Design and Implementation of a High-Fidelity AC Metering Network

Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler
Computer Science Division
University of California, Berkeley
{fxjiang, stevedh, prabal, culler}@cs.berkeley.edu

ABSTRACT

We present the architecture, design, and preliminary evaluation of *ACme*, a wireless sensor and actuator network for monitoring AC energy usage and controlling AC devices in a large and diverse building environment. The ACme system consists of three tiers: the ACme node which provides a metering and control interface to a single outlet, a network fabric which allows this interface to be exported to arbitrary IP endpoints, and application software that uses this networked interface to provide various power-centric applications. The ACme node integrates an Epic core module with a dedicated energy metering IC to provide real, reactive, and apparent power measurements, with optional control of an attached load. The network comprises a complete IPv6/6LoWPAN stack on every node and an edge router that connects to other IP networks. The application tier receives and stores readings in a database and uses a web server for visualization. Nodes automatically join the IPv6 subnet after being plugged in, and begin interactions with the application layer. We evaluate our system in a preliminary green building deployment with 49 nodes spread over several floors of a Computer Science Building and present energy consumption data from this preliminary deployment.

Categories and Subject Descriptors

B.0 [**Hardware**]: General; B.4 [**Hardware**]: Input/Output & Data Communication; J.4 [**Computer Applications**]: Social and Behavior Sciences

IPSN'09, April 15–18, 2009, San Francisco, California, USA.

General Terms

Design, Experimentation, Performance, Measurement

Keywords

Architecture, AC, Meter, Green, Energy, Power, Platform, Wireless, Sensor Network, Mote, Measurement

1. INTRODUCTION

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 this paper we focus on a large scale wireless sensor network for a very different kind of application–high fidelity, building-wide, electricity monitoring.

Electricity usage in residential and commercial buildings represent a significant fraction of total energy expenditure and, since electricity is generated from fossil fuels, it represents a significant portion of the carbon footprint of the occupant. The typical Pacific Gas & Electric (PG&E) customer uses about 540kWh of electricity per month, while the Computer Science Division building at the authors' institution consumes approximately 12MWh of electricity per day. This usage is divided between HVAC, lighting, plug-loads, and servers. Preliminary data show that the plug-load usage represents approximately 20%, or 2.4MWh, of the total usage and that much of this is 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. Our 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 adapt 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 [15] 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. This application is one the first deployments of a new, open source IPv6 network stack over 802.15.4 using 6LoWPAN. All of the routing on the motes is true IP routing and a collection of modified Meraki Mini nodes route between the 6LoWPAN subnet and the various other subnets. This enables a distribution, deployment, management, and data analysis model that is essentially that of a distributed collection of hosts: some clients, some servers, some both.

To evaluate our design, we apply the technology to the problem of excessive energy usage in our Computer Science Building. In this project, 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. Section 6 presents noteworthy results from this deployment.

2. SYSTEM ARCHITECTURE

The overall design of the ACme monitoring and control system is shown in Figure 1. The design decomposition is three tiers: the end nodes, the network, and the server-hosted, application-specific code.

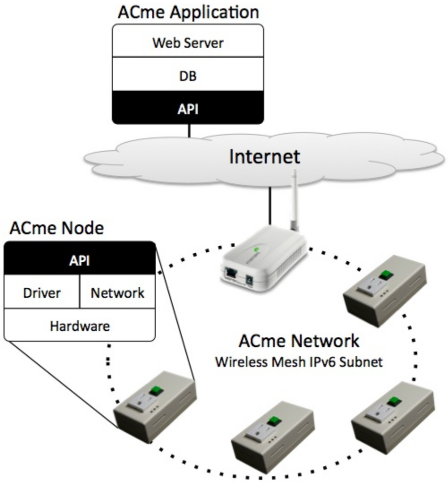


Figure 1: Three tier ACme system.

The ACme node supports a set of sampling and control operations such as `read_energy()`, `switch(state)`, and `report(ip_addr, rate)`. The ACme network is a wireless IPv6 network that uses a small dual-interface Linux device to route packets between the ACme nodes and other IP networks. The application tier is an energy application that interacts with the network of ACme nodes via its API.

In the following sections, we present the design and implementation of each tier of the ACme system. At the node level, we discuss various tradeoffs present in the hardware design, and present a narrow interface to access the hardware and we consider physical design issues such as form factor and thermal dissipation, which must be addressed for a deployment in an office setting to be practical. The result is the ACme meter: an integrated, small form-factor device with energy metering, control, and networking in one package.

A set of disconnected meters is much less interesting than real-time data. Traditional monitoring designs use sneakernet and a small LCD, a serial port, or other wired ports connecting instruments to data loggers. A key motivation of our work is to allow the quick deployment and instrumentation of a large numbers of AC plug meters and enable continuous real-time access. Requiring connectivity over USB, serial, Ethernet, or other wired channel would not be practical. Also, although most places we wish to measure have substantial 802.11 infrastructure installed, the reality is that it is not easy to add additional devices to these networks. Thus, our solution is the development of an ad-hoc IEEE 802.15.4 network layer that provides IP connectivity to ACme nodes without the use of either wiring or infrastructure. The network provides connectivity between the sensor nodes and other networks using a dual-interface router.

Finally, application specific code can be placed on any network connected to the ACme meter subnet. Applications use the node API as exported over the network. Typically, a server daemon populates a database that is accessed by a variety of web applications, but direct access to ACme including telnet and a UDP RPC is available as well. We present a green building power profiling application which allows users to view their individual real-time energy consumption using a web interface. It consists of a logger daemon which formats and inserts the reported data into a database, and a presentation layer which synthesizes reports from this logged data. A key architectural separation is that applications are developed independently from the network used.

3. ACME NODE

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 2(a).

The current-to-voltage converter converts the current used by the AC appliance to voltage, as described in Section 3.1. 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 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 2(b) and 2(c). We compare the two designs and evaluate their effectiveness for our application in Section 3.4.

3.1 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.

3.1.1 Shunt Resistor

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 2(b). 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.

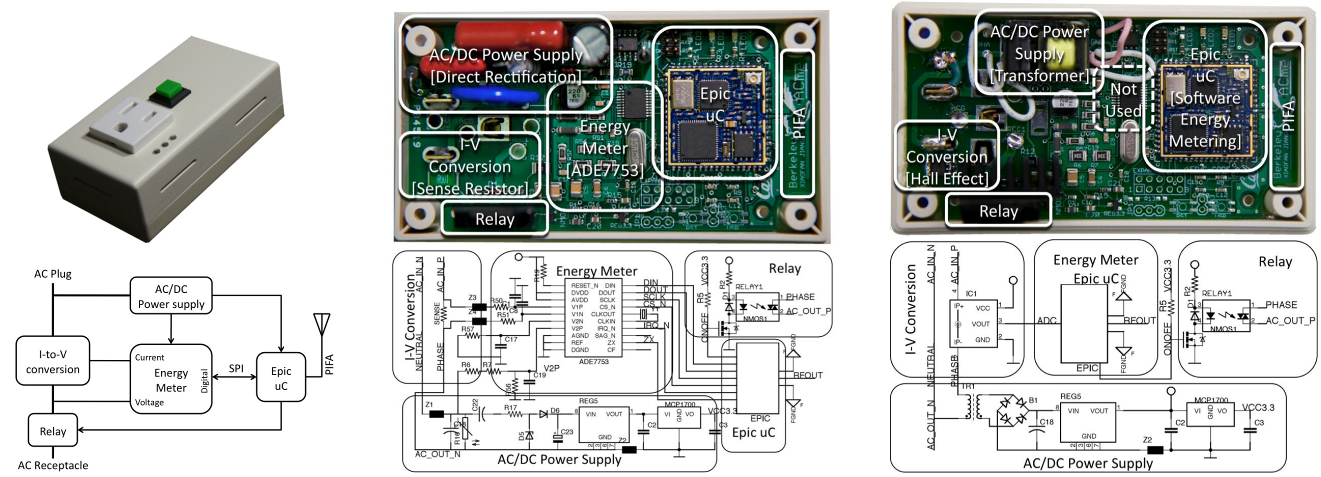
3.1.2 Current Transformer

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



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

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

(c) 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.

Figure 2: ACme architecture and simplified schematics.

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.

3.1.3 Hall Effect Sensor

A third method of converting current to voltage is the Hall Effect sensor. These devices use the Hall Effect [4] 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 2(c).

3.2 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 method, 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 2(b)) uses this chip.

3.3 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. However, this does not allow a shunt resistor to be used for the I-V conversion, due to a mismatch in ground level. Instead, we need to either use a current transformer, which is too large for our application, or an in-line Hall Effect sensor, which represents the design of ACme-B.

3.4 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 1: Properties of ACme-A and ACme-B.

ACme-A, shown in Figure 2(b) 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 2(c) uses the combination of in-line Hall Effect sensor, transformer-based power supply, and software-based energy metering. Table 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.

3.5 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).

3.6 Microcontroller and Radio

ACme uses the Epic Core [15] 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 connected 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.

3.7 Mechanical and Thermal

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.

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.

3.8 Embedded Software

The ACme node software provides a simple interface over the network, shown in Table 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.

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

Table 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.

3.9 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 3, ACme 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. ACme can be easily adopted into home and offices due 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 6.

4. ACME NETWORK

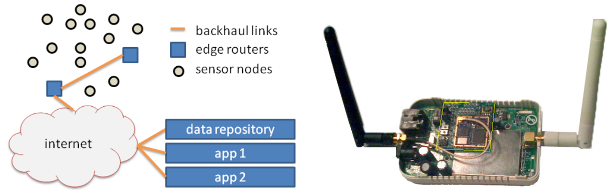
Many different approaches to networking a subnet of low-powered devices been presented in the literature, and some of the most relevant ones are evaluated in Section 8. However, at the start of the project, no available solution met our needs, and so we drew inspiration from ongoing IETF work to adapt IPv6 for low-power links. We have developed `blip`, a new open-source IPv6 implementation in TinyOS.

Measurements	real, reactive, apparent
Resolution	40mW
Sampling speed	14kHz
Report speed	2.8kHz
Maximum power	1800W
Energy accumulation	6.26min
Radio range	multiple floors
Idle power	1W
Size	10x5.6x4cm

Table 3: Summary of ACme

Key concepts necessary for bringing IPv6 to wireless embedded systems were explored in [19], 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 [25], and shows how IPv6 mechanisms like neighbor discover 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 address the specifics of transport and routing which are implementation-specific and critical for good performance.

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.



(a) Sensor nodes are connected to the Internet through IP routers. ACme nodes are the sensors and edge routers are Linux-class devices.

(b) 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.

Figure 3: Network architecture and edge router.

4.1 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. The network protocol improves forwarding reliability by using this set of default route choices to provide a small set of possible next-hops, which it sends to sequentially. The forwarding engine also employs link-layer retransmission for each unicast packet.

To enable communication with individual devices, each sensor node observes a subset of the local radio link connectivity and intermittently reports it to an edge router using topology update messages. These messages are either sent in dedicated messages, if the network is quiescent, or piggybacked on outgoing data messages as an IPv6 destination option header. From the union of reported topology, edge routers form paths back into the network. Packets destined into the network are source routed from the edge router to a particular node. This basic algorithm satisfies the requirement both for reliability and point-to-point communication. It provides reliable routing to the edge by maintaining multiple candidate next hops in the direction of an edge router, while allowing point-to-point communication by sending traffic to the edge, where it is re-routed. This solution is similar to that presented in [19].

4.2 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 [7, 8]. 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. To improve routing robustness within the subnet, all edge routers join an IPv6 link-local multicast group and forward topology updates to the group; consistency is maintained by periodic retransmissions.

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.

4.3 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 since 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 repair methods we employ of multiple next-hops and retransmission. When developing the protocol, we found that most datagram drops were 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.

4.4 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 four months. Figure 6(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 is 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 [26], we observe a relatively modest decline in yield, although our results are not necessarily comparable due to different environments.

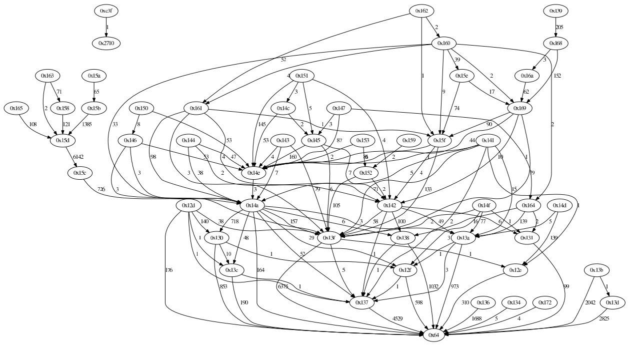


Figure 4: A snapshot of the network topology.

5. ACME APPLICATION TIER

The ACme application consists of a web front-end, a database back-end, and a daemon process. The application is built atop the network of ACme nodes and is logically distinct from both the nodes and the network. This architecture allows many distinct applications to use the underlying network concurrently. From the application's perspective, each ACme exposes operations to configure the node and report its energy measurements over UDP using a simple binary data format. 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.

ACme nodes are typically configured to report energy readings once per minute via UDP to a simple Python application daemon running on a 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. Users access historical energy data through a web interface that queries the database and presents the results in tabular and graphical form. Although most queries access archival data, the system also allows users to control plug-loads registered to them by switching power to them (this function is not exposed in the user interface). In this case, the web application can directly send a command to the node.

This application architecture marks a departure from interacting with the sensor *network* in the aggregate 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.

Since this architecture eliminates the application-layer gateway and the proxy role the gateway plays in the process, the architecture raises new concerns. Allowing unbridled IP network access to sensor nodes raises many security challenges, and they are likely to be more pressing since the nodes are resource-constrained. Techniques like NATs, firewalls, and policy-aware switching already exist to solve these problems in existing enterprise networks and they may also be applicable here.

6. GREEN BUILDING DEPLOYMENT

Motivated by research that real-time, per-appliance electricity usage feedback can induce behavioral changes that lead to 10% to 20% reductions in usage [14], we 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.

6.1 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 5 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.

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

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.

6.2 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 felt 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 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.

6.3 Preliminary Results

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 6(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 6(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 6(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.

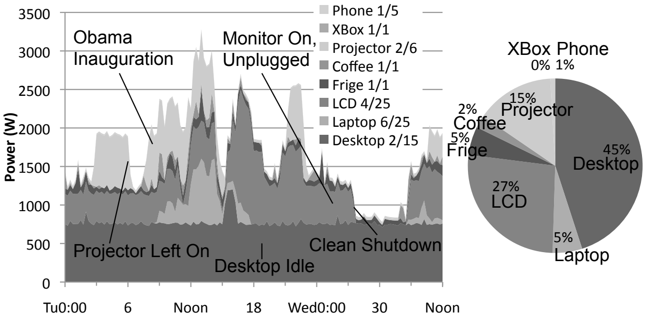
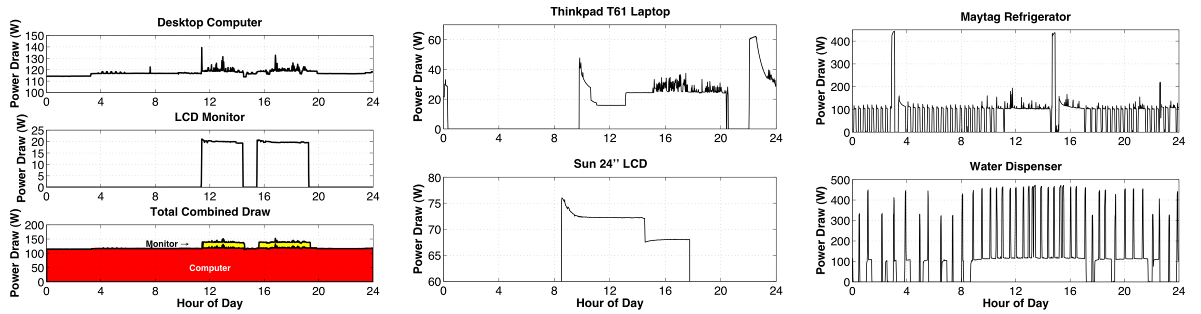


Figure 7: Aggregated energy usage by appliance type normalized by number of appliances in an enclosed environment. The legend indicates "*Appliance_type #metered/#count*."

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 6(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 6(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.

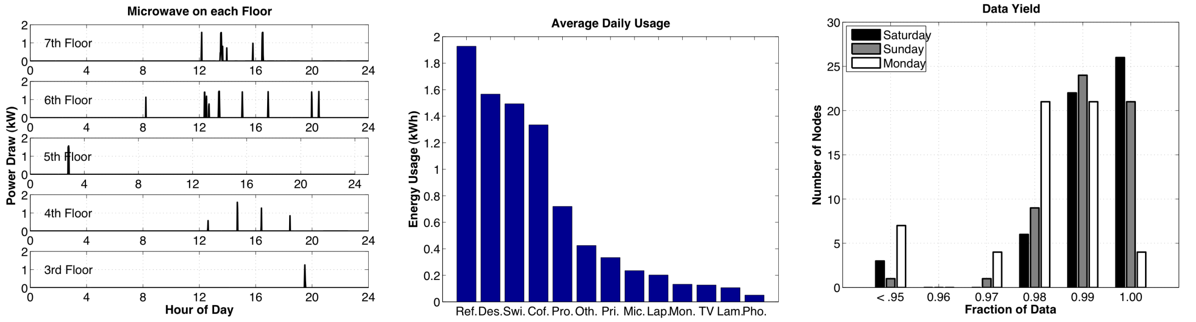
This view allows us to build models of energy usage that can be used to estimate the contribution of the various small and large loads based on the number of such loads and the average energy usage of their load class. It also focuses attention on the things that matter (in this case, the desktops and network switches), since they are heaviest consumers and they are numerous. Figure 7 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



(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 (top figure) floors on a typical day.

(e) Energy usage by plug-load type on a Monday averaged across all loads of the particular type.

(f) Data yield from 49 nodes over a three day period.

Figure 6: Power traces from various appliances under measurement.

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.

7. DISCUSSION AND FUTURE WORK

The development of ACme presented several interesting challenges. However, there are a number of equally important questions which we are now in a position to investigate using the machinery developed in this work. In this section, we discuss new research directions using the ACme platform, as well as several unresolved issues.

7.1 Discussion

We have learned many lessons through the design and implementation of ACme, which we now discuss.

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!

We have two deployments, one in the Computer Science building and the other in an apartment; both have been streaming data continuously to a single database for the past four month. This allows us to first verify changes in a small controlled environment before transferring them to the larger setting. These two deployments have produced a significant amount of data; the *energy* table alone has more than eight million rows. As the table fills, SQL queries become slower, which forced us redesign our database schema about two months into the deployment to be more modular and easier for data-mining. This did not interrupt the data collection since we simply inserted data into both databases during the transition period. However, queries that aggregate over appliance classes and time take several seconds to perform, suggesting that more attention should be paid to database and query optimization. In a future with tens of thousands or more ACme devices, we may need to investigate more scalable data storage mechanisms.

While ACme is relatively small, it still occupies significant power strip space. A simple modification is to add a multiplexer at the front-end to enable multiple input channels, or fold the entire ACme into a power strip. This decreases the cost per monitored appliance and saves space. Finally, our experience has shown the text based shell to be a convenient way to configure and debug our network of meters.

7.2 Enabled Research

Looking beyond the Green Building deployment, we consider how ACme could be used in service of future research goals. Many of these goals include human factors and sociological elements which we do not address but would be enabled by ACme.

Personal Accountability: The "Green Soda" application we have presented makes 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 [14]. 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 were 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 [18].

8. RELATED WORK

Virtually all households in the United States have electric meters installed. Unfortunately, the data they provide is not very useful for either identifying per-appliance energy consumption or generating real-time feedback of usage because of the low sample rates and lack of network connections. Work in this area has resulted in "smart meters," which replace traditional analog meters and are able to report usage data to a utility at a much finer temporal granularity then previously available. However, these meters do not extend within residences to provide the fine-grain view we provide.

With the recent increase of awareness in energy conservation, several commercial products have appeared on the market which measure single outlet energy consumption, commonly referred to as plug-load meters [6]. These are helpful for point measurements, but still require manual measurement. Such an approach does not address the deployment challenges at scale, which our solution addresses directly.

Several startups, such as Tendril [10], GreenBox [2], and EnergyHub [3], have recently announced proprietary wireless monitoring solutions that are not yet publicly available. Their technology is similar to ours, and provides an interesting parallel in the closed space.

A few sensor network systems have explored AC metering including MIT's "Plug" platform [23]. "Plug" provides apparent power measurements through a current transformer and uses an ADC for direct sampling. This design choice was discussed in this project but not used due to its size, lack of real and reactive power, and lack of energy accumulation. Microsoft Research's "smart-socket" [28] uses a similar design as ACme but is battery-powered. Their web services approach interoperates with other sensors and control points, and their results shows energy savings in a home context, but their focus is on application composition.

High fidelity power measurement is also very important. Several studies have looked at using high fidelity power traces to identify appliances and their states [20, 21, 22]. High fidelity data can also be used for fault detection [12], power grid health monitoring and fault forecasting [17].

Future experimentation with demand-response control loops will require communication with individual devices and so collection-only protocols like MultiHopLQI or CTP [30, 16] were not an option. CentRoute [29] provides communication with single routers in the network, but we were concerned 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 for this class of networks due to high control overheads or large routing state [27, 13]. While S4 appeared promising, actual use of it would require the design and implementation of a location service [24]. Proprietary stacks such as TSMP, WirelessHART, and PhyNet [9, 11, 5] also appear worthwhile but are not widely available.

9. CONCLUSION

This paper shows that wireless sensor network technology has a great deal to offer energy monitoring, management, and control applications. Conversely, energy applications drive the sensor network research agenda in new directions with different requirements and constraints. Power, for example, while no longer being a constraint, is still very intertwined with the design. The network architecture and protocols are still limited by low bandwidths and small memories, but are now required to provide support for different traffic patterns than those used in traditional sensor networks. Finally, while radio connectivity is still an important part of design, the focus switches from designing extremely low duty-cycle media access protocols to coping with the crowded environment found indoors.

The natural evolution of this work is to move metering from a separate device into the appliance. Cost is an important constraint on the types of devices which will find favor, and for this reason the metering devices will still have very inexpensive microcontrollers and low-power radios, although appliances may export more functionality than is present in our design. Critically, power applications will require a general purpose network rather than specialized protocols and application gateways. Future research may integrate programmability into the end devices to allow them to be tasked with more elaborate data processing and generation tasks than our current model allows.

The source code, harware designs, and data are available at `http://acme.cs.berkeley.edu/`.

10. ACKNOWLEDGEMENT

Special thanks to Peter Corke, Koen Langendoen, and Albert Goto for their help. This material is based upon work supported by the National Science Foundation under grants #0435454 ("NeTS-NR") and #0454432 ("CNS-CRI"), an NSF Graduate Fellowship, a Microsoft Graduate Fellowship, and generous gifts from HP and Intel.

11. REFERENCES

[1] Ade7753 evaluation kit documentation. `http://www.analog.com/static/imported-files/eval\_boards/ADE7753_EvalBoard.pdf`.

[2] Green box. `http://www.getgreenbox.com/`.

[3] Green hub. `http://www.energyhub.net/`.

[4] Hall effect sensing and application. `http://content.honeywell.com/sensing/prodinfo/solidstate/technical/hallbook.pdf`.

[5] Ip-based wireless sensor networking: Secure, reliable, low-power ip connectivity for ieee 82.15.4 networks. Arch Rock.

[6] Kill-a-watt. `http://www.p3international.com/products/special/P4400/P4400-CE.html`.

[7] Meraki, inc. `http://www.meraki.com`.

[8] Open mesh mini-router. `http://www.open-mesh.com`.

[9] Technical overview of time synchronized mesh protocol (tsmp). White paper, Dust Networks.

[10] Tendril. `http://www.tendrilinc.com/`.

[11] Wirelesshart technical data sheet. Data sheet, HART Communication Foundation, 2007.

[12] M. E. H. Benbouzid. A review of intruction motors signature analysis as a medium for faults detection. *IEEE Transactions on Industrial Electronics, 47(5)*, 2000.

[13] I. D. Chakeres and C. E. Perkins. Dynamic manet on-demand (dymo) routing (draft-ietf-manet-dymo-10), 2007.

[14] J. Dobson and J. 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.

[15] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler. A building block approach to sensornet systems. In *Conference on Embedded Networked Sensor Systems (SenSys '08)*, 2008.

[16] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis. Collection tree protocol. `http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html`.

[17] P. Ge, Q. Guo, H. Zhou, X. Li, W. Xu, G. Tang, and Q. Xu. Wide area power grid health state diagnosis and early warning system based on fault power flow fingerprint identification and mas technology. *DRPT2008*, 2008.

[18] M. He, E. Reutzel, X. Jiang, R. Katz, S. Sanders, D. Culler, and K. Lutz. An architecture for local energy generation, distribution, and sharing. *IEEE Conference on Global Sustainable Energy Infrastructure (Energy2030 '08)*, 2008.

[19] J. Hui. *An Extended Internet Architecture for Low-Power Wireless Networks– Design and Implementation*. PhD thesis, University of California at Berkeley, Berkeley, CA, USA, 2008.

[20] M. Ito, R. Uda, S. Ichimura, K. Tago, T. Hoshi, and Y. Matsushita. A method of appliance detection based on features of power waveform. *Applications and the Internet, IEEE/IPSJ International Symposium on*, 0:291, 2004.

[21] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power & Energy Magazine*, 2003.

[22] W. Lee, G. Fung, H. Lam, F. Chan, and M. Lucente. Exploration on load signatures. *International Conference on Electrical Engineering (ICEE)*, 2004.

[23] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. 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.

[24] Y. Mao, F. Wang, L. Qiu, S. Lam, and J. Smith. S4: Small state and small stretch routing protocol implementation.

[25] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of ipv6 packets over ieee 802.15.4 networks, 2007.

[26] R. Musaloiu-E. and A. Terzis. Minimising the effect of wifi interference in 802.15.4 wireless sensor networks. *Int. J. Sen. Netw.*, 3(1):43–54, 2007.

[27] C. E. Perkins and E. M. Belding-Royer. Ad-hoc on-demand distance vector routing. In *WMCSA*, pages 90–100, 1999.

[28] B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao. Tiny web services: Design and implementation on interoperable and evolvable sensor networks. In *ACM/IEEE Conference on Embedded Networked Sensor Systems (SenSys '08)*, 2008.

[29] T. Stathopoulos, L. Girod, J. Heideman, D. Estrin, and K. Weeks. Centralized routing for resource-constrained wireless sensor networks (sys 5), 2006.

[30] A. Woo, T. Tong, and D. 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.