this. To request a periodic report, a client sends a

FORMATTING:

```
{
  "$schema" : {
    "$ref" : "http://schema.berkeley.edu/format.json"
  },
  "UnitofTime" : "second",
  "MeterType" : "electric",
  "UnitofMeasure" : "kW",
  "Divisor" : "24"
}
```

PARAMETER:

```
{
  "$schema" : {
    "$ref" : "http://schema.berkeley.edu/parameter.json"
  },
  "UnitofTime" : "second",
  "SamplingPeriod" : 60,
  "Settings" : { }
}
```

Figure 5.8: Formatting and parameter resources.

Client (HTTP_POST) ->

```
http://326.acme.berkeley.edu/report
      \-----/ \___/
        |         |
        ACme's IP address  object
```

With JSON object:

```
{
  "ReportResource" : "/data/325/meter/*/reading",
  "ReportDeliveryLocation" :
    "http://repo.berkeley.edu/receiver.php",
  "Period" : 60, "Minimum" : 50, "Maximum" : 100
}
```

Figure 5.9: Setting up a per-minute periodic report for the *reading* resource from ACme 326 to a data repository.

JSON object to the sMAP server specifying what resource on the server should be reported, and the URL of the destination, as shown in Figure 5.9. Periodic reporting is critical to allow sMAP to integrate into full-fledged stream processing system, since it allows a system to create event triggers and avoid inefficient polling. Please see Section 6.1 for an example.

5.3 Summary

Using an edge router and an IPv6-based ad-hoc network, sensors, meter, and actuators in this architecture can communicate with each other and with any IP endpoint, making energy data readily **accessible**, both within and externally.

However, different devices speak different languages; in order for them to **interoperate**, we create an application profile called sMAP, which defines an uniform representation for physical information. sMAP defines only three data types – sensors, meters, and actuators, and predefines all allowable units. This choice to be **restrictive** in effect enforces all devices to behave like simple data sources, which is essentially a quantity with an unit. While sMAP is not as expressive in terms of data types, it is designed to be self-descriptive, with facilities for any client to easily interpret and derive meaning from its data. From our experiences in implementing sMAP on various sensors and meters, we find it to be sufficient for most types of instruments found in buildings. sMAP also requires very little resource to implement, making it ideal for resource-constrained and older devices.

And with the use of proxies, this architecture supports an **extensive range** of existing and legacy building instruments and data sources. For example, a proxy can implement sMAP over IP on one interface while communicating with a panel meter using Modbus over RS-485 on another.

By using the RESTful server-client model to represents the relationship between physical data feeds and applications, this architecture enables data sources to easily **interoperate** with external devices and applications. Sensors, meters, and actuators are exposed as resources at standardized URLs and communicate with external applications using HTTP following the RESTful paradigm. Furthermore, by using JSON as the object representation, encoding and decoding sMAP object is easy since JSON is widely accepted.

In summary, sMAP abstracts heterogenous building data source into RESTful web services with an uniform data representation, and therefore **shortening** development time for applications and allows them to be more **portable**.

Chapter 6

Applications and Services

In our architecture, applications and services are built atop the network of meters, sensors, and actuators, and are logically distinct from both the nodes and the network, as shown in Figure 3.3. This architecture allows many distinct applications to use the underlying network concurrently. From the application’s perspective, each sense point exposes its functions through sMAP, such as configuring the node and reporting energy measurements over UDP. Applications can use these services divorced from the node-level details of TinyOS programming or energy metering. In this model, sensornet application development looks much more like web application development with its familiar *N-tier* model with the meters playing the role of an external data feed.

This chapter describes the data acquisition portion of the *ACme* application, while Chapter 7 and Chapter 8 address the data analysis, modeling, and feedback of this data.

6.1 ACme Application

In the ACme application (Figure 6.1), nodes are typically configured to report energy readings once per minute via UDP to a simple Python application daemon running on a database server. Each UDP packet includes the energy used in the previous minute as well as the average, minimum, and maximum instantaneous power observed during the same interval. The daemon parses the UDP packets, extracts the relevant readings, timestamps the data, checks for duplicate data, and inserts the readings into a MySQL database.

This application architecture marks a departure from interacting with the sensor *network* in the aggregate only, to interacting with the individual *nodes* themselves as IP endpoints. This approach provides an opportunity to leverage standard networking tools and libraries like `ping`, `wireshark`, and `netcat` to monitor the nodes, debug networking problems, and build applications.

As a collection of IP end-points, it is straightforward to introduce traditional proxies for providing additional functionalities. While we have fully implemented sMAP on individual

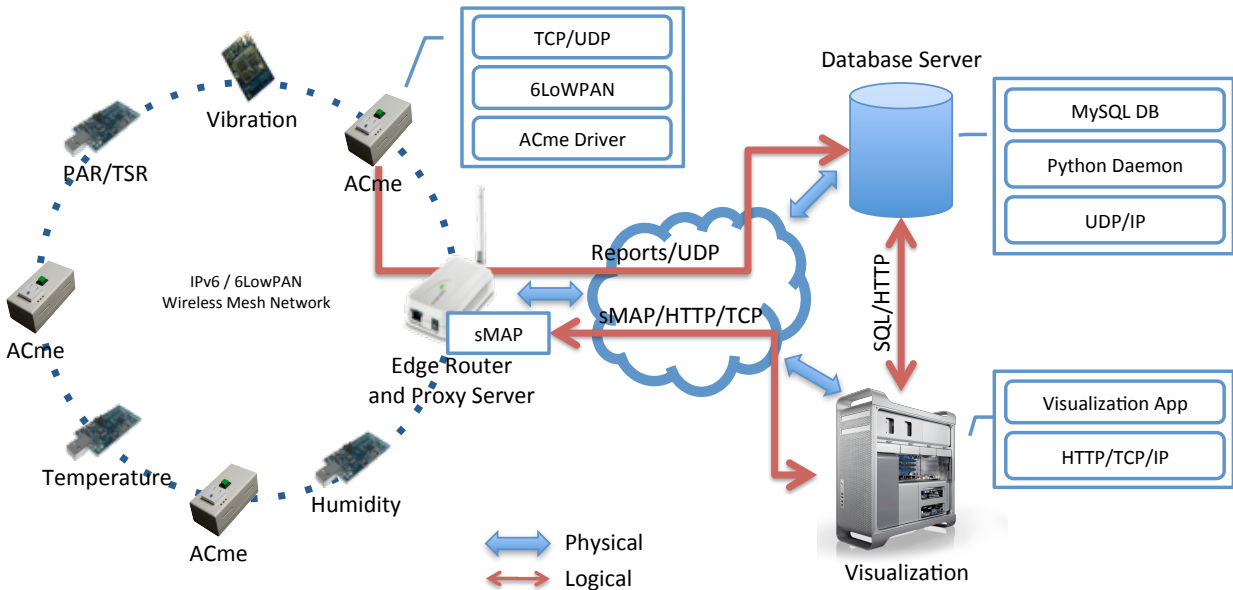


Figure 6.1: The ACme application architecture.

ACme nodes and prove that it is feasible, as seen in Section 5.2.2, we find that it is more desirable to implement sMAP at the proxy server. Since ACme’s are severely resource-constrained in processing power, memory, and network bandwidth, moving sMAP to a PC-class proxy server allows our system to service more requests without congesting the 802.15.4 network. In this configuration, the sMAP server resides at the proxy and presents a single resource with multiple sense points, each corresponding to an ACme.

Applications can access real-time data by issuing a *HTTP_GET* request to the “/reading” resource, as shown in Figure 6.2, or switch an ACme on or off by *HTTP_POST* to its “/onoff” resource, as shown in Figure 6.3.

In the visualization application, users access ACme energy data through a web interface that queries both the sMAP service for recent data and the database for historical data, and presents the results in tabular and graphical form, as shown in Figure 8.8 and 8.9.

6.2 Database and Metadata

Historical data is often used in the analysis, modeling, and visualization of energy usage in buildings. As a result, database is a commonly used service in our architecture, shared by many applications. However, without *context*, these raw measurements are simply series of incoherent numbers, and reveal little useful information. Therefore, in addition to storing time-series readings, an important part of this architecture is to store the contextual information, or *metadata*. For example, information such as locations of the sensors, types of

```

Client (HTTP_GET) ->
    http://acme.berkeley.edu/data/325/meter/energy/reading
           \-----/          \_/          \-----/
           |                   |           |
    Proxy IP address    ACme ID    Object

Server (HTTP_RESPONSE) -> Client:
{
  "$schema" : {                               // URL for schema
    "$ref" : "http://schema.berkeley.edu/reading.json"
  },
  "SummationDelivered" : "12572", // energy
  "SummationInterval" : "60",    // duration
  "LocalTime" : "1277706985"    // timestamp
}

```

Figure 6.2: Reading from an ACme.

loads, network traces, and even the weather, are all valuable in helping to understand and make sense of the energy data, as we will demonstrate later in this thesis.

In the process of the ACme experiment, we found that several types of metadata are crucial for understanding the measurements and should be stored. For example, the load being measured is distinct from the meter that measures it. We need to store both separately since it’s conceivable that a meter is time-shared between several loads. The *type* of the load allows us to classify and group loads into functional categories, and is used in Section 8.1.1. The *coordinate* of the load allows us to aggregate loads based on location, and is used in Section 8.1.2. We have also created several auxiliary tables for “book-keeping”. For example, the *binding* table helps us keep track of time of association and dis-association between loads and meters; the debug table allows us to record the states of the network; and the contact and user tables enable us to perform a true user-study where real users register ACme’s themselves via a web portal. Several *views* are also created to speed up data retrieval of aggregates.

In our deployments, around a hundred ACme’s have been reporting data to a single database over the past two years, and have produced a significant amount of data. The *energy* table, used to store raw energy readings, alone has roughly thirty million rows. As the table fills, SQL queries become slower, which forced us redesign our database schema to be more modular and easier for data-mining. However, queries that aggregate over appliance classes and time still take several seconds to perform, suggesting that more attention should be paid to database and query optimization. In our vision where tens of thousands of ubiquitous sense points continuously feed multiple data repositories, a more scalable data storage mechanism is needed, as discussed in more detail in the future works section.

```

Client (HTTP_POST) ->
    http://acme.berkeley.edu/data/325/actuator/switch/onoff
           \-----/           \_/           \___/
           |                   |           |
    Proxy IP address    ACme ID       Object

With JSON object:
{
  "$schema" : {                               // URL for schema
    "$ref" : "http://schema.berkeley.edu/onoff.json"
  },
  "Onoff" : "on",                             // turn on
  "LocalTime" : "1277707345"                 // timestamp
}

```

Figure 6.3: Actuating an ACme.

6.3 Summary

Our architecture enables developers to write **cross-platform** building applications much like web application. The ACme application uses the familiar N-tier model with meters expressed as simple data feeds.

The ACme application and the database service are only two instances of a **diverse** set of applications that can be built on top of this architecture. In fact, for the rest of this thesis, we focus on services, tools, and applications for a particular purpose – reducing energy consumption in buildings.

Visualizing where energy is consumed in real-time and at different granularities is valuable for both building managers and occupants alike, and is an important step for enabling voluntary energy reduction. We have implemented several applications on top of this architecture for visualizing energy data, including web based plotting engines and adapters for external resources such as Google PowerMeter. We refer the reader to Chapter 8, and in particular Sections 8.2 and 8.3, for a more detailed description.

Chapter 7

Analysis and Modeling

7.1 Structure of Energy Flows

Energy is distributed through a building as various subflows in a tree-like structure – the *load tree* – as shown in Figure 7.1. Visibility into the load tree is fundamental to understanding how energy is distributed and used within a building. In an ideal world, we would have *full* and *detailed* coverage of the load tree by directly monitoring not only the root of the tree, but also every single load at the leaves of the tree. However, this is rarely possible. In reality, people are often faced with a tradeoff between full coverage and detailed but partial coverage. A small number of instrumentation points close to the root of the tree provides a complete, albeit aggregated, picture of the entire building. This level of visibility is common for many buildings, and is an appropriate extent of coverage for most building managers. Appliance-level metering such as that provided by plug-load meters provides detailed power profiles of individual loads. This level of visibility is appropriate for understanding the energy consumption at the appliance level.

To ground the discussion in the remainder of this paper, we review key aspects of the load tree used in our case study. The load tree begins at the root where a single high-voltage power line connects the building to its parent substation. This line delivers power to the entire building. The incoming power is typically stepped-down into several high and low voltage lines, distributing 3-phase AC power into sets of electrical panels across multiple floors. The low voltage panels distribute 120V/208V 3-phase power and the high voltage panels distribute 277V/480V 3-phase power. Panels are normally divided by location and function. For example, in Figure 7.1, there are two low voltage panels, one for the machine room and one for supplying all the AC outlets in the northwest region of the 4th floor. There are also two high voltage panels, one for all the overhead lights and one for the fans.

From the panels, 3-phase power is fanned out into multiple 2-phase or 3-phase breakers, depending on need. In our example, the northwest region panel splits into thirty 2-phase breakers, with each breaker supplying multiple single-phase AC outlets (labeled as power

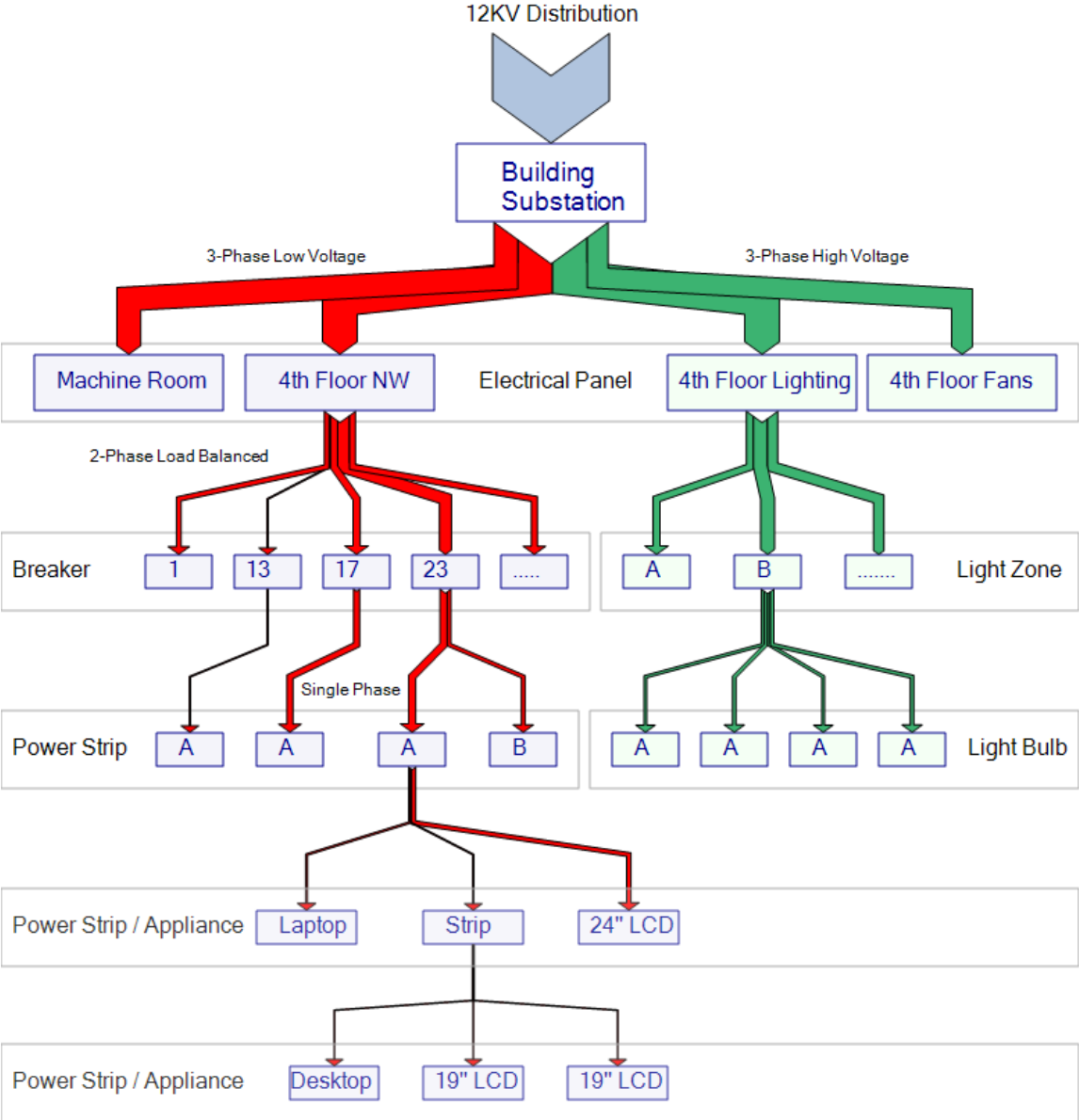


Figure 7.1: Snapshot of a small portion of the load tree in a computer science building. Voltage is stepped down from the building substation into floor-level electrical panels which in turn distribute power into either AC outlets in the low-voltage case or lights and major equipments in the high-voltage case.

strips), spread throughout the physical space in a balanced fashion. Power strips are chained to establish more levels in the load tree. The lighting panel fans out into multiple light zones in which all lights in the same zone are turned on and off simultaneously, using one or more switches. Within a particular light zone, light bulbs of varying wattages are combined together to provide an appropriate lighting level.

7.2 Modeling and Disaggregation

A substantial challenge in constructing a high-fidelity electricity measurement network is that we have neither the ability nor the budget to measure every device. To address this shortcoming, we instead create models of the behavior of each type of appliance by using measured data of similar devices. In this section, we describe strategies that use multi-modal data collected throughout our network to construct accurate appliance electricity consumption models, enabling us to infer consumption of unmetered devices.

A model of appliance behavior is only as good as the data collected to support that model; however, there are often multiple ways to measure the usage of a single appliance. For example, the electrical consumption of a refrigerator can be obtained in a variety of ways – directly measured using a power meter, estimated using a log of door opening events or a time-series of internal light measurements, or inferred through a record of proximity events where people in the vicinity of the refrigerator imply increased consumption, for example. In each of these cases, the sensors are capturing different phenomena that describe the same underlying behavior, namely the electricity consumption of a refrigerator. Thus, to begin our appliance modeling, we examine some fundamental questions concerning sampling: what behavior of the appliance should be measured? How often should we sample to capture the specific phenomena we are interested in observing? In this section, we first explore how various sample window strategies affect the conclusions we reach.

We continue by presenting multiple strategies for constructing appliance models, including using empirical measurements to calculate average energy consumption by minute, hour, and day, accounting for the behavior of individual components at the sub-appliance level (within a machine), and substituting alternate sensors to infer the consumption of electricity of loads that are not easily measured directly. For each of these cases, we show an instance of applying our strategy using data collected by our network, and discuss the applicability of these strategies to other scenarios. However, we emphasize that these strategies themselves are not novel – we simply employ them to study the traditional systems questions of coverage and fidelity in the context of a multi-modal electricity monitoring network.

7.2.1 Additivity

Like all flow graphs, an intrinsic property of energy load trees is additivity - the sum of the power of children nodes equals the power of the parent. For example, Figure 7.2 shows a

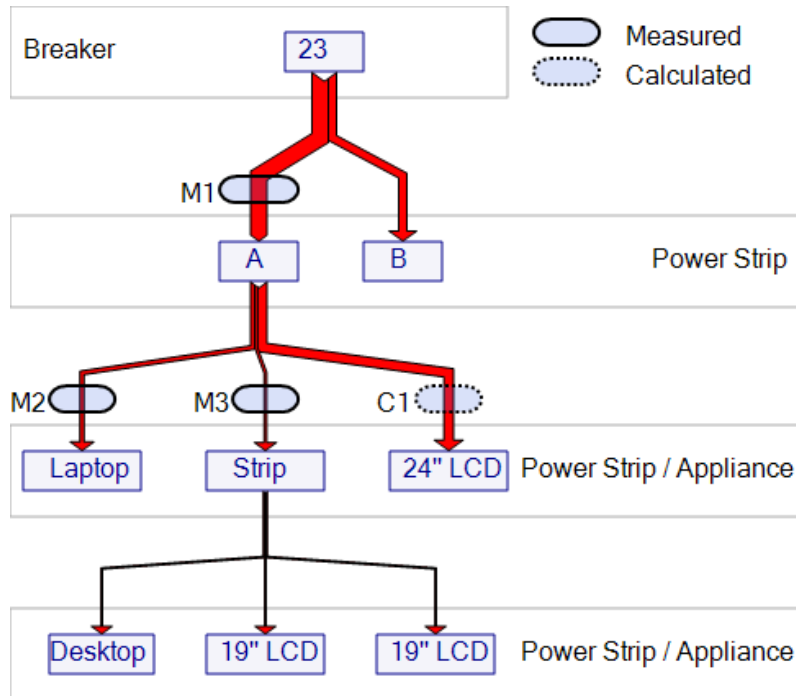


Figure 7.2: The power that flows through power strip A is the sum of the power consumed by the laptop, the 24" LCD, and the sub-power-strip. We can calculate the power at C1 by simply subtracting M2 and M3 from M1.

branch of the load tree with breaker 23 as the root. The total power flowing out of breaker 23 is the sum of the power drawn by power strips A and B; the power through power strip A is the sum of a laptop, a sub-power-strip and a 24" LCD screen.

Intuitively, this allows us to increase coverage by summing all the children, given that measurements are available for all of them, or increase fidelity by calculating the difference between a measured parent node and measured children nodes. In Figure 7.2, if we place meter M1, M2, and M3 at power strip A, the laptop, and the sub-power-strip respectively, we can calculate the power of the 24" LCD using simple subtraction. In practice, M1, M2, and M3 perform power measurements according to their own clock, and will produce readings that are not synchronized. One solution to address this issue is to perform network synchronization and time-stamp each measurement, at the cost of additional network bandwidth and energy; alternatively measurements can be time-stamped at the collection point, with less accuracy. In either case, power traces from different meters are usually shifted in phase and often differ slightly in periodicity, in addition to holes in the data. As a result, before computing differences or sums, we need to first clean up the data by resampling it, which can be easily done *post hoc*. Figure 7.3 shows both measured and calculated power of a student's desk appliances for 5 hours. The load tree for this student is shown in Figure 7.2,

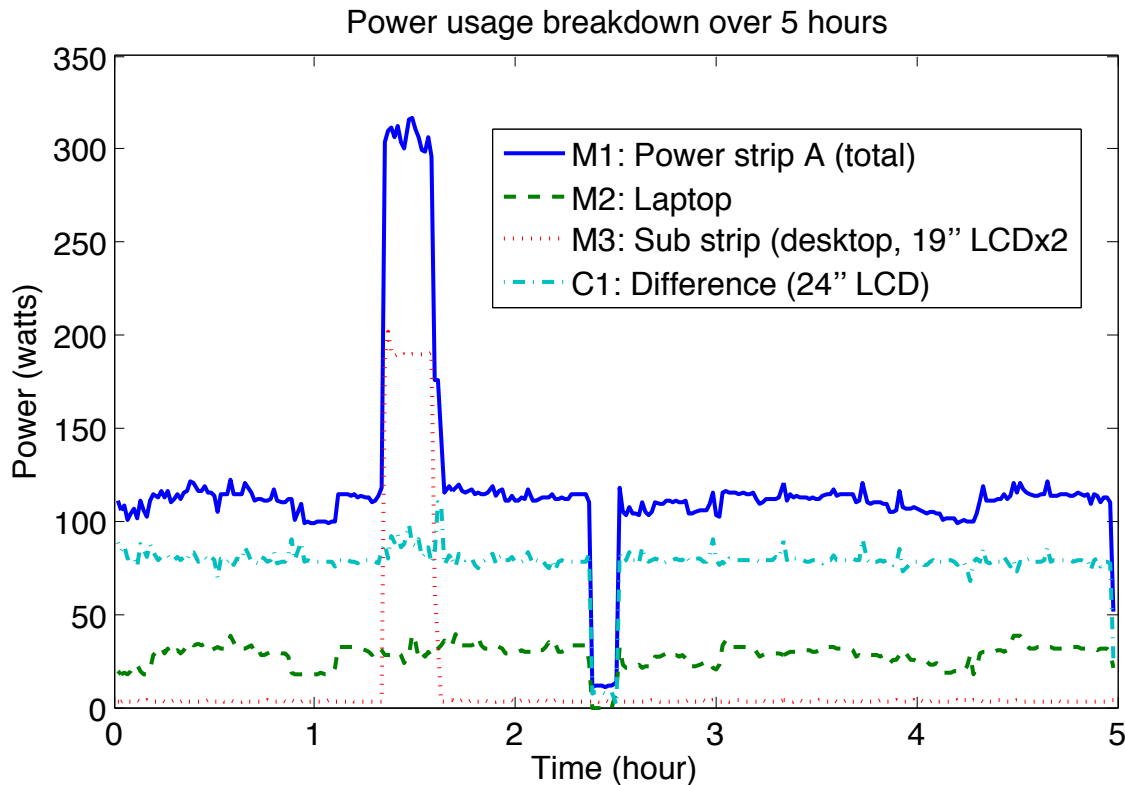


Figure 7.3: The power profile of a 24" LCD can be calculated by subtracting both the laptop metered at M2 and the sub-power-strip metered at M3 from the power strip A, metered at M1. At 1.5hr, the desktop and two 19" LCDs are turned on, using an additional 200W seen by M3 and M1 for about 20 minutes. At 2.5hr, the laptop was removed from the workspace, reducing M2 to 0W. The 24" LCD drops into standby mode after being disconnected from the laptop, drawing around 10W via subtraction.

rooted at power strip A. We can observe from this figure that the 24" LCD is not connected to the desktop but to the laptop, since its power drops from 80W to 10W when the laptop is disconnected momentarily at 2.5hr.

While applying this technique is straight-forward, it only works when all $N-1$ nodes are directly measured in a size N parent-children set. However, if the number of metered nodes is less than $N-1$, in the absence of any other disaggregation techniques, any configuration of metering is equivalent and results in the same degree of visibility.

What is perhaps a more powerful result of additivity is that one can freely combine a subset of children belonging to the same parent into a new subtree in the graph without changing its flows. Physically this corresponds to inserting a power strip to a parent outlet and connecting some subset of devices plugged into the outlet to the new power strip. This allows us to instrument the aggregate power trace of the newly formed subtree rooted at the

new power strip. In Figure 7.2, the desktop and two 19" LCDs can be thought as originally belonging to power strip A (measured by M1); a new power strip is inserted to create the current subtree, measured by M3. This gives us more flexibility in choosing what combination of loads to monitor in aggregate. We can separate devices with similar signatures and group together devices with easily detectable signatures, for optimal disaggregation, as described by [44], [56], and [27].

7.2.2 Multi-resolution

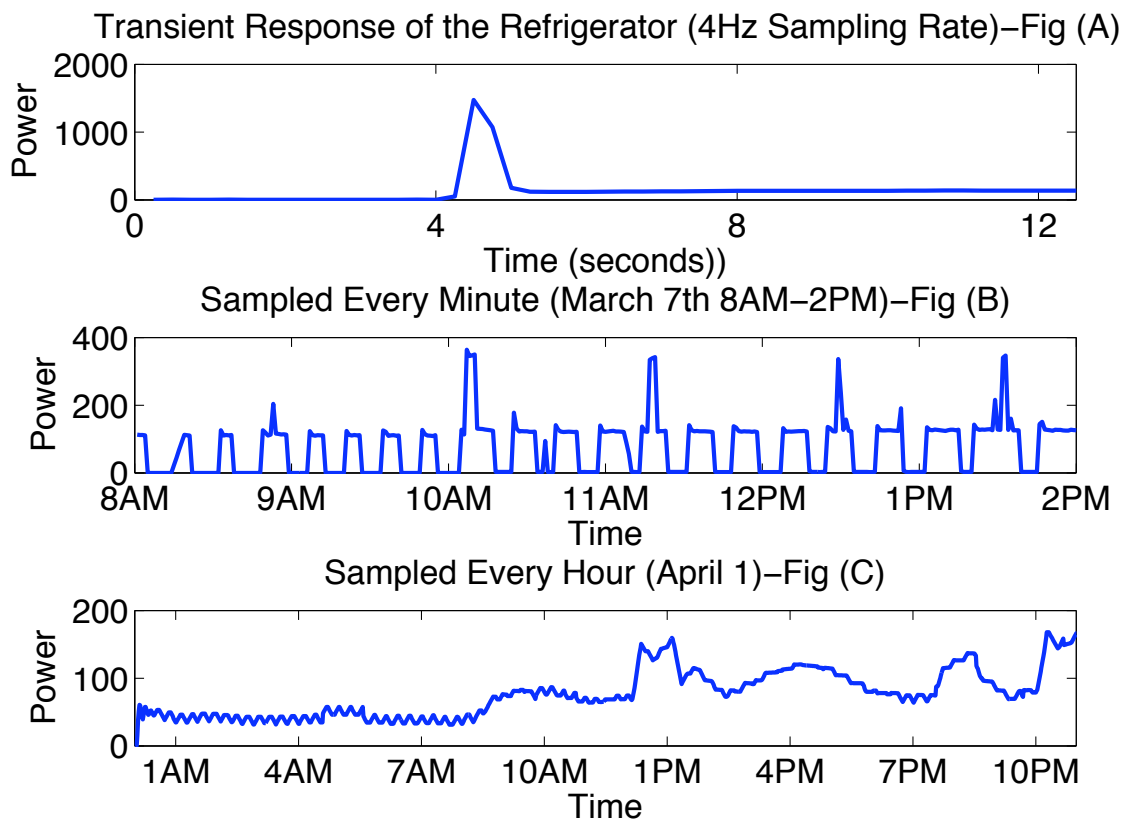


Figure 7.4: Sampling rate top to bottom: 4Hz, minute sampling, hourly sampling. (A) reveals the turn-on transient; (B) reveals the compressor and defrost modes of the refrigerator; and (C) reveals the human influence.

Certain features of a particular device can only be seen at certain resolutions. Using the power profile of a refrigerator as an example, shown in Figure 7.4, different resolutions reveal different stories. Figure 7.4 (B), at a resolution of one sample per minute, clearly shows two intrinsic characteristics or modes of the refrigerator - the compressor kicks in about every 15 minutes while the defrost cycle has a period of around 1 hour. If we zoom in

to the turn-on transition and view at a resolution of 4Hz, as shown in Figure 7.4 (A), we can observe a spike of more than 1000 watts. This observation, which is potentially important for load disaggregation algorithms, would have been lost at the minute resolution. If we step back and view its load profile at an hourly resolution, we start to see human influence in the refrigerator’s energy usage. The increase in energy consumption around 1PM and 8PM indicates increased usage during lunch and dinner. In some sense, the refrigerator becomes an instrument for characterizing the daily “power” profile of people.

It is clear that each graph reveals a different story about the refrigerator’s power usage by observing it at a particular resolution. Therefore, we need to choose an appropriate resolution, sampling, filtering, and smoothing windows to both preserve the power characteristics of the device, as well as provide clarity to the human who observes and analyzes the data.

7.2.3 Empirical Model

To construct models of devices that are not measured, we average the time-series of devices that are measured. These measured devices are of three varieties: (1) loads measured directly, (2) loads calculated using the additivity method described in Section 7.2.1, and (3) loads calculated using measured power strips. We describe each of these in turn.

The first variety is the simplest - measurements are provided directly from AC meters connected to similar devices throughout the network. Since the electricity data are recorded every minute, that is the minimum granularity of our power models, though in practice we use hourly consumption in our calculations.

Next we find each device whose power is not directly measured by an AC meter, but can be calculated because all of the other devices in its subtree are measured directly. We call this a *constrained subtree*. By leveraging the hierarchical nature of load trees, we can subtract to find the consumption of this type of device. Together with the directly measured devices of the same type, these measurements combine to form the baseline *core* power model for each appliance type.

The final step in creating individual device power models, called the *proportional scaling step*, is to find those devices that have a parent appliance measured but are not part of a constrained subtree (e.g. multiple unmeasured appliances connected to a power strip instrumented with an AC meter). In this type of case, we begin with the core power model for each appliance. We then proportionally scale the estimate of the unmeasured devices by using the available aggregate measurement from the parent device – that is, we scaled what we would expect from the composition of the unmeasured devices by what we have actually measured at the power strip. Figure 7.5 shows a core model and scaled model for four specific appliances in our deployment. If aggregate measurements are unavailable for a device, we use only the core power model for that appliance.

After this process, we arrive at models for each appliance in the system that incorporate as much relevant empirical data as possible – devices not measured are a composition of measured devices of similar type, while devices that are measured indirectly additionally

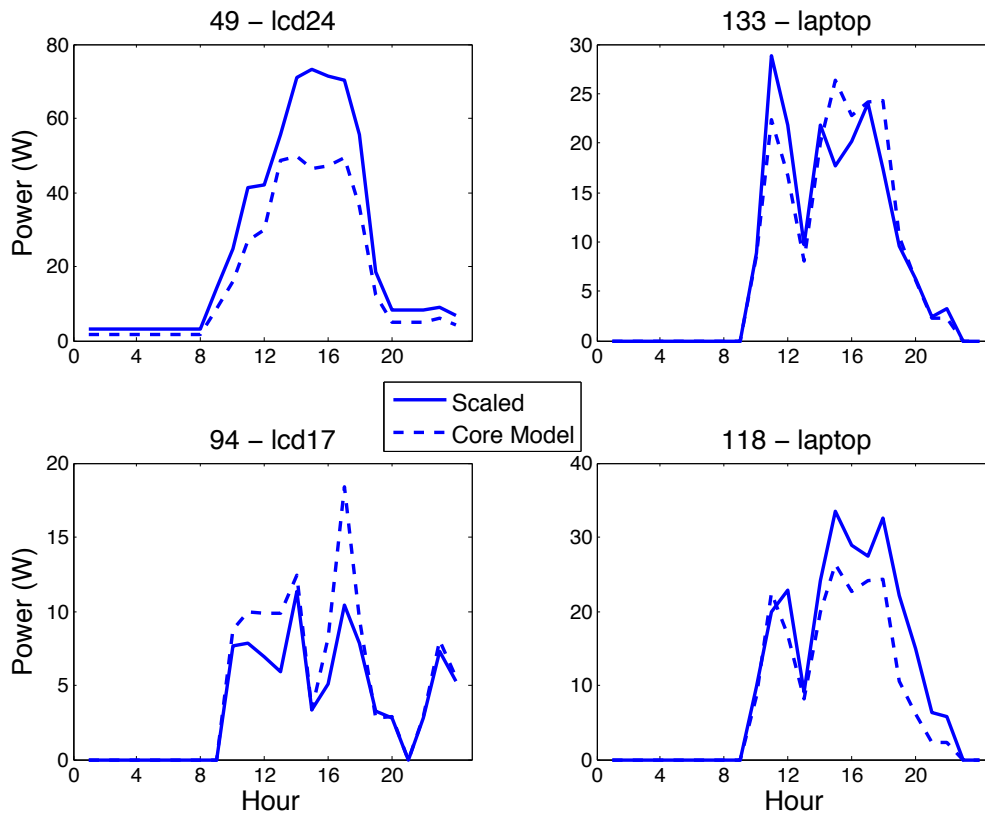


Figure 7.5: Result of calculating an improved empirical model using proportional scaling from a measured aggregate. By using upstream power measurements in addition to generic models of appliance behavior by type (“core power models”), models for specific appliances can be augmented to incorporate device-specific measurements. Here we look at a 24” LCD, a 17” LCD, and two laptops.

reflect those measurements. We use these individual appliance models as the basis for the office-wide composition models throughout this paper.

7.2.4 Appliance Signature Analysis

Modern electronic devices are a composition of many sub-components. These multi-component, multi-state devices have distinguished power traces per state that uniquely identifies them. This leads to the conjecture that perhaps the natural level of disaggregating a load tree is not at the appliance level but at the sub-components of the appliance. Figure 7.6 (A) shows the power trace of the laptop. We can pull out two components – the charging curve of its battery, as modeled in Figure 7.6 (B), and the rest of the laptop consisting primarily of the CPU, LCD, and fans, as shown in Figure 7.6 (C). In this case,

we model the laptop charging curve as an exponential decay with formula shown in Equation 7.1. This may aid in disaggregating the laptop power because now we have a generic model for the sub-components of this type of laptop.

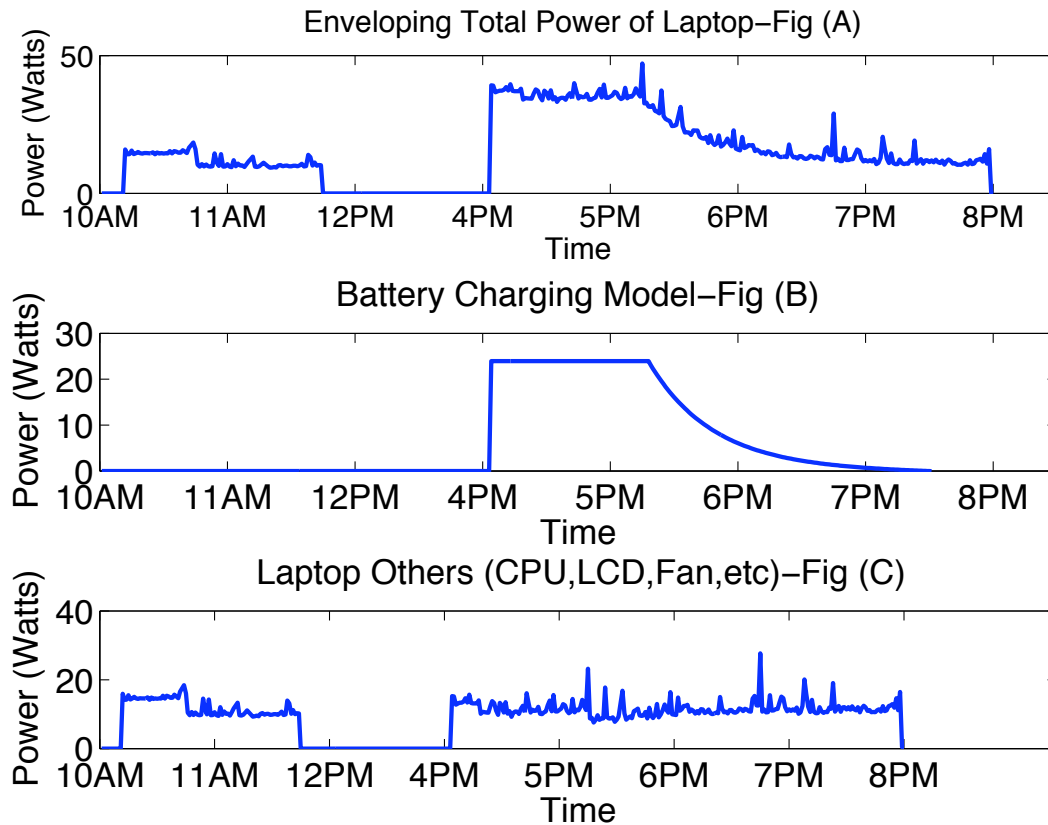


Figure 7.6: Laptop energy consumption disaggregation at the sub-component level.

$$26.33 \times e^{-3.366 \times 10^{-2}(x+4)} + 12.33e^{-7.217 \times 10^{-4}(x+4)} - 10.48 \quad (7.1)$$

Additionally, devices that exhibit daily patterns in their power traces allow for creation of accurate models of daily consumption from historical data, precluding the need to meter such devices. In Figure 7.7, we see the power consumption of the Water Dispenser over the same day of the week for three weeks, excluding the week of spring vacation. Though no clear pattern exists, a rough average can be extracted from Figure 7.7 (B), which shows the daily human influence on the device's power trace. However, looking at the cumulative graph of Figure 7.7 (C), a clear correlation in the cumulative power consumption of each Wednesday emerges.

The fundamental fact is that the power trace of an appliance is the superposition of power traces of multiple sub-components within that appliance. In the case of a laptop, the power

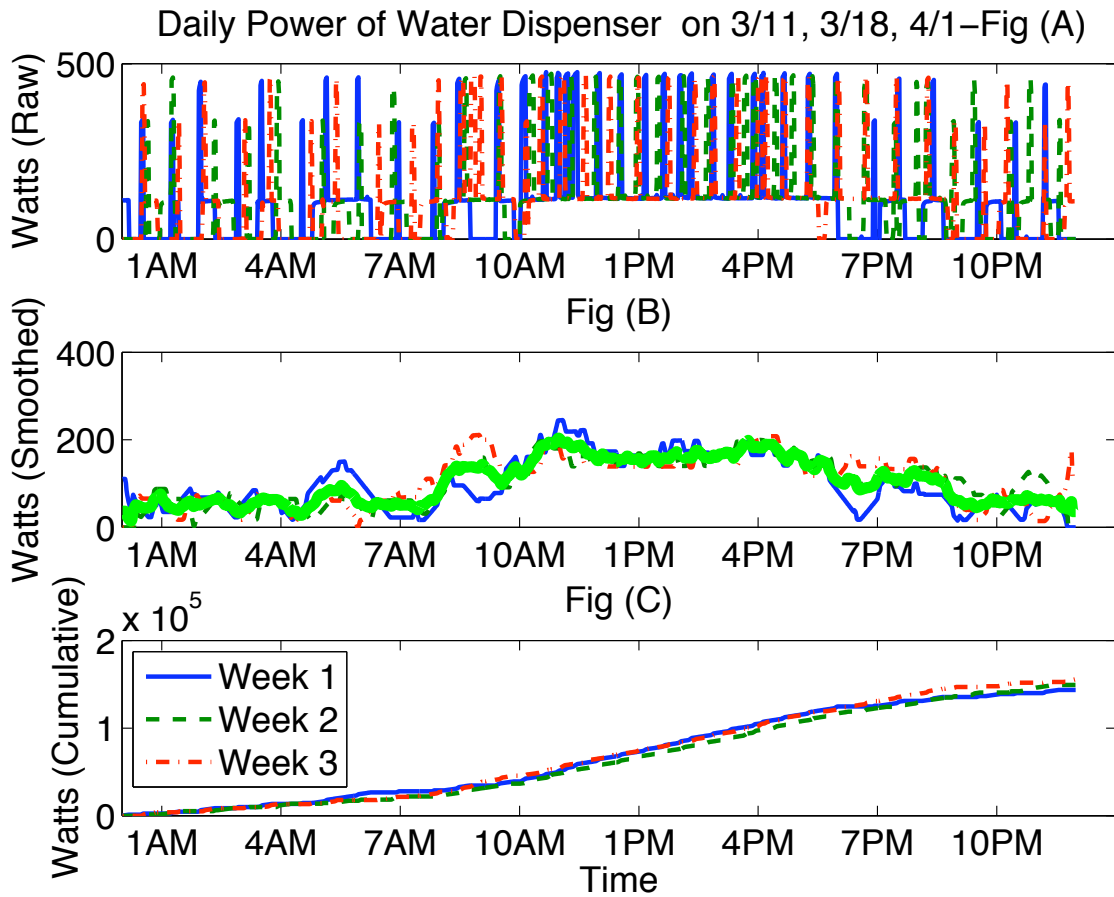


Figure 7.7: Water dispenser daily energy consumption over the same day of the week for three weeks. Notice the general trend in power consumption in (B) as well as in (C).

trace of the charge and discharge state of its battery, the CPU, and the LCD can be modeled separately; therefore, the most basic unit of disaggregation may not be the appliance, but actually the functional units within it. Moreover, the process of inference and disaggregation involves identifying not only the patterns within the device, but also the effects of human interaction with the device.

7.3 Summary

Techniques such as additivity, multi-resolution, and appliance signatures represent some of the analyses possible with high-fidelity per-appliance data, and reveal interesting phenomena of either the appliance itself or the human behind it.

But more importantly, we show that using detailed measurements of a subset of moni-

tored appliances, we can create empirical time-series based models for classes of appliances. While this idea is not novel, our approach is different from others in that our models are continuously updated in real-time, as opposed to static traces stored in a database. This approach allows us to model the time-varying nature of actual consumptions of individual appliances with reasonable accuracy. Furthermore, our approach uses empirical measurements at aggregate points in the load tree (such as power strip and panel level data) to scale individual appliance models proportionally, which bonds the error and results in accurate aggregate usage statistics as well. This method is also scalable, and becomes increasingly more accurate as more sensors are installed.

While using these models to infer real-time consumptions of unmetered devices is still far from ideal, this approach results in usage statistics, empirical or modeled, of all the end-uses, and can be re-aggregate at will to create interesting views. In this sense, modeling increases the coverage, but at lower fidelity than what would be possible with a real meter.

Chapter 8

Feedback

8.1 Re-aggregation

Disaggregating the load tree down to its leaves - the individual loads - gives us the basic building block from which we can recompose according to other grouping criteria. Re-aggregation allows one to better understand the data, identify areas for improvement, and create more actionable forms of visualization.

In this section, we present three ways to recompose our computer science building load tree - by function, by space, and by individual. Functional re-aggregation recomposes the load tree by function. For example, in our initial load tree in Figure 7.1, we can group all three LCDs into one functional group - LCD, and the laptop with other students' laptops into another group - laptop, and so forth. This is arguably the most natural form of recomposing a load tree - a direct regrouping of leaves according to their function. Spatial re-aggregation recomposes devices based on where they are. This allows us to answer questions such as "Which room uses the most energy?" or "What is the energy consumption of the machine room?" Unlike the functional re-aggregation, the unit of recomposition is often higher in the load tree, such as power strips or even circuit breakers. However, for some devices that are shared in space such as lights, one might need to subdivide it based on what proportion of it is illuminating a unit of space. Individual re-aggregation recomposes the loads by personal usage. For most appliances in an office, it is a simple one-to-one correspondence, but for some shared appliances such as refrigerators and lights, they should be attributed proportionally to different individuals over time based on some definition of usage.

To enable functional, spatial, and individual re-aggregations, we perform a detailed survey of a portion of the 4th floor, as shown in Figure 4.11. We associate meta-data with each device in a database, such as its type of appliance, where it is, and to whom it belongs. The number of loads and sensors are shown in Table 8.1. We apply techniques described in Sections 7.2 to disaggregate the load tree into 96 appliances, each with average hourly power consumptions over a day, some directly measured and some inferred.

Category	Counts
Rooms and cubicles	6
Students	28
Appliances	96
Lights	26
AC Meter	19
Light sensors	4

Table 8.1: Survey of densely instrumented region, depicted in Figure 4.11.

8.1.1 Functional Re-aggregation

Functional re-aggregation groups together loads that are of similar type. This type of aggregation is useful in studying characteristics for a particular load class, finding trends, and making comparisons between load classes. This view also helps in validating the effects of a targeted energy reduction effort by looking at the aggregate power of a particular load class over time.

For example, Figure 8.1 shows a small portion of the load tree that includes four different functions or load classes - laptop, desktop, LCD, and light. Using techniques described in the previous sections, we disaggregate the load tree into individual loads (leaves of the tree), albeit with varying degrees of accuracy. With meta-data associated with each load, in this case the function, we re-aggregate by simply summing across functions. Other statistics over these functional groups such as average and variance are often useful for understanding characteristics of loads as well.

By grouping 96 appliances and 26 lights into four functional classes, we can observe their respective contribution to the total load and their trend over the course of a day, as shown in Figure 8.2. We see that desktops are by far the largest energy consumer, accounting for more than half of the total energy. Furthermore, they stay on regardless of the time of day. Using functional re-aggregation, we identify that students and professors fail to turn off desktops after leaving the lab, wasting approximately 30kWh of energy every day. In comparison, laptops are much more efficient, using only 5% of total energy. Lighting power consumption is also very revealing. During the day, it remains at peak usage, as one might expect since most office buildings do not adjust lighting level based on ambient light. However, from this figure, we see that lights remain at maximum power from hour 20 until midnight (and likely later if we choose a longer view) even when there is very little computer usage, as evidenced by the minimal LCD power. This shows us that either the time out for the light auto-shutoff (triggered by the motion sensors) is longer than the periodicity of human movement at night, or that one or two students are keeping all the lights on. The power consumption of LCDs seems to track laptops fairly accurately, showing that in this laboratory, most students and

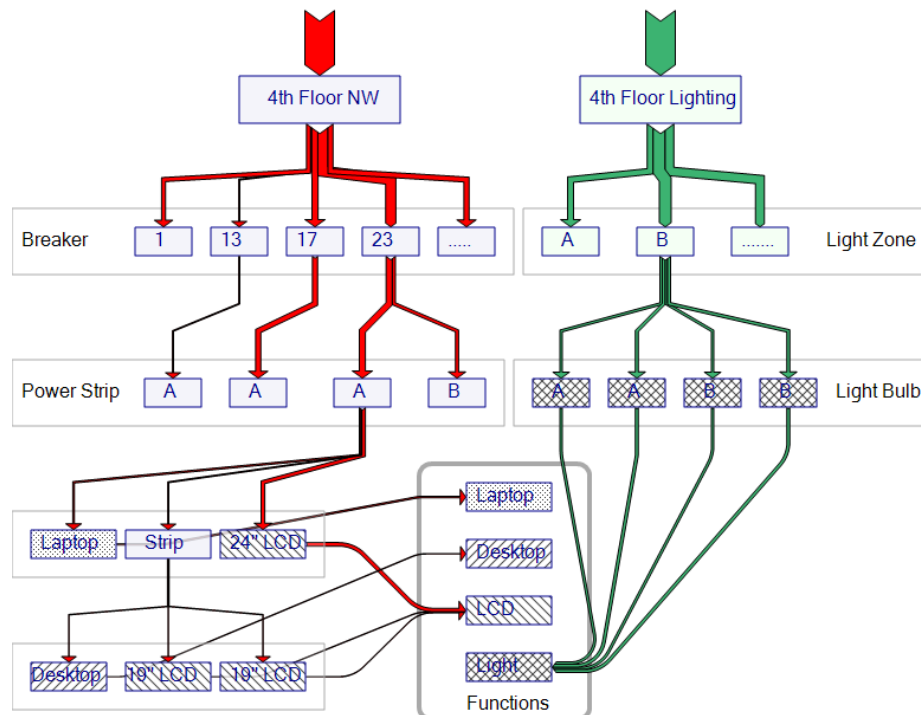


Figure 8.1: A portion of the load tree is decomposed into individual appliances, then re-aggregated into four functional classes - laptop, desktop, LCD, and light.

professors use laptops as the primary computing platform, as opposed to desktops.

8.1.2 Spatial Re-aggregation

Disaggregating the load tree provides visibility into the energy consumption of individual loads, but it does not provide visibility into how energy is consumed over space. For example, how much energy does the fourth floor use per day? What is the average power drawn by each office? What does the instantaneous power draw of a laboratory “look” like? Answering such *spatial re-aggregation* questions requires energy consumption to be projected onto space, potentially from multiple load trees, and reaggregated along spatial boundaries. Figure 8.3 illustrates this problem for a typical environment and the remainder of this section presents our initial efforts to project energy consumption over space.

For each disaggregated load in our database, we maintain coordinate fields that include a load’s approximate x, y, and z coordinates within the building. We also maintain a database of logical and physical spaces like offices, conference rooms, hallways, floors, or even the entire building. Each one of these spaces is defined by a bounding box consisting of six planes. For simplicity, we assume a rectangular building with bounding planes orthogonal

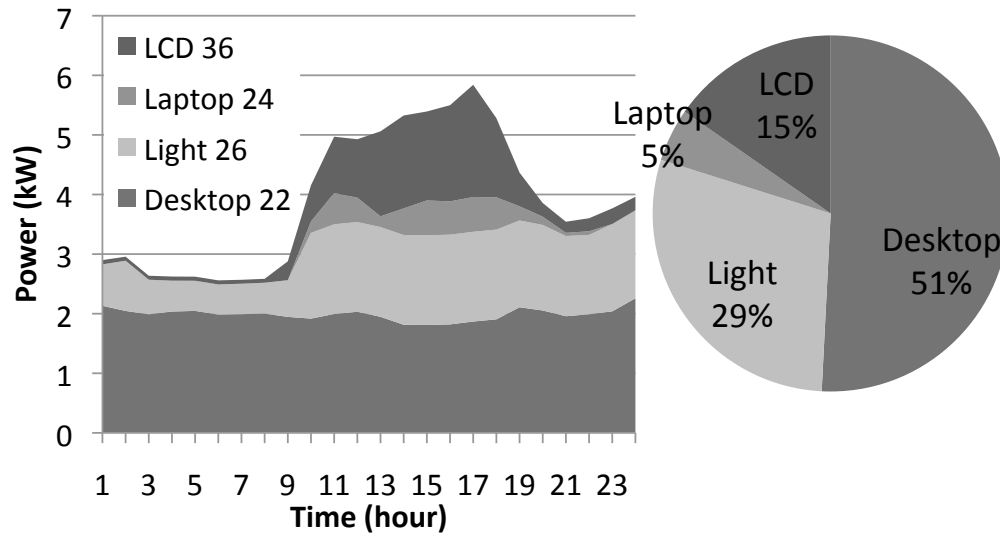


Figure 8.2: Stacked graph of aggregate power consumption grouped by functional class over a day.

to the building walls, but the basic idea can be generalized to arbitrary-shaped bounding boxes. Computing a spatial re-aggregation is then a simple matter of filtering loads by their coordinates.

Although coordinates provide point locations of energy consumption, in some cases the utility of an electrical load actually spans a wider physical area. Lighting, for example, can illuminate a single desk or an entire room. Heat can propagate across a room from a single space heater. And a projector, radio, or TV can inform or entertain all the occupants in the room. In such cases, it can be useful to distribute the energy used by a load over space using a smoothing kernel, especially if the consumption data are to be used for visualization purposes. Visualizing spatial energy consumption, however, presents an additional wrinkle to the problem: energy consumption in one space is often unrelated to consumption in an adjacent space. For example, a halogen lamp placed next to a wall usually illuminates only the room in which it is located. While it may be useful to smooth or spread such point loads over space, indiscriminately doing so leads to unexpected and perhaps inaccurate results.

We currently visualize spatial energy consumption as follows. We begin with a blueprint-style floorplan of the space. We then create a pair of two-dimensional matrices whose ratio of rows to columns match the aspect ratio of the physical space. The actual number of rows and columns depends on the desired spatial granularity (in our current design, each square foot of space is mapped to one element in the matrix). For each load in this space, we first add its consumption to the corresponding element in the first matrix, then run a load-specific smoothing kernel over the matrix, then zero all elements that fall outside of the load's nearest enclosing space (typically the surrounding walls), and finally add the resulting values to the

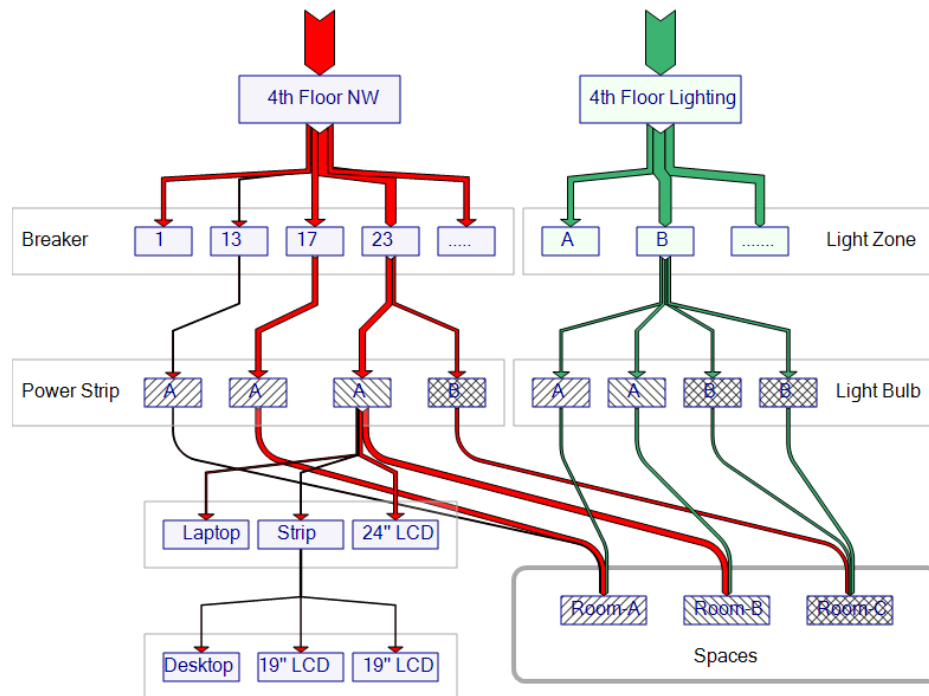


Figure 8.3: Spatial re-aggregation of the load tree. Three rooms (Room-A, Room-B, and Room-C) contain several plug loads (laptop, desktop, LCD screens) and light fixtures. Directly measuring the energy used by a particular space may not be possible if power arrives via multiple load trees, like wall outlets and built-in lighting.

second matrix. This process is repeated for each load in the space to be visualized. Once all loads have been smoothed, truncated, and projected onto the second matrix, we generate a contour plot over the second matrix and project a transparent copy of the contour onto the floorplan. Figure 8.4 shows the output of this process.

8.1.3 Individual re-aggregation

A common question among building occupants is the deceptively simple, “What is *my* energy consumption?” None of the re-aggregation approaches already described answer this question. Functional re-aggregation can provide the energy usage of each load, but a single load may be shared by multiple people, as in the case of the lights or a refrigerator. Spatial re-aggregation can provide visibility into the energy consumed over space, and identify power hotspots in a building, but it does not tie this usage to specific occupants. In this section, we present our preliminary approach to addressing the *individual re-aggregation* question, illustrated in Figure 8.5, and some thoughts on extending this approach.

To estimate the energy consumed directly by an individual, each dedicated load is tagged

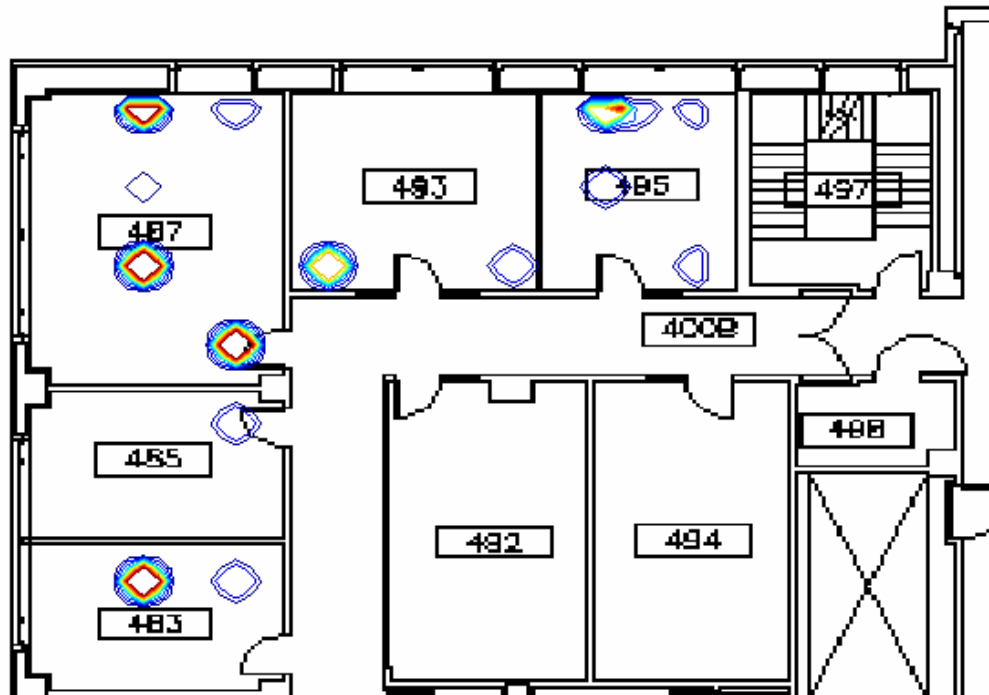


Figure 8.4: The spatial energy consumption across several offices in an office building. The empirically measured average power draw of several loads is averaged over space using a square smoothing kernel with rounded corners. This figure highlights that the occupants in room 487 have a higher average power demand than those in neighboring offices.

with its owner’s information. Computing the *direct* energy footprint becomes a matter of aggregating the consumption across all loads “owned” by a single user. Computing the fractional contribution due to shared load is a little a bit more challenging. For each user, we identify the nearest enclosing space (e.g. office or wing), aggregate the energy usage of all shared loads in the space (e.g. lights, refrigerator), and then divide by the number of occupants whose “home” coordinates fall within that enclosing space. The resulting figure provides a *per capita* energy usage of the shared resources.

Figure 8.6 illustrates the individual re-aggregation gathered from our empirical data. The data reveal a number of interesting observations. Most notably, individual consumptions have a wide distribution and the “typical” consumer is the one who does not appear to actually spend time near his or her devices. Of the daily average energy usage of 89 kWh, 63 kWh is due to dedicated resources and 29 kWh is due to shared resources. For just over half of the occupants, the dedicated component of consumption dominates the shared component. However, for just under half the occupants, the shared contribution is slightly greater than the dedicated one (which is due almost exclusively to idle LCD panels). This suggests that some users who do not use their LCD panels regularly should turn them off.

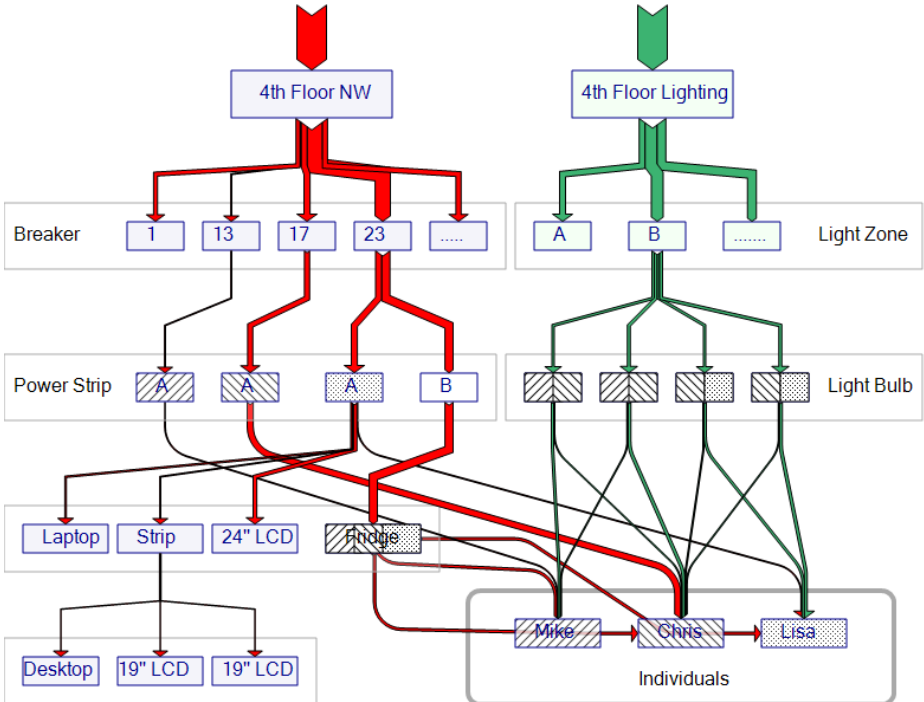


Figure 8.5: Individual re-aggregation of the load tree. Three building occupants use a mix of electrical loads. Some loads, e.g. laptops or desktops, are dedicated to one user while other loads, e.g. lights or a refrigerator are shared by some subset of the occupants. Directly measuring an occupant’s fractional usage may not be possible so approximation techniques are needed.

While Figure 8.6 provides some insight into individual re-aggregations, and provides some guidance into how group usage could be reduced, it does not easily distinguish useful and wasted energy. For example, if a user is near his or her computer and actively using the LCD screen, we might call that energy consumption useful. In contrast, an LCD screen that displays the screen manufacturer’s logo as a screen saver all night while the user is asleep might be called wasted energy. Currently, we do not distinguish between these two cases. One possibility for this additional level of re-aggregation that we are currently exploring is to have users carry a wireless keyfob or amulet that listens for radio transmissions and periodically reports the list of recently heard nodes to a server. Using this crude form of localization, we may then be able to further decompose individual consumption into times when a load spends energy in the presence of its owner and times when it spends energy in the absence of its owner. Additionally, by attributing energy consumption to a mobile user, we can readily calculate how an individual’s energy footprint spreads across time and space.

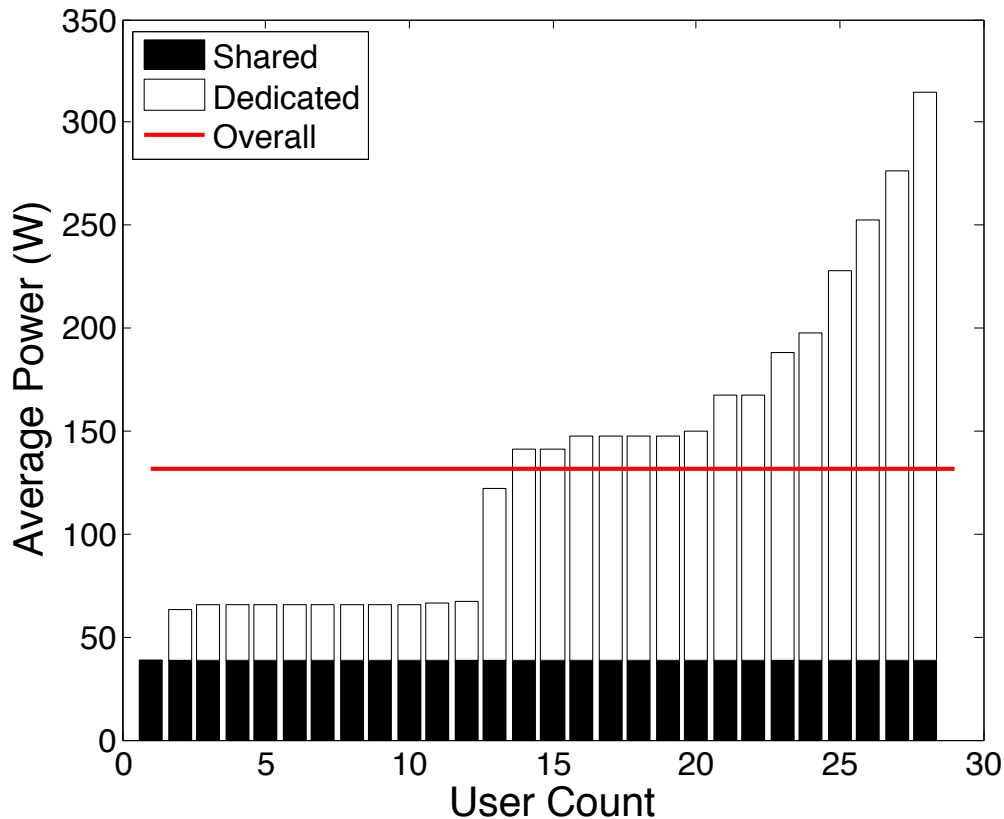


Figure 8.6: The distribution of per-occupant average power across a group of nearby occupants. The overall average draw (across all occupants in the monitored area) is 132 W. Each bar includes the fractional contribution from the shared lights and refrigerator (38.5 W), as well as the dedicated contribution from each user’s personal loads like laptops, desktop, and monitors.

8.1.4 Electrical Re-aggregation

Beyond the scope of individual consumers of electricity, another critically important stakeholder in the electricity tree of an office building is the facility personnel. These people are principally concerned with aggregate electricity measurements, and often only monitor electricity at a whole-building scale. Though the functional re-aggregation in Section 8.1.1 provides many powerful analytical methods to building personnel, this section aims to examine one more: a re-aggregation based on electrical system structure. This is especially relevant to facility personnel because they are often the sole stakeholder aware of the electrical layout of a building: the distribution, breakers, and branches of the load tree. To display the benefit of having measurements categorized by electrical structure, we examine a particular anecdote that arises as an artifact of our office’s electrical system but may provide

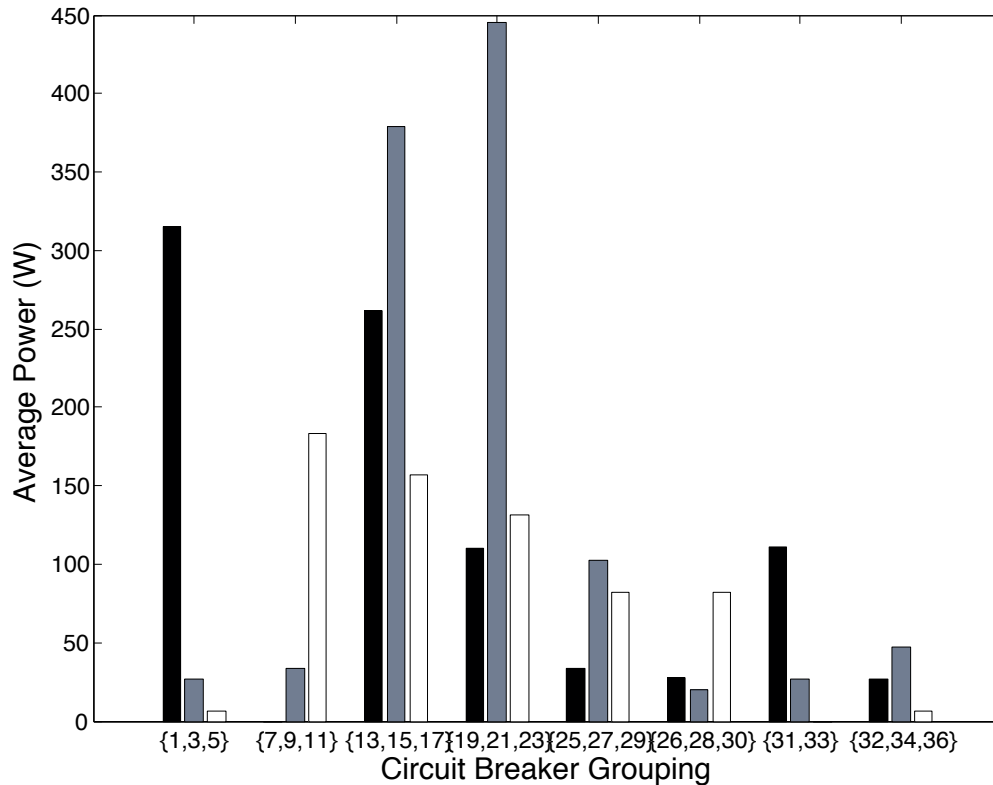


Figure 8.7: Per-circuit loads in our office deployment. Loads can be migrated within each grouping simply by moving to a different socket on the same outlet. This data could guide a building manager to improve load balancing among circuits.

guidance for future electricity re-aggregation exercises.

In the office setting under study, electrical sockets are made available to people through a series of *floorboxes*, where a floorbox consists of connections to all the wired services of the building: telephone lines, Ethernet ports, and electricity. In each floorbox, there are two sets of two outlets each, with each set of outlets in a particular floorbox on a different breaker circuit. However, the pair of breaker circuits in a floorbox is unique – that is, no other floorbox will have access to the same two circuits. Additionally, to further distribute load in the office, an individual circuit is always shared between two floorboxes, resulting in a situation of *circuit triples*, whereby each member of the triple shares exactly one floorbox with each other member of the triple. This design is intended as a strategy for distributing load across the circuits; Figure 8.7 shows both the electrical load of each circuit as well as the distribution of loads across a circuit triple. In cases where the loads on a circuit triple are extremely disparate (e.g. circuits 1, 3, and 5), distributing load can be achieved by simply relocating plug-loads from one outlet in a floorbox to another, protecting the electrical system from overload.

8.2 Visualization

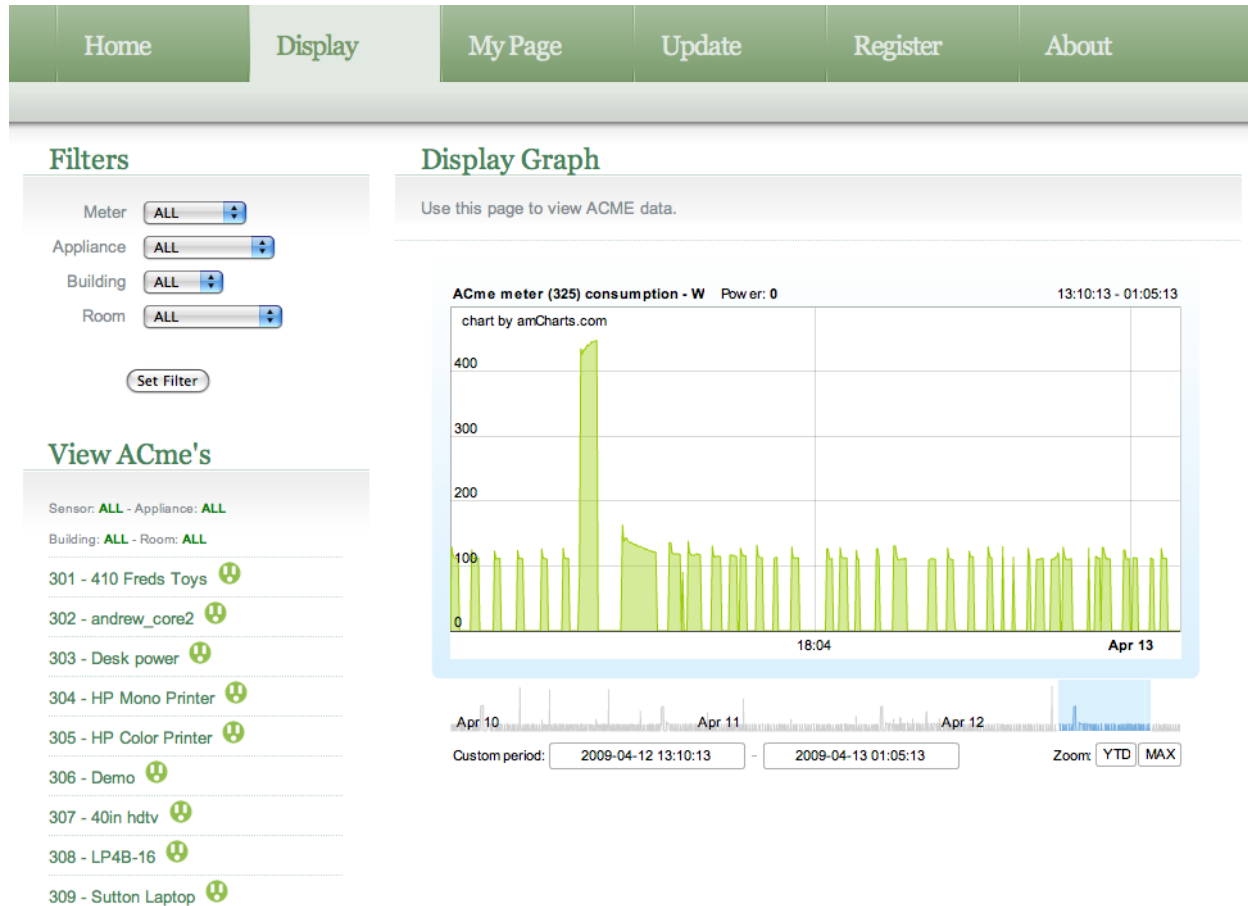


Figure 8.8: A web portal for monitoring of individual energy usages in real-time.

In the previous section, we explored various ways to re-aggregate energy usage to provide insights into how and where energy is consumed. Functional and electrical re-aggregations result in visualizations that are particularly useful for building managers to identify waste and problems; spatial and individual re-aggregations provide feedback to building occupants and incentivize saving.

In contrast to homes, in a commercial building environment, occupants usually do not directly pay for the energy costs. As a result of this lack of feedback and ownership, building occupants often use energy in a wastefully manner. An important aspect of this work is to establish a connection between individuals working in a commercial building and the share of energy that he or she is responsible for.

The first step in this direction is providing a way for the individual to easily *visualize* electrical loads that he or she is directly responsible for in real-time. To support this, we

have created a web portal where individual occupants register for accounts and associate appliances under their own “home pages”, as shown in Figure 8.8.

This web page is a simple *application* on top of our architecture, and communicates with the database service using mostly PHP calls and some javascripts. The top navigation allows an user to perform several actions. The “register” link allows an user to create a new account. The “update” link allows user to associate or update energy streams, in this case ACme plug-load data or vibration and light intensive data, with the current user account. “My page” displays and graphs energy streams associated with current user’s account. And “display” shows all data streams. Data streams such as ACme’s are listed on the left; clicking to a particular ACme will graph its measurements in a frame on the right and real-time.

The visualization is Flash-based (using the amCharts [2] graphing library) and provides robust user interaction. User can zoom in to minute level granularity or zoom out to see historical usage over past few months. This provides a simple tool for individuals to visualize and understand his or her energy usage at the plug-load level and compare to the past. Over the past year, we have a total of 140 registered users with 106 data feeds. Data feeds can be shared between users, creating a community where energy usage is transparent to everyone.

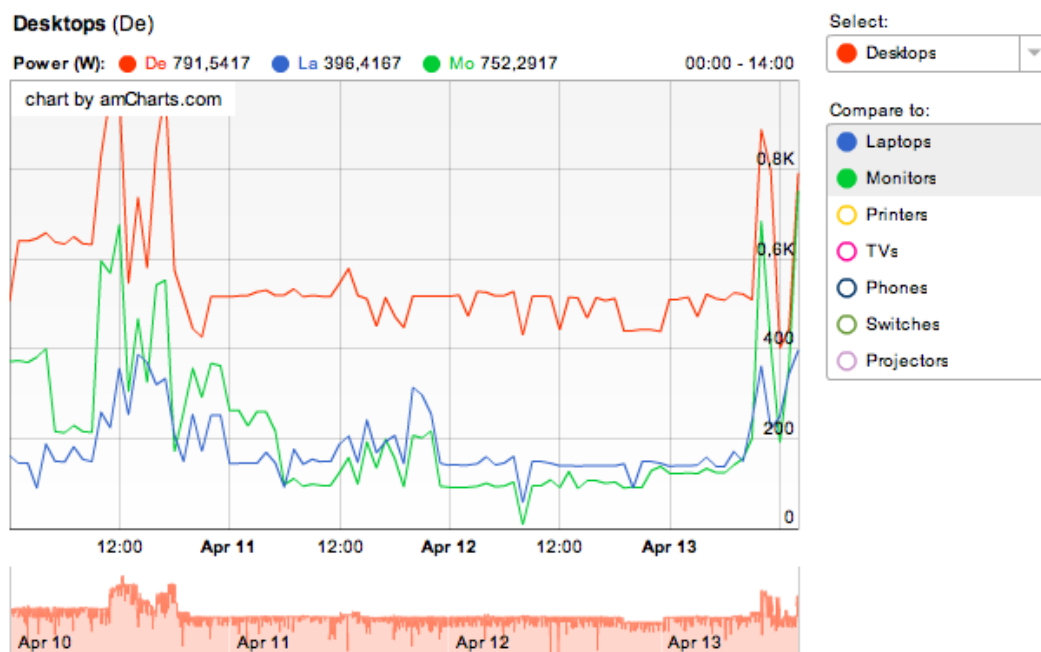


Figure 8.9: A web based visualization displaying the real-time power usages grouped by appliances types.

Because we store meta-data such as types of appliances during the association step, we can readily compute and display energy breakdown by appliance type in real-time, as shown in Figure 8.9. In this graph, we re-aggregate energy usage based on the appliance types, such

as “desktops”, “laptops”, “monitors”, and etc. The user can select which types to graph and compare to. Because multiple users could access this page simultaneously and at fast rate, we need a way to *cache* the computed data for this graph. To reduce load, a simple cron job runs once every minute and computes the aggregate electrical usage based on appliance type and stores the results in a database table. This application represent yet another simple application implemented on top of our architecture. This view allows occupants to visualize their collective effect on energy usage and identify top energy consuming appliances.

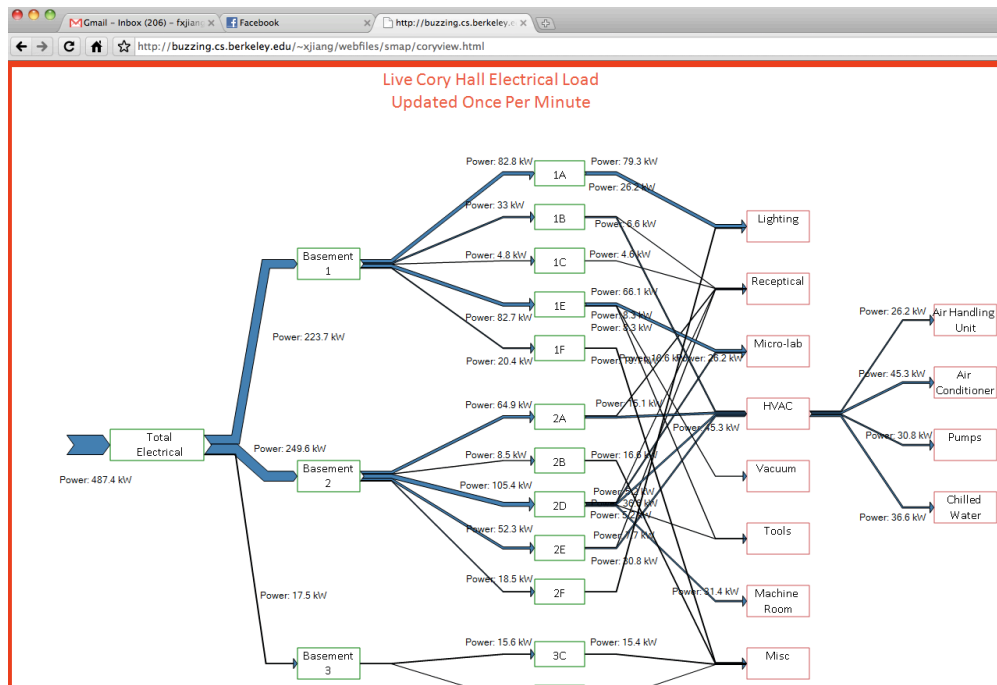


Figure 8.10: A live Sankey diagram showing the energy flows in the building electrical load tree.

While these visualizations are valuable to individual occupants, building managers of large commercial buildings are often less concerned with individual loads, but more concerned with the overall health of the building, as embodied by its electrical load tree. Our architecture also enable real-time monitoring and visualization of energy flows in the load tree, as shown in Figure 8.10. This application is a live Sankey [81] diagram using data from mostly circuit level meters. The green rectangles represent nodes in the load tree and correspond to physical sense points. The lines represents the edges in the load tree and is proportional to the energy flowing on that edge. The structure of this diagram is created based on the electrical drawing of the building; energy flows from left (building substation) to right, as the load tree branches deeper into the leaves. The leaves are further re-aggregated into end uses, represented by red rectangles. The widths of flows are updated dynamically

at a rate of once per minute, using real-time data from the circuit meters. This visualization enables building managers to identify problems and respond in a timely fashion.

8.3 Google PowerMeter

Google PowerMeter [43] is a free web based energy visualization “gadget” created by Google. PowerMeter can be integrated in an user’s iGoogle tool and displays real-time power traces based on data from utilities and compatible meters. Google PowerMeter uses the Google GData API as the data protocol and XML as the interchange format. In some ways, Google PowerMeter and sMAP serves the same purpose of decoupling energy data suppliers from consumers. However they are also very different and are designed with different purposes in mind, as illustrated in Table 8.2.

sMAP	Google PowerMeter
Device-centric API (rooted at the device level)	User and provider centric API (rooted per user)
Measurement hierarchy built into directory	Flat measurement structure
Does not include security model	Authentication built into directory hierarchy
Integrated contextual/meta data	Context through naming conventions

Table 8.2: Comparison of sMAP and Google PowerMeter

Google PowerMeter is a great visualization and feedback tool and provides simple analysis and prediction on personal energy usages. It is no different from any other building applications that consumes energy data. From Google PowerMeter’s perspective, ACme is just another GData-compatible meter. To enable Google users to visualize their plug-load usage in real-time through their iGoogle page, we have implemented a simple adapter application that bridges ACme data to Google PowerMeter. During initialization, variables corresponding to ACme’s are created under the “Berkeley” namespace; and authentication tokens returned from Google are stored. A python based application periodically reads ACme energy data using the sMAP interface, then creates and uploads the appropriate XML-encoded energy data structures to the corresponding variables’ “durMeasurement” attributes at 15 minute intervals.

As shown in Figure 8.11, Google PowerMeter widget displays the historical data in real-time at 15 minutes granularity and provides a number of statistics such as energy used per day, approximate predicated cost in dollars, and breakdown by segments of the day. While fairly rudimentary, these types of personal energy feedback has been shown to incentivizes the user to use less energy [31][65].

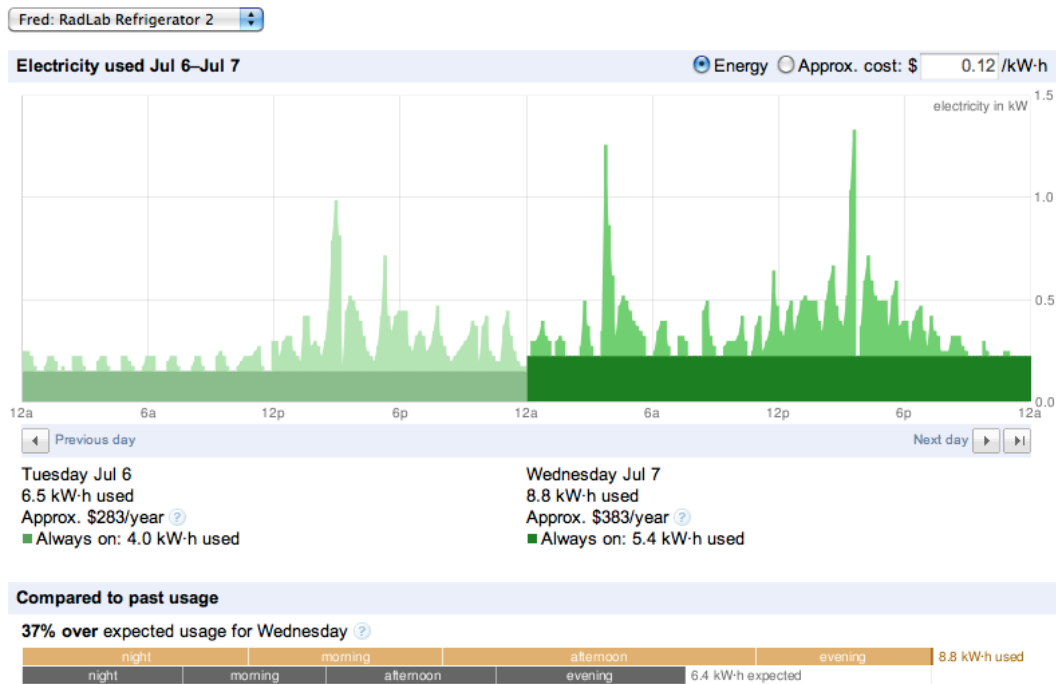


Figure 8.11: Google PowerMeter displaying real-time power usage of a refrigerator measured by an ACme. It provides simple historical statistics and future usage predictions.

8.4 Mobile Applications

Because our architecture uses standard web technologies such as HTTP and JSON, it is easy to develop compatible applications. Most platforms and programming frameworks readily support interfacing with our system, including iPhoneOS and Android based smart phones. Mobile phones represent a unique opportunity for us – it provides localization of the occupant; it provides a screen for visualizing energy usage information; and it provides a means of feedback. Many interesting applications can be created on this near-ubiquitous platform in conjunction with our building monitoring system.

By using the phone as a simple visualization tool, one could monitor energy usages continuously and set alarms based on various criteria. For example, a parent could monitor a child’s computer energy usage while at work (and subsequently infer activity) or set an alarm if lights at home continue to consume energy during work hours (in the case when lights are left on unintentionally). By using the phone as a localization device, one could compute what portion of energy consumed in a space is a particular occupant responsible for, based on proximity to loads, or by actively scanning bar codes on appliances pre-registered to that occupant. This enables breaking-down energy usage by person, and assign responsibilities. From the user’s point of view, one could map out his or her personal energy footprint by

summing all the energy attributed to him or her throughout the day. By utilizing the phone’s ability as an input device, one could provide feedback to the building. For example, occupants of a common space could collectively “vote” on the desirable temperature and arrives at some mean value.

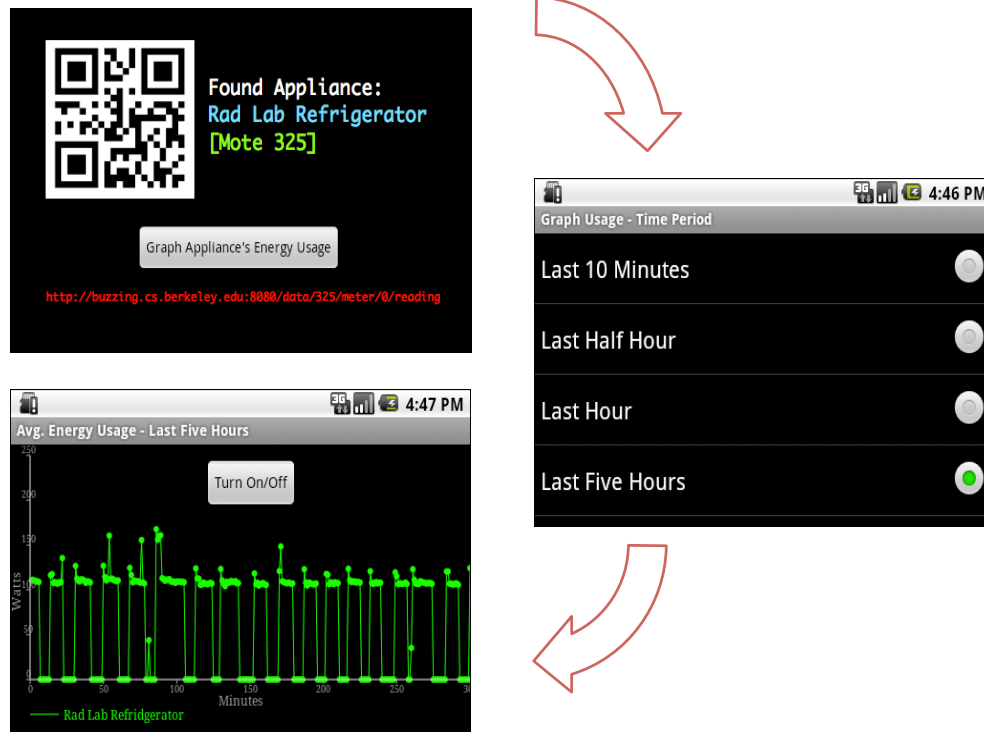


Figure 8.12: An Android application that scans a bar code associated with an appliance monitored by an ACme and graphs its power consumption in real-time.

One particular mobile application, created by two undergraduate students Jeff Hsu and Sushant Shankar, is a simple Android application that interfaces with our building system and provides feedback to the user. This application scans a bar code associated with an appliance monitored by an ACme and graphs its power consumption in real-time, and allows the user to turn it on or off, as shown in Figure 8.12. Please refer to [34] for more details.

8.5 Summary

Because this architecture provides high-fidelity energy monitoring data at the per-appliance level, and the proposed process flow infers unmetered devices through modeling, we can essentially approximate full coverage of all loads in the load-tree. This enables a **wide range** of visualization for feeding back to occupants, as seen in this chapter.

There are many different ways to categorize these visualizations, such as by target platform, visualization fidelity, dynamics of visualization, and granularity in the load tree. However, since our ultimate goal is to reduce consumption by voluntary reductions, we group these visualizations here by the target audience.

For building managers, functional and electrical re-aggregations help to identify waste and problems. Combined with meta-data, functional re-aggregation is valuable as a real-time diagnostic and energy auditing tool for identifying the building's energy response to a particular stimulus. The live Sankey visualization also helps building managers to identify problems or areas for improvement in the electrical load tree.

To incentivize individuals to reduce energy, it is important to establish a connection between individuals' actions and their impact on the energy consumption of the building. The live web portal provides real-time feedback on the individual appliances an occupant is responsible for, and gives visual feedback on their consumptions. The Google PowerMeter gadget provides simple historical statistics and future usage predictions to further induce reduction. Additionally, mobile applications built on top of this architecture allows occupants to view appliance usages in-situ via geo-located bar codes, and provides the ability for turning them on and off remotely. This combination of localization, visualization, and user feedback, provides an ideal platform for individualized energy tracking and actuation. Spatial and individual re-aggregations provide feedback to building occupants and incentivize saving by comparing individual usages to others' usages in a social, and possibly competitive, environment.

Chapter 9

RadLab as a Green Building Testbed

Motivated by research that real-time, per-appliance electricity usage feedback can induce behavioral changes that lead to 10% to 20% reductions in usage [37], we have built and deployed a network of nearly 50 ACme meters across a Computer Science department, and a web application to monitor the data produced by these nodes.

The goal of this deployment is to enable interactive, near real-time monitoring of numerous small loads and a few of the larger ones found in an office environment so that occupants can understand their electricity usage patterns and alter their behavior to reduce their energy footprint. This deployment enables building occupants to observe the energy usage of their everyday devices. It also allows us to better understand where energy goes inside the building as a whole.

The green building deployment also serves as a concrete design point to evaluate the utility and performance of the ACme node and the `blip` network as a platform for energy metering and control applications. This application builds on numerous campus-wide efforts to monitor and control energy usage to control operating costs and reduce the campus carbon footprint.

9.1 Network

A network of meters is essential to obtain time-correlated coverage of energy consumption over large spatial and temporal extents. Traditional energy monitoring solutions use a serial port or other wired backchannel to connect instruments to data loggers, which is not scalable or practical at large scales. Our wireless network allows quick deployment and instrumentation of a large number of AC plug meters by using an ad-hoc network layer which provides IP connectivity to the meters without requiring either wiring installation or support infrastructure, as described in more detail in [51]. The network provides connectivity between the meters and other networks using an IP router.

Figure 9.1 shows the connectivity graph of 44 wireless sensors over the deployment space.

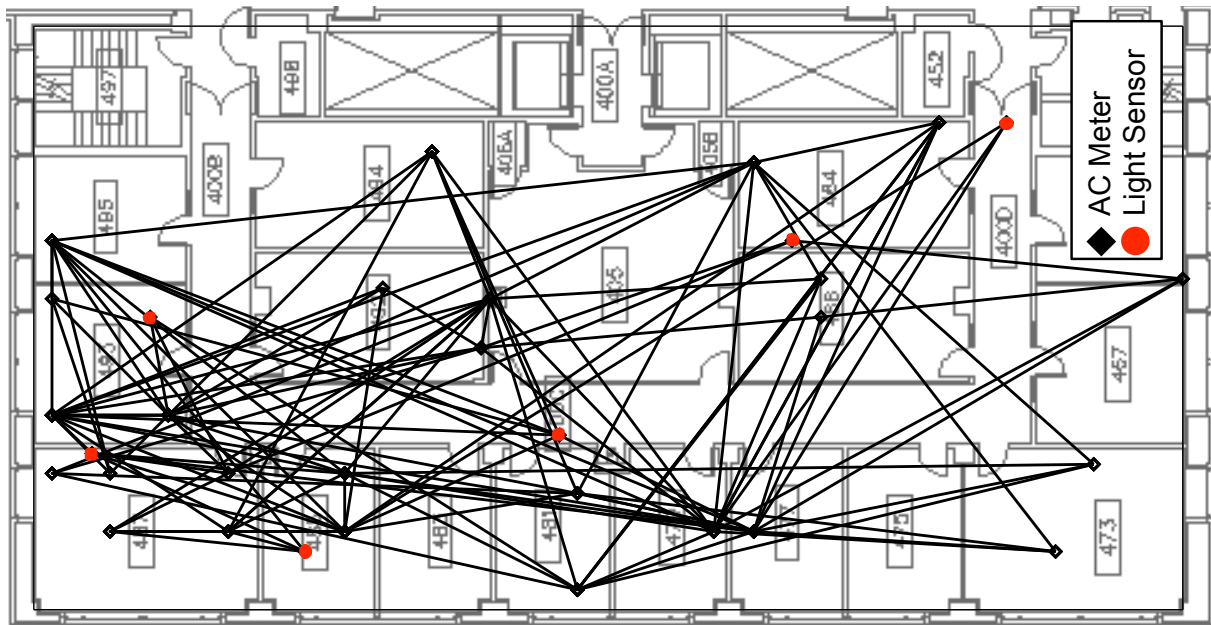


Figure 9.1: Location and connectivity graph of AC meters and light sensors. There are a total of 38 AC meters, 28 of which are spatially distinct, and 6 light sensors. The vibration sensor, not shown here, is attached to the HVAC infrastructure in the basement of the building.

They form a moderately dense network with an average degree of 4. Each sensor node is configured to report energy readings once per minute via UDP to a simple daemon process running on a server. Each UDP packet includes a sequence number, the energy used in the previous minute, and average, minimum, maximum, and last instantaneous power observed during this interval. The server process timestamps the readings and stores them in a database for later processing.

9.2 Usage Profiling

Building occupants check out one or more ACme meters from our research group, register their meters online, plug the loads they want to monitor into the meter, and plug the meter into a power outlet. Figure 9.2 shows a typical ACme meter installation as well the aggregate statistics of the type and number of plug-loads currently monitored with this system in our self-contained lab.

In addition to enabling individuals to view and adapt their electricity, data collected from this application will allow us to build models of energy usage aggregated over space, time, person, load type, or other factors.



Load Type	Count
Desktop	9
Other	6
Microwave	5
Projector	5
Monitor	5
Laptop	6
Printer	4
Lamp	3
Switch	2
Phone	1
TV	1
Refrigerator	1
Coffee Maker	1

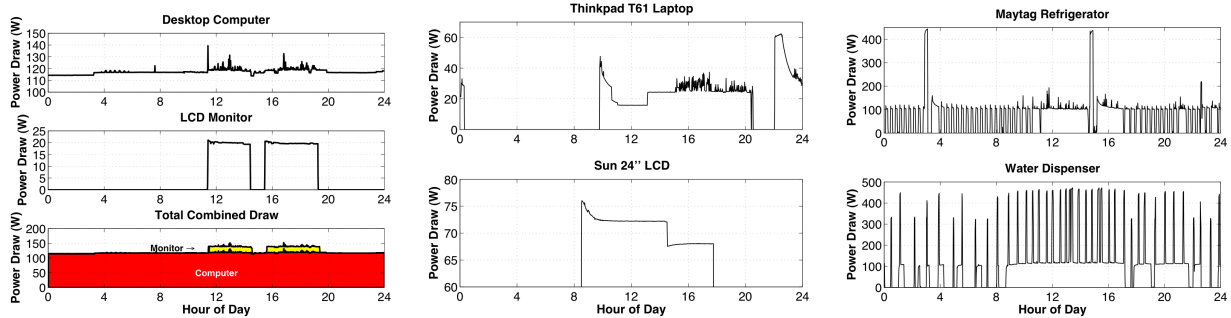
Figure 9.2: The type and number of plug-loads currently monitored using ACme nodes.

9.3 Deployment Model

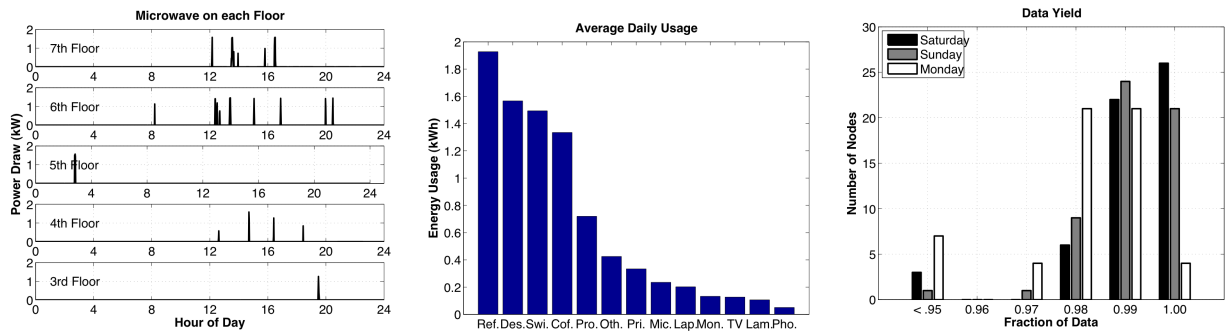
To demonstrate that we have met our goal of unplanned deployment with minimal configuration, we walk through the steps necessary to deploy our network configured with the data-collection application. For a given deployment, the first step is to deploy an edge router which provides routing between our embedded IP network and the Internet. For us, this typically means connecting one of our edge routers to an available LAN. If the subnet is IPv6 enabled, no particular configuration is necessary as the router will automatically acquire an IPv6 prefix. If not, we configure a tunnel to an external Point of Presence using one of several freely available IPv6 tunnel brokers. The ACme’s are pre-configured with the IPv6 address of a machine “in the cloud” to which they report their data. While it would be possible to map the IPv6 subnet of sensors to an IPv4 address range, or used NAT, we feel that an IPv6-only network represents a better long-term perspective on how billions of future devices will access the Internet.

When a sensor node is plugged in, it is automatically configured with a globally-routable IPv6 address using IPv6 stateless autoconfiguration. Once a node establishes a link with a neighbor, the node sends periodic reports to the central server containing power measurements. To enable users to configure their nodes and views their data, each device is labeled with a random 32-bit key. In order to “claim” a particular sensor node, users need only to type that string in the application’s web page, which then displays the device’s data.

9.4 Results



(a) Power profile of a student’s computer and monitor over a typical day. (b) Power profile of a laptop and a large LCD over a typical day. (c) Power draw of a refrigerator and a water dispenser on a typical day.



(d) Power draw of the microwaves on the 3rd (bottom figure) to 7th on a Monday averaged across all three day period. (top figure) floors on a typical day. (e) Energy usage by plug-load type loads of the particular type. (f) Data yield from 49 nodes over a Monday.

Figure 9.3: Power traces from various appliances under measurement.

This section presents several cuts through the data we have collected continuously from our operational network over the past 4 months. The “green building” application lets users pose a range of questions that allow users to meter and visualize the energy usage of their everyday devices. For example, Figure 9.3(a) shows the power profile of a student’s computer and monitor during a typical Monday. Monitor power draw is roughly correlated with computer usage, since the monitor enters a power saving mode after a few minutes of inactivity. However, the energy savings that accrue from the monitor’s automatic power down mode pale in comparison to the idle power of the computer, as evident from their total combined draw shown in the stacked bottom plot. This observation might cause a student to power down his or her computer over the weekend or even every night. Figure 9.3(b) shows the power profile of a different system– a laptop and a 24” LCD screen (not connected to each other).

We observe that the laptop has much better power management than the desktop, consuming less than half of the desktop during active and going to sleep when not used. The 24" LCD, on the other hand, draws more power than the laptop when active. The LCD power trace also reveals something we did not previously notice: at hour 14, the laptop connected to the LCD was unplugged, but instead of going to sleep, it displays a vendor logo and cable disconnect warning. This consumes 68W for almost four hours before sleeping. This suggests that we could save 272Wh per day just by turning off the LCD manually after disconnecting the video source. Figure 9.3(c) shows the power draw of a refrigerator and a water dispenser over a day. In the top graph, we note that the compressor during the refrigeration cycle draws around 100W and starts every 15 minutes; the heating element is active during the defrost cycle and draws 440W every 12 hours. Similarly, the cooling cycles in the water dispenser consume less than heating in the water dispenser but has a higher duty-cycle. Unsurprisingly, the average power is highest during the work day— from about 10 AM to 5 PM.

In addition to these individual questions, the application lets one pose queries over the entire population of nodes and their plug-loads. For example, Figure 9.3(d) shows the power draw of the microwaves across a building. Although it is clear from this data that microwaves contribute relatively little to the department's carbon footprint, it suggests natural ways of using electrical consumption for human activity monitoring. A different aggregation of the data yields the energy usage, in kWh, of each type of plug-load, as shown in Figure 9.3(e). The load types, from left to right, are: (1) Refrigerator, (2) Desktop, (3) Switch, (4) Coffee Maker, (5) Projector, (6) Other, (7) Printer, (8) Microwave, (9) Laptop, (10), Monitor, (11) TV, (12) Lamp, and (13) Phone.

This view allows us to build models of energy usage that can be used to estimate the contribution of the various small and large loads based on the number of such loads and the average energy usage of their load class. It also focuses attention on the things that matter (in this case, the desktops and network switches), since they are heaviest consumers and they are numerous. Figure 9.4 shows a breakdown of energy usage by appliance type in a laboratory that encompasses half of an entire floor in the building. The average measured power is multiplied by the number of appliances to yield the total energy footprint of each plug-load type. The pie chart shows that desktops are the largest energy consumer, followed by LCDs and laptops. The time series shows that (1) desktops consume a lot of energy even when idle; (2) projector consumption is determined only by utilization; and (3) some monitors draw significant power when the source is unplugged.

9.5 Improvements

By far, the most beneficial result in building a dense energy measurement network is opening the door to a potential wealth of electricity savings. In this section, we relate two specific anecdotes that resulted in reduced electricity consumption, the first of which resulted

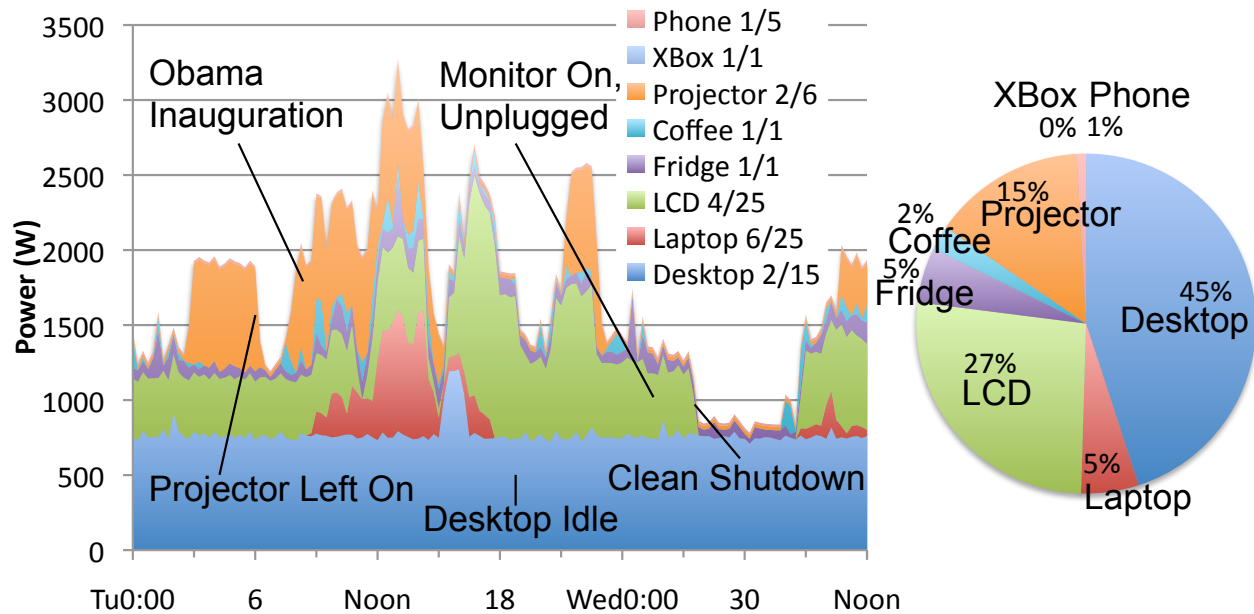


Figure 9.4: Aggregated energy usage by appliance type normalized by number of appliances in an enclosed environment.

in up to a 32% reduction in LCD electricity consumption, and the second of which slashed the sleep power of a desktop computer by a factor of 50.

Figure 9.5 provides an aggregate consumption of all of the LCD monitors measured in the laboratory over the course of four weeks. Table 9.1 provides the weekday average for each five-day period, represented by a white rectangle on the graph. At the beginning of this timeframe, the authors made a presentation to the laboratory that enumerated the waste of LCD monitors unplugged from laptops but not turned off. This, along with generally spreading the word about the electricity measurement network, led to a significant initial reduction in LCD energy consumption - nearly a third from week 1 to week 2. However, as the effect of this impulse diffused, so did the reduction in electricity consumption, nearly returning to week 1 levels by week 4. Without drawing any concrete conclusions from this experiment, it appears that a single notice, though initially powerful, may taper off in effect over time without reinforcement.

Week	1	2	3	4
Average Power (kW)	5.92	4.03	4.53	5.70
Change (%)	N/A	-31.9%	+12.5%	+25.8%

Table 9.1: Work-week average aggregate power for measured LCDs, along with week-by-week percentage changes.

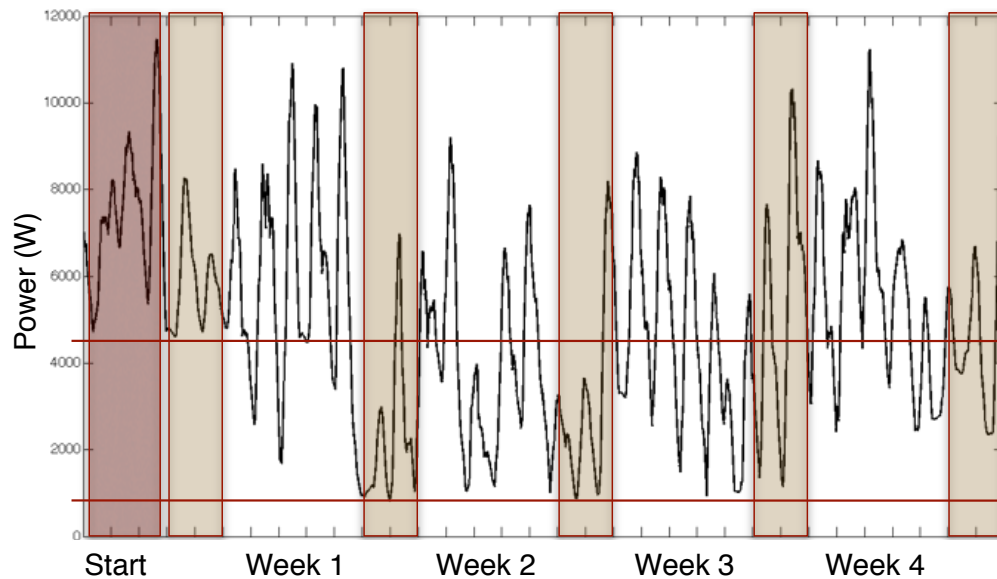


Figure 9.5: Aggregate trace of measured LCDs in the laboratory over a four-week period. The white rectangles represent the work week, shaded vertical strips highlight the weekends, and the solid horizontal line represents the average consumption over the entire duration.

We note that the aggregated measurements may include loads that are themselves aggregated, such as power strips that contain both an LCD and a desktop. Though this places the meaning of the absolute values from the figure in jeopardy, we feel that the relative pattern is not obfuscated.

The second anecdote deals with our experience reducing the idle energy consumption of desktops running Windows as their operating system. Our analysis begins with observing that the idle average wattage of a typical Windows machine in the laboratory is roughly 119.0 Watts. As this is excessive for a computer doing no useful work, we sought out various energy management solutions, including both third-party packages as well as Windows' own energy management suite. We decided on a third-party product named Auto Shutdown Manager [39], which provides the ability to set both thresholds and more complex rules for shutting down and waking up the PC and its components.

In our experience, the software detected keep-alive packet traffic that was preventing any of the existing energy-saving algorithms from taking hold by triggering the Wake-on-LAN function of the machine's network card. By blocking these requests and experimenting with a variety of thresholding schemes (i.e., go to sleep when there is no mouse or keyboard activity for 10 minutes and CPU utilization remains less than 40% for the entire duration), we were able to get the machine to regularly go to sleep when not in use. The effect of this change is shown in Figure 9.6, a time-series of power readings during the day when the auto shutdown software was enabled. The average power usage after the change is only 2% of the

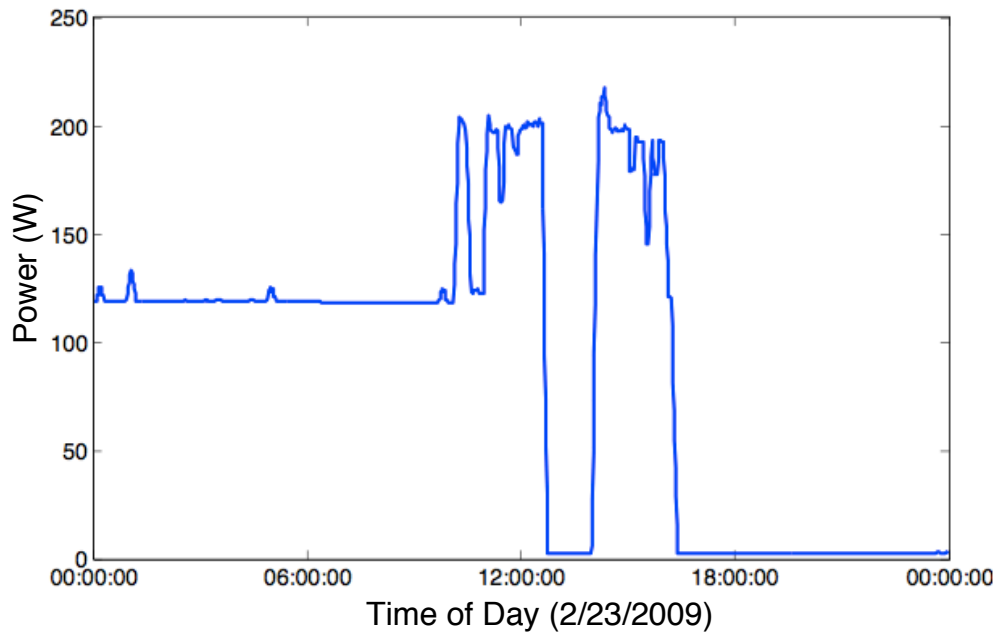


Figure 9.6: This is a trace of a student’s desktop machine running Windows XP before and after installing power management software. The machine was reconfigured on this day to drop into full sleep mode after 15 minutes of idleness, reducing the idle power from 119.0 Watts to 2.4 Watts.

usage prior to the change (2.4 W versus 119.0 W). No doubt this change makes a significant difference, but it does come with the tradeoff of increased wake-up times – machines take about 10 seconds to become operational when triggered after sleeping.

To quantify the potential total energy reduction if all of the desktop PCs in the laboratory were running Windows and experienced similar proportional reductions in their idle power usage, we applied the same rules as on the initial machine to traces of power consumption from each of the other measured desktops. We continued this analysis by recalculating the models for each desktop in the lab, and present the results in Figure 9.7. This graph shows a 15% reduction in the peak envelope and a 30% reduction in the baseline desktop energy consumption. Going forward, we intend to deploy this software more widely, both in the lab as well as externally.

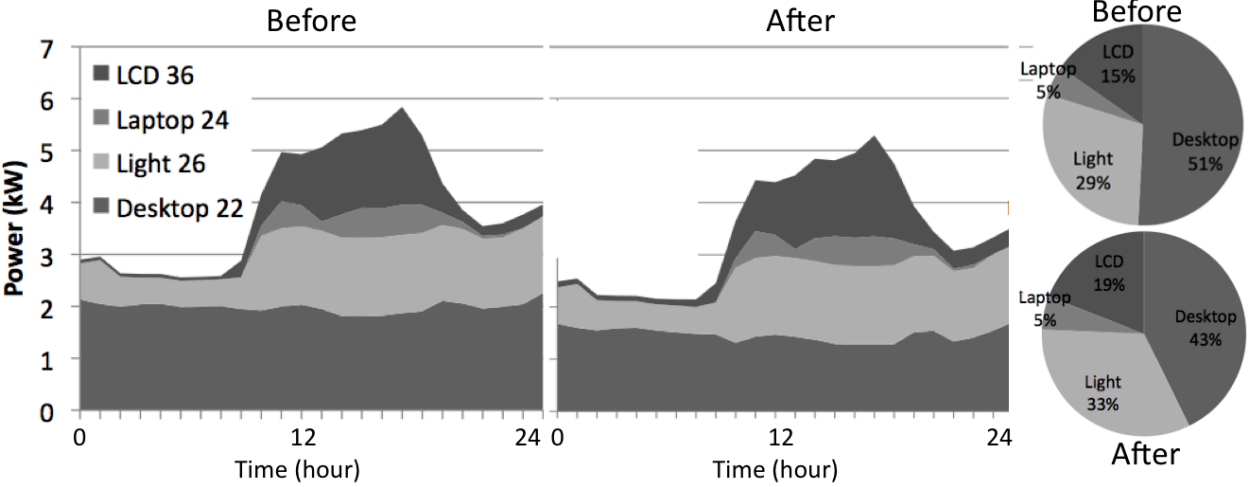


Figure 9.7: Estimated saving of the laboratory energy consumption if all of the desktops incorporated energy management software.

Chapter 10

Ongoing and Future Research

10.1 Deployments

Over the course of this project, we have found ACme to be a versatile platform for enabling various energy related research projects. Several on-going deployments are using ACme as a building block for energy data collection. Figure 10.1 shows some of these deployments, described in more detail below.

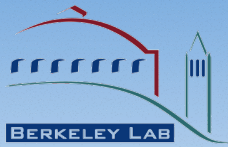
The Miscellaneous and Electronic Loads (MELs) project is a collaborative effort between Lawrence Berkeley National Laboratory (LBNL) and UC Berkeley, to study the energy use of plug load devices in several different buildings using ACme devices. 500 ACmes are used in combination with other sensors to monitor and analyze energy usage of Building 90 in LBNL. The goal is to develop a metering methodology that will allow us to measure, for individual devices, the long-term energy consumption, typical load (power) profile, and time spent in various power modes. In addition, we want to develop a categorization for collections of devices, such as by device type and end-use, based on load profiles.

The Cory Hall project is a building energy monitoring project aimed at developing a methodology for instrumenting large scale commercial buildings and as a build-to-grid testbed. Following the architecture proposed in this thesis, various meters and transducers at different granularities are used to instrument different points in the load tree, including Dent PowerScout 18's, PSL PQube, Dent RoCoill, Dent Split-Core CT, PSL Split-Core CT, and 100 ACmes. The ACme's are used to monitor plug-loads in computer labs and offices.

The Free Speech Movement (FSM) Cafe deployment is a small ACme deployment for tracking energy usage in a cafe environment. Six ACmes are used to monitoring various equipments such as soup warmers, coffee grinders, receipt printer, freezer, and etc. Data is also used to identify possible savings and customer consumption patterns.

The Apartment deployment is another small ACme deployment for remote monitoring and control of apartment lighting and appliances. Six ACme's are installed on various appliances throughout the apartment, including refrigeration, TV, playstation, and lights.

Lawrence Berkeley National Lab 500 ACme



500 ACme's used in combination with other sensors, such as temperature, humidity, and flow sensors, to monitor and analyze energy usage of Building 90 in Lawrence Berkeley National Laboratory.

Cory Hall – 100 ACme



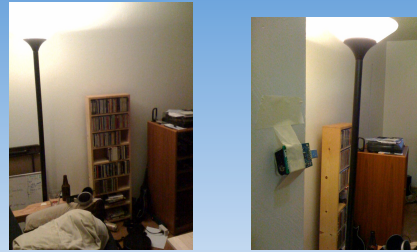
100 ACme's are used to monitor plug-loads in computer labs and offices, as part of a larger building-to-grid testbed involving various sensors and meters.

Free Speech Café – 6 ACme



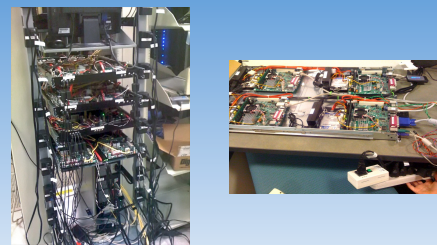
Energy usage tracking in a café environment. Six ACme's are used to monitoring various equipments such as soup warmers, coffee grinders, receipt printer, freezer, and etc. Data is also used to identify possible savings and customer consumption patterns.

Apartment – 6 ACme



Remote monitoring and control of apartment lighting and appliances. Energy information is viewable globally in real-time. A mote is used as a light switch.

Atom Cluster – 16 ACme



High-fidelity power monitoring of a 16-node Atom cluster. ACme data is used to identify major power consuming components.

Figure 10.1: ACme deployments and related research.

Because ACme is a standard IP device, and is connected to the cloud through the apartment’s Internet router, energy information is viewable globally in real-time; and appliances can be turned on or off from anywhere in the world. For convenience, a Telos mote next to the door is used as a light switch.

To study the power proportionality of modern computers, a group of graduate students uses 16 ACme’s to monitor power usage of a 16-node Atom cluster. High-fidelity ACme data is used to understand, in real time, the change in power drawn by the cluster versus the change in load on the cluster. This is used to vary quality of service offered by services on the cluster depending upon it’s energy allowance, described in more detail in [55].

10.2 Building-to-Grid

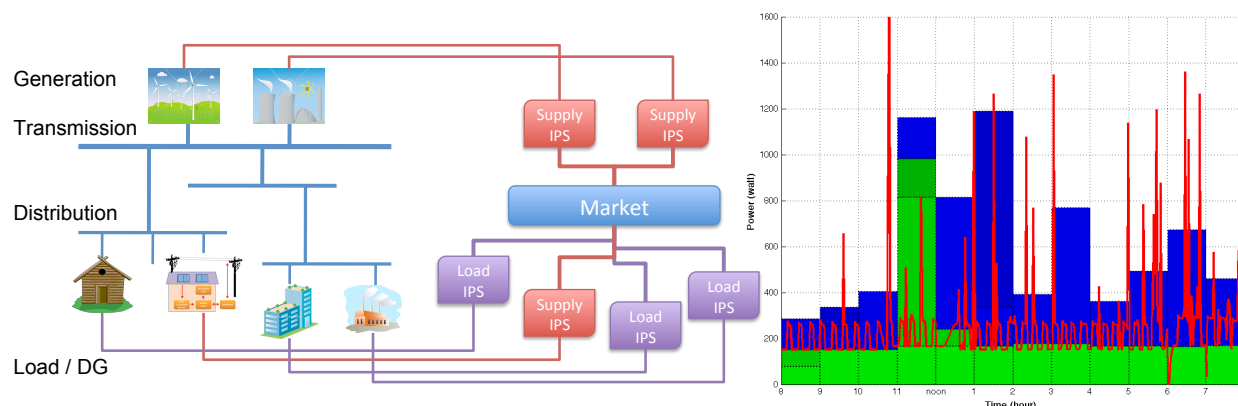


Figure 10.2: Using historical load traces and simulated supply, the LoCal simulator predicts future load and performs trades on a virtual market consisting of two types of goods: base power and variable power. The right side of the graph shows how the prediction (green is purchased base power and blue is variable power) matches the actual load (shown in red).

Extending the notion of load monitoring from within buildings to the outside, we can view buildings as large loads to the power grid. In fact, these are loads that we can optimize (load-shaping) and match to the supply side, if a real-time monitoring, prediction, and communication network were to overlay on the existing electrical grid. LoCal (A Network Architecture for Localized Electrical Energy Reduction, Generation, and Sharing) [46] is the smart-grid project at Berkeley that I have been working on that borrows principles from the Internet to the existing grid. The LoCal simulator, as seen in Figure 10.2, further demonstrates the importance of load-side monitoring and prediction on the stability and optimal operation of a future smart grid system. I believe it will be extremely meaningful to continue this work and bring buildings to the grid as large loads that can be continuously shaped and optimized, far beyond what current demand-response envisions.

10.3 Online Energy-Climate Model

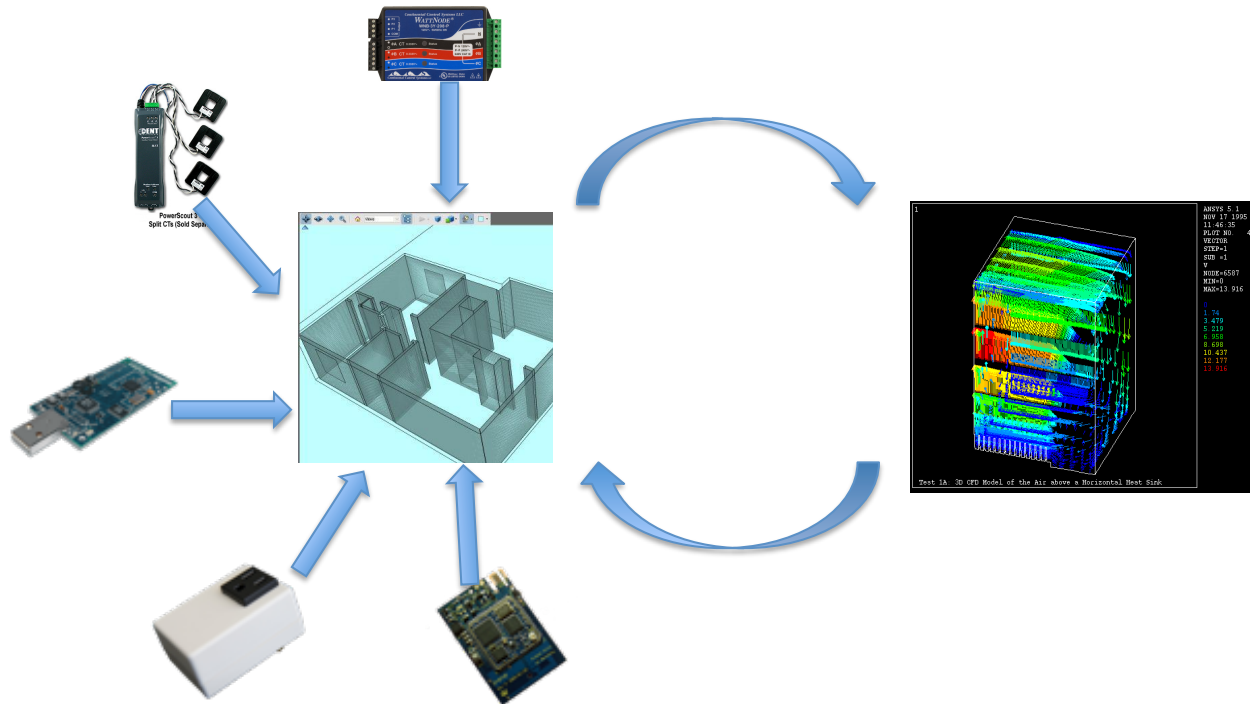


Figure 10.3: An online energy-climate model that combines real-time sensor feeds with architectural model of the building, enabling dynamic optimizing of energy and comfort.

The building energy monitoring architecture enables us to collect temporal and spatially correlated sensor data in real-time, including power, temperature, light, humidity, and etc. By combining these data with the architectural model of the building (i.e. a CAD model), we can create an energy climate model of the building, updated in real-time, as seen in Figure 10.3. Such model will have a huge impact on our ability to optimize for energy utilization and meanwhile maintain comfort. This approach will enable us to model, predict, and control the thermal and electrical densities in real-time. Current state-of-art CAD simulators allow input of very simple stimulus such as trajectory of the sun, and output the thermal densities. I hope to create a system that combines building CAD models with on-line distributed sensing, climate modeling, and prediction.

Actuation will also play an important part in completing the control loop. I envision that, in parallel to the sensing layer described in this thesis, there will be an actuation layer consisting of thermostats, ACme's (specifically the relay function), VFDs, vents, fans, controllable window shades, and various set-points that already exist in building HVAC systems. Similar to the sensing layer, this actuation layer is represented by sMAP, and is easily accessible by portable applications and services through the networking layer.

10.4 None-Intrusive Load Monitoring

A parallel but closely related research area to pervasive power monitoring is non-intrusive load monitoring (NILM). NILM describes the set of techniques that uses signal analysis to disaggregate loads from a single high-fidelity aggregate measurement. However, most if not all NILM algorithms require time-consuming training on individual loads before they are effective. Using the architecture proposed in this thesis and the vast amount of per-appliance data I have collected, I plan to build an open appliance power signature database, which will allow users to minimize the training overhead and enable more accurate load signature matching. I hope to continue this work and develop a set of NILM algorithms that can be widely used in real environments.

Chapter 11

Conclusion

Existing solutions in commercial building energy monitoring are insufficient in identifying waste or guide improvement because they only provide usage statistics in aggregate, both spatially and temporally.

By viewing the building as a system, we see that this lack of visibility into fine grain energy data is due to several factors inherent in current building infrastructures:

- Current buildings optimize for human comfort, instead of energy efficiency. They are open-loop systems where occupants, the primary load to the building, is given little or no feedback on the states of the building’s energy statistics, and therefore could not act to reduce it.
- Sensors and actuators are tightly integrated inside vertical stovepipe solutions, communicating using legacy and proprietary protocols, and without an open interface for external communication. This design prevents data to be consumed programmatically and impedes the proliferation of portable building applications.
- Preliminary efforts to feedback energy data to occupants consist of displaying simple visualizations of single-dimensional and unstructured metering data directly to the occupants. This approach is insufficient in inducing occupants to reduce usage.

To significantly and sustainably reduce energy usage in buildings, we need an architecture and a system implementation that provide high-fidelity real-time visibility into each component of the building.

In the first half of this dissertation, we propose a **three-tiered architecture** consisting of “sensing”, “data delivery and representation”, and “applications and services”. We show that this layering allows us to cleanly abstract the low-level details of the myriads of disparate monitoring instruments and protocols, provide an uniform data representation interface, and enable innovation in portable building applications.

We first show that *ACme* plug-load meter enables us to wirelessly instrument a large number of appliances continuously at high-fidelity, providing us visibility into the end-uses inside a building. In addition, the integrated IPv6 stack allows *ACme* to seamlessly interoperate with other IP devices. We further demonstrate that by using other types of sensors such as vibration and light sensors, we can increase our coverage, especially in places where direct electrical measurement is inappropriate.

The “data delivery and representation” layer functions as a narrow-waist, decoupling portable application and services from the sensing layer below. We show that sMAP – Simple Measuring and Actuation Profile, is able to provide uniform data access to a large subset of devices typically found in buildings, allowing heterogenous sensors, meters, actuators, and other data sources to intercommunicate over IP.

Building on top of “sensing” and “data delivery and representation” layers, we show that it is easy to create portable building “applications and services”. We present a simple application that collects periodic energy reports from a mixture of sensors via UDP and store them inside a database, along with associated meta data. This data is used later for creating actionable feedbacks to the occupants.

This architecture provides the facilities for observing high-fidelity energy statistics, but does not create *actionable* views of those data. Therefore, in the second part of this dissertation, we propose an **application process flow** for energy data analysis and visualization, instantiated by a real deployment, which also serves to evaluate our architecture. This process consists of three parts – we begin by understanding and instrumenting the load tree, followed by data analysis, modeling, and disaggregation of energy usage statistics; and finally, combined with meta-data, we re-aggregate individual load usages into actionable representations for visualization and feedback to the occupants.

To understand how energy is consumed and wasted, we need to first observe the energy flows inside a building. Hence the first step in this process is understanding the structure of the load tree and instrumenting it with various sensors to obtain empirical measurement. However, it is impractical to instrument all points in the load tree and obtain both full and detailed coverage. Realizing this constraint, we measure aggregate energy usage where possible, and we subsample plug-load appliances and HVAC systems to obtain measurements of representative end-use usage traces. In the second step of this process, we develop several techniques for improving fidelity of appliance models. We show that these set of techniques allow us to build representative models and disaggregate whole building energy consumptions.

Finally, we present various ways to feedback energy data to the occupants that result in energy reduction. We show various ways to re-aggregate energy usage to provide insights into how and where energy is consumed. Functional and electrical re-aggregations result in visualizations that are particularly useful for building managers to identify waste and problems; spatial and individual re-aggregations provide feedback to building occupants and incentivize saving.

We further demonstrate the potential of our architecture with several visualization ap-

plications. The ACme web portal provides a way for individuals to easily visualize electrical loads that he or she is directly using in real-time, and the relative proportions amount different types of loads. This applications helps establishing a connection between individuals working in a commercial building and the share of energy that he or she is responsible for. A load-tree visualization application shows the real-time energy flow of the load tree, as represented by the width of the edges. This view gives the building managers an visual indication of the health of the building and areas for concentrating reduction efforts. The Google PowerMeter application is a great visualization and feedback tool that can be easily integrated into user's iGoogle page and provides simple analysis and prediction on personal energy usages. Mobile phones also represent an unique opportunity for our system – it provides localization of the occupant, a screen for visualizing energy usage, and a means of feedback. We demonstrate its interaction with our system with a simple Android application that displays energy usage graphs (measured by ACme's) of appliances physically close to the cellphone, and presents an interface for actuation.

In conclusion, this work demonstrates that **high-fidelity** energy monitoring enables occupants to reduce electrical use in buildings in the following ways:

- The ACme wireless power meter enables real-time and networked energy monitoring of individual plug-loads, providing high-fidelity measurements in terms of power resolution, sampling rate, and granularity in the load tree.
- By abstracting a wide range of sensors and meters, the three-tiered architecture enables time-correlated, whole-building scale energy monitoring, providing high-fidelity usage statistics in terms of spatial and temporal coverage. In addition, the uniform data representation layer enables building applications to easily access high-fidelity energy data.
- And finally, in contrast to existing building systems where occupants are unaware of consumption, the application process flow utilizes high-fidelity real-time energy data made accessible by our architecture, to create actionable views of energy usages, which lead to significant and sustainable energy reductions by the occupants.

Bibliography

- [1] Ade7753 evaluation kit documentation. http://www.analog.com/static/imported-files/eval_boards/ADE7753_EvalBoard.pdf.
- [2] amCharts. <http://www.amcharts.com/>.
- [3] Arch Rock. <http://www.archrock.com/>.
- [4] BroadWin Technology. <http://broadwin.com/>.
- [5] Campbell Scientific, Inc. <http://www.campbellsci.com/scada>.
- [6] DOE-2. <http://www.doe2.com/>.
- [7] Google PowerMeter. <http://www.google.org/powermeter/>.
- [8] Green Box. <http://www.getgreenbox.com/>.
- [9] Green Hub. <http://www.energyhub.net/>.
- [10] Hall effect sensing and application. <http://content.honeywell.com/sensing/prodinfo/solidstate/technical/hallbook.pdf>.
- [11] Ip-based wireless sensor networking: Secure, reliable, low-power ip connectivity for ieee 82.15.4 networks. Arch Rock.
- [12] Johnson Controls. http://www.johnsoncontrols.com/publish/us/en/products/building_efficiency/building_management.html.
- [13] Kill-a-watt. <http://www.p3international.com/products/special/P4400/P4400-CE.html>.
- [14] Meraki, inc. <http://www.meraki.com>.
- [15] Microsoft Hohm. <http://www.microsoft-hohm.com/>.
- [16] Open mesh mini-router. <http://www.open-mesh.com>.

- [17] SCADA Group. <http://www.scadagroup.com/pages/>.
- [18] Schneider Electric. <http://www.schneider-electric.com/corporate/en/products-services/buildings/intelligent-buildings/building-management-system.page>.
- [19] Sentilla. <http://www.sentilla.com/>.
- [20] Technical overview of time synchronized mesh protocol (tsmp). White paper, Dust Networks.
- [21] Tendril. <http://www.tendrilinc.com/>.
- [22] TRNSYS: The Transient Energy System Simulation Tool. <http://www.trnsys.com/>.
- [23] Wirelesshart technical data sheet. Data sheet, HART Communication Foundation, 2007.
- [24] Aleka Pappas. Building Energy Modeling: An Overview of Available Software and Modeling Strategies.
- [25] Autodesk. Autodesk Ecotect Analysis. <http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=12602821>.
- [26] Autodesk. Autodesk Green Building Studio. <http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=11179508>.
- [27] M. Baranski and J. Voss. Genetic algorithm for pattern detection in nialm systems. *In Systems, Man and Cybernetics*, 2004.
- [28] Chris Beckmann, Sunny Consolvo, and Anthony Lamarca. Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. pages 107–124. 2004.
- [29] D.B. Belzer. Energy end-use flow maps for the buildings sector, 2006. http://www.pnl.gov/main/publications/external/technical_reports/PNNL-16263.pdf.
- [30] Ian D. Chakeres and Charles E. Perkins. Dynamic manet on-demand (dymo) routing (draft-ietf-manet-dymo-10), 2007.
- [31] Marshini Chetty, David Tran, and Rebecca E. Grinter. Getting to green: understanding resource consumption in the home. In Hee Yong Youn and We-Duke Cho, editors, *UbiComp*, pages 242–251. ACM, 2008.
- [32] Stephen Dawson-Haggerty. Design, implementation, and evaluation of an embedded ipv6 stack. In *Master's Thesis*, 2010.

- [33] Stephen Dawson-Haggerty, Xiaofan Jiang, Gilman Tolle, Jorge Ortiz, and David Culler. smap – a simple measurement and actuation profile for physical information. In *SenSys*, 2010.
- [34] Stephen Dawson-Haggerty, Jorge Ortiz, Xiaofan Jiang, Sushant Shankar Jeff Hsu, and David Culler. Enabling green building applications. In *The Sixth Workshop on Hot Topics in Embedded Networked Sensors (HotEmNets 2010)*, 2010.
- [35] DENT Instruments. PowerScout Press Release. http://www.dentinstruments.com/pr_power_scout_meters_monitors_launched.html.
- [36] J.K. Dobson and J.D. Griffin. Conservation effect of immediate electricity cost. feedback on residential consumption behavior. *Proceedings of the 7th ACEEE Summer Study on Energy Efficiency in Buildings*, 1992.
- [37] J.K. Dobson and J.D. Griffin. Conservation effect of immediate electricity cost. feedback on residential consumption behavior. *Proceedings of the 7th ACEEE Summer Study on Energy Efficiency in Buildings*, 1992.
- [38] Prabal Dutta, Jay Taneja, Jaemin Jeong, Xiaofan Jiang, and David Culler. A building block approach to sensornet systems. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 267–280, New York, NY, USA, 2008. ACM.
- [39] EnviProt. Auto shutdown manager, 2009. <http://www.enviprot.com/>.
- [40] L. Farinaccio and R. Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 30(3), p.245-259, 1999.
- [41] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, Irvine, California, 2000.
- [42] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. Collection tree protocol. <http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html>.
- [43] Google.org. Google PowerMeter Overview. <http://www.google.com/powermeter/about/about.html>.
- [44] G.W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 1992.
- [45] Mike He, Evan Reutzler, Xiaofan Jiang, Randy Katz, Seth Sanders, David Culler, and Ken Lutz. An architecture for local energy generation, distribution, and sharing. *IEEE Conference on Global Sustainable Energy Infrastructure (Energy2030 '08)*, 2008.

- [46] Mike M. He, Evan M. Reutzel, Xiaofan Jiang, Randy H. Katz, Seth R. Sanders, David E. Culler, and Ken Lutz. An architecture for local energy generation, distribution, and sharing. In *IEEE Energy2030 Conference Proceedings*, 2008.
- [47] Jonathan Hui. *An Extended Internet Architecture for Low-Power Wireless Networks—Design and Implementation*. PhD thesis, University of California at Berkeley, Berkeley, CA, USA, 2008.
- [48] Wireless systems for industrial automation: Process control and related applications. Draft, ISA, 10.-September 2009.
- [49] Ulla Janson. Passive houses in sweden: Experiences from design and construction phase. *Report EBD-T-08/9*, 2008.
- [50] Xiaofan Jiang, Stephen Dawson-Haggerty, and David E. Culler. smap: simple monitoring and actuation profile. In *IPSN*, pages 374–375, 2010.
- [51] Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler. Design and implementation of a high-fidelity ac metering network. In *IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 253–264, Washington, DC, USA, 2009. IEEE Computer Society.
- [52] Xiaofan Jiang, Prabal Dutta, David E. Culler, and Ion Stoica. Micro power meter for energy monitoring of wireless sensor networks at scale. In *IPSN*, pages 186–195. ACM, 2007.
- [53] Younghun Kim, Zainul M Charbiwala, Akhilesh Singhanian, Thomas Schmid, and Mani B Srivastava. Spotlight: Personal natural resource consumption profiler. *HotEmNets 2008*, 2008.
- [54] Younghun Kim, Thomas Schmid, Zainul M. Charbiwala, and Mani B. Srivastava. Viridiscopes: Design and implementation of a fine grained power monitoring system for homes. *11th international conference on Ubiquitous Computing (UbiComp 2009)*, 2009.
- [55] Andrew Krioukov, Prashanth Mohan, Sara Alspaugh, Laura Keys, David Culler, and Randy Katz. Napsac: The design and implementation of a power proportional web cluster. In *Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking*, New York, NY, USA, 2010. ACM.
- [56] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power & Energy Magazine*, 2003.
- [57] K. D. Lee. Electric load information system based on non-intrusive power monitoring. *PhD Thesis, MIT*, 2003.

- [58] S. B. Leeb, S. R. Shaw, and J. R. Kirtley Jr. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *IEEE Trans. on Power Delivery*, 10(3):1200-1210, 1995.
- [59] Joshua Lifton, Mark Feldmeier, Yasuhiro Ono, Cameron Lewis, and Joseph A. Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, 2007.
- [60] Yun Mao, Feng Wang, Lili Qiu, Simon Lam, and Jonathan Smith. S4: Small state and small stretch routing protocol implementation. <http://www.cs.utexas.edu/~lili/projects/s4.htm>.
- [61] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of ipv6 packets over ieee 802.15.4 networks, 2007.
- [62] Razvan Musaloiu-E. and Andreas Terzis. Minimising the effect of wifi interference in 802.15.4 wireless sensor networks. *Int. J. Sen. Netw.*, 3(1):43-54, 2007.
- [63] National Renewable Energy Laboratory. Energy10. <http://www.nrel.gov/buildings/energy10.html>.
- [64] Pacific Gas and Electric. Demand Response Incentives. <http://www.pge.com/demandresponse/>.
- [65] Danny Parker, David Hoak, and Jamie Cummings. Pilot evaluation of energy savings from residential energy demand feedback devices. In *Contract Report FSEC-CR-1742-08*. ACM, 2008.
- [66] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. *9th international conference on Ubiquitous Computing (UbiComp 2007)*, 2007.
- [67] Charles E. Perkins and Elizabeth M. Belding-Royer. Ad-hoc on-demand distance vector routing. In *WMCSA*, pages 90-100, 1999.
- [68] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 48, Piscataway, NJ, USA, 2005. IEEE Press.
- [69] A. Prudenzi. A neuron nets based procedure for identifying domestic appliances pattern-of-use from energy recordings at meter panel. In *Power Engineering Society Winter Meeting*, 2002.

- [70] *Smart Energy Profile 2.0*. Zigbee Alliance, 2010.
- [71] Thanos Stathopoulos, Lewis Girod, John Heideman, Deborah Estrin, and Karen Weeks. Centralized routing for resource-constrained wireless sensor networks (sys 5), 2006.
- [72] W. Swan. The language of bacnet. *Engineered Systems. Vol. 13, No. 7, pp. 24-32*, 1996.
- [73] Tektronix. SDH Telecommunications Standard Primer. http://www.tek.com/Masurement/App_Notes/sdhprimer/2RX_11694_2.pdf.
- [74] Tektronix. SONET Telecommunications Standard Primer. http://www.tek.com/Masurement/App_Notes/SONET/.
- [75] The Modbus Organization. Modbus Application Protocol Specification. <http://www.modbus.org/specs.php>.
- [76] U.S. Department of Energy. EnergyPlus. <http://apps1.eere.energy.gov/buildings/energyplus/>.
- [77] U.S. DOE Energy Information Administration. Commercial buildings energy consumption survey, 2003. <http://www.eia.doe.gov/emeu/cbecs/>.
- [78] U.S. DOE Energy Information Administration. Buildings energy data book, 2008. <http://buildingsdatabook.eere.energy.gov/>.
- [79] Kamin Whitehouse, Chris Karlof, Alec Woo, Xiaofan Jiang, and David E. Culler. The effects of ranging noise on multihop localization: an empirical study. In *IPSN*, pages 73–80, 2005.
- [80] Wikipedia. Building management system — Wikipedia, the free encyclopedia, 2010. [Online; accessed 9-August-2010].
- [81] Wikipedia. Sankey diagram, 2010. [Online; accessed 9-August-2010].
- [82] Wikipedia. Scada – Wikipedia, the free encyclopedia, 2010. [Online; accessed 9-August-2010].
- [83] *WirelessHART*. HART Communication Foundation, 2009.
- [84] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27, New York, NY, USA, 2003. ACM.