@scale: Insights from a Large, Long-Lived Appliance Energy WSN

Stephen Dawson-Haggerty†, Steven Lanzisera‡, Jay Taneja†, Richard Brown‡, and David Culler†
†Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
‡Environmental Energy Technologies Division
Lawrence Berkeley National Lab
Berkeley, CA

Abstract

We present insights obtained from conducting a year-long, 455 meter deployment of wireless plug-load electric meters in a large commercial building. We develop a stratified sampling methodology for surveying the energy use of Miscellaneous Electric Loads (MELs) in commercial buildings, and apply it to our study building. Over the deployment period, we collected over nine hundred million individual readings. Among our findings, we document the need for a dynamic, scalable IPv6 routing protocol which supports point-to-point routing and multiple points of egress. Although the meters are static physically, we find that the set of links they use is dynamic; not using such a dynamic set results in paths that are twice as long. Finally, we conduct a detailed survey of the accuracy possible with inexpensive AC metering hardware. Based on a 21-point automated calibration of a population of 500 devices, we find that it is possible to produce nearly utility-grade metering data.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; J.7 [**Computers in Other Systems**]: Industrial Control

General Terms

Design, Measurement, Performance

Keywords

Energy, Audit, Building, Power, Wireless, Sensor Network

1. INTRODUCTION

More than a decade after wireless sensor networks emerged as a topic of research, much progress has been made in understanding the contours of the field. There is now a large body of work from which to draw inspiration and ideas. From the early "let chaos reign" days to the current state-of-the-art, the field has evolved to the point where it is possible to deploy large-scale applications over a long period with the expectation that they will work to produce useful, scientifically-relevant data. Deploying these systems at scale requires rigorous attention to both engineering and deployment management. Recent studies have begun to show successes and failures [4, 11, 17] as well as offering practical guidance on deployments, continuing the sequence of papers performing science in the real world [31, 35].

In this paper, we present results and insights from a massive application, developed and deployed over the past two years. This application consists of 455 wireless energy plug-load meters and 7 load-balancing routers deployed across four floors of a commercial building for the past year. It was motivated by a need for a better understanding of the power consumption and usage patterns of electric plug-loads, or "miscellaneous electric loads." These are estimated to make up nearly 30% of the electric load in commercial buildings [32], but are difficult to study because they are so numerous and diverse. They are a good target for a wireless sensor network because they require both high density and a large number of metering elements.

When conducting our deployment, we took care to heed the lessons of the past, and thus avoided many of the pitfalls of previous deployments. By considering both the innovative aspects of our system (the density and scale) while limiting our innovation in other areas, we have been able to develop a system which performed well and met our science goals, while developing new insights in the types of applications and practices which work well at scale.

One set of insights relates to the networking technology used. Our networking stack uses 6loWPAN/IPv6 to form all meters into a single subnet, the largest such deployment we are aware of. This results in a variety of lessons about the **routing requirements** needed for this scale, validating the need for point-to-point routing and support for multiple egress routers. We discuss **data loss** at both the routing layer and the application layer, and distinguish them where possible, as well as issues which arise when a low-power subnet is connected to other networks. Finally, even though there is little node mobility in this deployment, the usage model implies that nodes intermittently disappear, resulting in interesting **network dynamics** that emerge. We find that the set of intermittent links even in a static deployment is large, many times the size of the set of "good" links. Ignoring links that disappear due to changing noise conditions results in path lengths twice as long as using routes that incorporate variable links.

We also take to a new level inexpensive energy metering technology similar to that used in the past. The science

IPSN'12, April 16–20, 2012, Beijing, China.

goals of the deployment require consideration of the **accuracy** required by the meters; we developed an extensive automated calibration procedure that could be efficiently applied to hundreds of meters to achieve better than 2% accuracy. Furthermore, previous studies have not explicitly considered the importance of **sampling methodology** in choosing which plug-loads to monitor; we propose that *staged stratified sampling* is the proper methodology for observing usage patterns and power states across a broad range of devices.

In the remainder of the paper we first present an overview of the science goals and methodology of our study. We then present our system design, noting where we have learned from mistakes published in the past and pointing out how simplicity is key to conducting large deployments with limited resources. In the body of the paper, we examine the lessons learned by deploying our large system, and how being driven by energy science (in addition to computer science) goals led to new insights.

2. MISCELLANEOUS ELECTRIC LOADS

Auditors of commercial buildings and residences typically attribute energy to end uses such as heating, cooling, hot water, and lighting. All other energy use is attributed to the "miscellaneous" category (MELs). As major categories of electric usage are reduced through more stringent efficiency requirements and better design practice, this category of load becomes more of the overall energy spend and thus a larger target for reduction. However, MELs are both difficult to study and difficult to reduce, because they comprise a large number of relatively small loads, many of which are infrequently used. Our goal was to conduct a representative sampling of the energy used by this category of devices to build usage models of many different types of devices, as well as energy models for particular devices.

To do so, we conducted a large-scale sensor network deployment in a typical commercial office. The study building is a 1960s-era facility largely used as a traditional office space. It has a total floor area of 89,500 square feet, with approximately 450 occupants in six working groups located among four floors and a basement. Certain aspects of the study presented challenges which are not immediately apparent; for instance, although the study building was located only a few miles from our campus, due to human subjects regulations we were not supposed to enter the building to diagnose particular devices. Physical contact with individual meters was restricted to our on-site partners. This limited our opportunities for diagnosing or fixing problems with the deployment.

2.1 MELs Device Inventory

The first phase of the study consisted of a full inventory of the MELs devices in the office to serve as ground truth when comparing various sampling approaches. Due to the diversity of devices, a standardized system of identifying and recording MELs is essential for inventory and energy data analysis. We updated an extensive taxonomy developed in [23] to include new device types found today, such as tablet computers. The taxonomy consists of three levels: End Use, Category, and Product Type. MELs are divided into three major end uses – Electronics, Miscellaneous, and Traditional. Each end use is in turn composed of categories, and each category contains many product types. For example, an "LCD computer display" is a product type in the "Display" category, which is part of the end use "Electronics." The inventory categorized every plug-load in the building according to this taxonomy, resulting in the identification of almost 5,000 devices. The inventory was collected by two-person teams and recorded in a relational database.

2.2 Device Sampling Methodology

With such a large number of MELs in our study building, metering all devices would be time- and cost-prohibitive, and not all data generated would provide useful insights. The sampling must be driven by science criteria, not networking expedience. The selection of an appropriate sampling method is driven by the multi-fold purpose of our energy data collection and analysis:

- Provide a statistically-relevant survey of power and energy measurements of the population of MELs devices in a typical commercial setting; and
- Capture traces and derive usage patterns of MELs to build an appliance energy signature database and construct models for future analysis; and
- Study usage correlations between devices, *e.g.*, computer, display, and lighting within the same occupant's office.

We developed a multi-stage, stratified random sampling approach to select devices for metering. Devices were first divided into stages by physical location or organization owning the devices. For each stage, a subset of devices were then selected from a stratified sample by Device Category to meet our data collection objectives. A stratified sample is critical because a simple random sample would result in metering a large number of uninteresting devices (*e.g.*, computer speakers, external disk drives) instead of devices with significant energy use such as computers or LCD displays.

In the second phase of the study, we deployed a total of 455 meters on the selected devices. The deployment took approximately 120 person-hours. No effort was made to deploy meters to ensure network connectivity, but load-balancing routers (LBRs) were placed with connectivity and hop count in mind. Once the meters were in place, we had only limited opportunities to perform on-site troubleshooting. This made our remote debugging setup more important.

Figure 2.2 shows meter locations from the third floor of the deployment; a couple features are evident. First, the deployment was physically dense, with several meters often placed within a single office. Second, the meters are well distributed spatially. This was intentional, since the stratified sampling procedure was performed within each physical area and administrative unit.

3. SYSTEM DESIGN

We developed the metering system for this deployment using a mix of existing and custom software, combined with a custom hardware platform. Figure 2 shows a schematic of the overall metering system design, with particular emphasis on the networking. Overall, the system can be decomposed into three tiers: the metering tier, the backhaul tier, and the database.

The **metering tier** is made up of a large number of low-cost electric meters, each designed around a custom hardware platform similar to the ACme [14]. They contain an

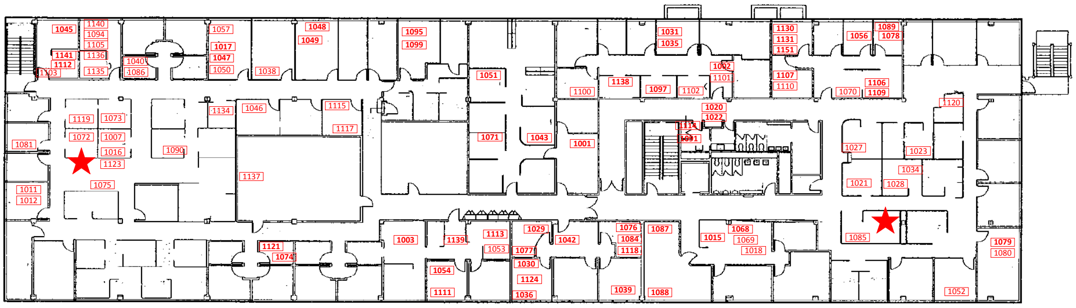


Figure 1: The third floor of the deployment. Small boxes are meters, while the two stars are LBRs.

msp430 microcontroller integrated with an 802.15.4 radio and Analog Devices ADE7753 energy metering chip. Each device runs the TinyOS operating system and uses `blip`, an IPv6/6loWPAN stack to provide IPv6 network connectivity [7].

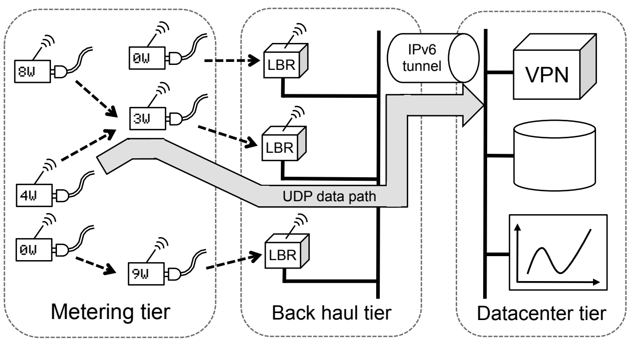


Figure 2: System design. The metering tier, left, forms an IPv6 mesh network of 802.15.4 links and transmits metering data through the backhaul tier comprised of load-balancing routers, center; these routers communicate with each other using the building's Ethernet. The UDP data packets travel over an IPv6 tunnel from the building to the Internet where they reach the database, located in a datacenter.

To provide scalability to hundreds of nodes, the back haul tier consists of a number of LBRs that provide connectivity to and from the metering elements. Unlike a deployment with multiple gateways, *these LBRs are transparent to the other tiers*. Additional units can be deployed without any extra configuration. This is a key result of our use of the IP architecture. Along with the meters, these devices make up a single IPv6 subnet where all devices participate in the routing protocol called HYDRO [8], designed to provide efficient any-to-any IPv6 routing over constrained links. Each LBR communicates with neighboring meters using an 802.15.4 interface, and distributes the topology it learns from them to the other routers using the building's Ethernet. We ultimately deployed seven LBRs over the four floors of the building. Each LBR advertises a minimum-cost path to neighboring meters. Each meter then chooses the closest LBR as its default router, and sends all traffic to the selected LBR. This allowed us to increase both network and backhaul capacity by deploying new meters and routers at will.

Data generated by the meters are sent in UDP packets, destined to a machine in our datacenter. This **datacenter tier** makes up the final part of the system, which runs as a hosted web application. Data packets from the meters traverse several network segments en route to the data center, moving from the `6loWPAN` network out over a local subnet before traveling through an IPv6 tunnel to the open IPv6 Internet. By extending our deployment to the Internet, we are able to share backend infrastructure between this and other meter deployments.

3.1 Embedded Application

Due to the large scale and long duration of the planned deployment, key goals were simplicity and reliability while maintaining enough flexibility to accommodate shifting deployment conditions and science requirements. Although there are sophisticated techniques available for over-the-air reprogramming, time synchronization, and distributed debugging, the reality is that each of these components add code size and testing complexity. Therefore, we either simplified or eliminated many complicated services. An invaluable simplification was to extract the most commonly changed configuration parameters like sampling rate and calibration parameters, and make them settable without reprogramming. Although we also included a simple over-the-air image update utility, in practice it was rarely used since we could update parameters without reflashing the entire image. Since we used an IPv6 routing protocol to support point-to-point and multicast traffic, implementing this functionality in the application was much simplified, yet extremely valuable to debugging. For instance, over-the-air reprogramming is implemented using TFTP instead of a complicated epidemic protocol [12].

3.2 Data Generation

The metering devices sample average power and total energy every 10 seconds, and send a packet every 20 seconds with the previous two readings. This period was chosen to be the highest we could reasonably support in high-density deployments. We pack two readings into a single packet. One of the most important aspects of sensing is the notion of time. It allows correlation of readings across meters. Even though approaches to time synchronization of various sophistication have been used successfully [4, 19, 25], problems with it are well documented [35]. Given our requirements, our approach was very simple. Therefore, we included redundant clocks and counters to distinguish the various events we expected in such a large deployment; these counters are presented in Table 1.

Clock	Rate	Purpose
LocalTime	32kHz	Time since reset or rollover
GlobalTime	1Hz	Unix timestamp
SequenceNumber	1/pkt	Monotonic value for the life of the meter
InsertTime	1/pkt	Timestamp at database

Table 1: Counters stored with each data packet.

In normal operation and for energy analysis, only GlobalTime is needed, since data are timestamped at the meter with the number of seconds since January 1, 1970. Other timestamps become useful for *post facto* analysis. For instance, packet delivery ratio (PDR) is a commonly-reported metric of network performance: it consists of the ratio between packets delivered and packets originated. In a large system where there is uncertainty about device state (*e.g.*, some devices may be broken or off), it is not possible to compute a "true" PDR. However, using SequenceNumbers which are stored in non-volatile flash memory, we approximated the number of packet originations even when the device is frequently rebooted. In another example, device reboots in our study change the set of motes available over time, and may indicate end-user attempts to save energy. The LocalTime clock quickly indicates reboot events, which would otherwise need to be inferred from data loss and could not be distinguished from network outages.

3.3 Design Takeaways

In reviewing recent deployment literature, we found numerous instances of malfunctioning networking stacks. By using a well-tested existing stack rather than starting from scratch, we avoided many of the networking bugs that plague deployments. By constraining the system to a single use and a set of protocols that have been tested together, we avoid the need for more complicated resource-sharing arrangements [13].

Our key insight was the value of *configuration, not reprogramming*. In the two or so years since our mote software was "released to manufacturing," we have not developed a new image; instead we have been able to work around the few, minor, and well-understood bugs in the existing software. Because of the ability to change parameters, we have been able to deploy meters in other systems not discussed here with differing needs but the same software. This results in a significantly smaller testing surface. Reprogramming hundreds of meters by unscrewing the case and using a wired programmer only needs to be experienced once to motivate a better solution. Second, although only a global clock is needed for data analysis, *you can never have enough clocks*. Since there are several notions of ordering, it is difficult or impossible to compute metrics like data yield independent of power failure, or node uptime without multiple counters.

Finally, IPv6 allowed us to develop compact implementations of many services wished for in previous deployments, like a configuration manager and a software updater. Although some of these were less efficient than the state-of-the-art, they worked when required at a very large scale.

4. NETWORKING INSIGHTS

An important component of our network is the routing protocol that provisions routes from the individual meters towards the edge of the access network. The protocol needed to be reliable and perform at a scale of hundreds of individual meters. We used HYDRO [8], a conceptual predecessor to the RPL protocol currently being advanced in the IETF [36]. HYDRO builds a directed acyclic graph (DAG) towards a set of load-balancing routers (LBRs). Traffic originating from inside the network is routed down this DAG to one of the LBRs, where further routing decisions are made. Traffic originating from an LBR or another network is first routed to the "nearest" LBR, and then source-routed to its final destination.

HYDRO contains numerous mechanisms to improve reliability and scalability in the face of shifting link conditions and deployment sizes. Each router maintains a list of potential next-hop default routes and will attempt delivery to several of them, using link-layer acknowledgements to determine whether a particular packet was delivered. Each embedded router also maintains only a subset of its neighbor set to limit the amount of state and periodically attaches this information to outgoing data packets. The LBRs use this information to build a view of the link state of the network and construct source routes back into the network.

At its peak, our network consisted of 455 nodes spread across four floors, routing through 7 LBRs. The average node density was at least 16; we estimate this by counting the number of distinct links reported over the life of the deployment.

4.1 Routing Requirements

When in production, our application collects data using a typical multipoint-to-point (MP2P) traffic pattern, which is well-known to be a key traffic pattern for embedded networks [20]. This traffic pattern is optimized in HYDRO by maintaining a DAG towards egress routers with a large amount of redundancy. Point-to-point routing in this type of network is less common. We made use of unicast routing primarily for management. For example, during meter calibration, we sent calibration parameters to be written to non-volatile flash, and during deployment, commands were sent to change the destination of data.

The ability to communicate with each meter *in situ* to troubleshoot was valuable, but infrequently used. This leads us to the main insight: *point-to-point routing should be available, but may be expensive.* In the case of HYDRO, all unicast traffic is source-routed from an LBR. Since most traffic flows out of the network to an LBR, adding a small amount of extra data to maintain the topology is inexpensive, and eliminates the need for a flood of discovery messages as would be required in an on-demand protocol. This is somewhat in opposition to the networking structure that has emerged in recent years in TinyOS, positing that *collection* (MP2P) and *dissemination* (P2MP) are the dominant traffic patterns. Although these are invaluable and common traffic patterns, at times it is simply convenient to contact a single node individually.

A second requirement was the ability to *scale using multiple LBRs.* HYDRO supports this by extending the routing topology over a backhaul link; in our case, this was the building Ethernet. This allowed us to have multiple, redundant LBRs; indeed, it was common for one or two of them to be offline for various reasons. Until protocols and implementations support this functionality, they should not be considered for large-scale deployments.

4.2 Data Loss

Since our network was situated in a real environment for upwards of a year with a large number of devices, we experienced practically every form of data loss at one time or another. Table 4.2 summarizes the causes of missing data which we diagnosed at various points in time. Although protocol loss is the most well-studied form of loss, we found that the amount of data missing from other causes dwarfed the amount dropped due to routing errors or forwarding path errors. The high-level insight is that *if perfect reliability is required, buffering must be present between every link which may fail.* This is a notable consequence of our use of IPv6, the data path spanning multiple links, and the end-to-end principle. Given that there are devices of varying capabilities and links with varying reliabilities on the path from a meter to the database, it may be worthwhile in the future to use an application protocol with support for intermediate caching proxies.

One of the most unexpected sources of missing data was that of device unpluggings. During the deployment, occupants were instructed that meters were attached to devices, not receptacles. Therefore, they frequently unplugged the meter along with the device. Figure 3 shows the frequency of these "on" and "off" events in time. An on event is when a meter begins reporting after having been turned off, while an off event is when it stops sending data because it is unplugged. After interviewing several study participants, we discovered that it was common for a meter to be plugged into a power strip, which was turned off at the end of each workday. This behavior was intended to reduce the leakage power consumed by devices plugged into the strip, but also caused network churn and gaps in the data!

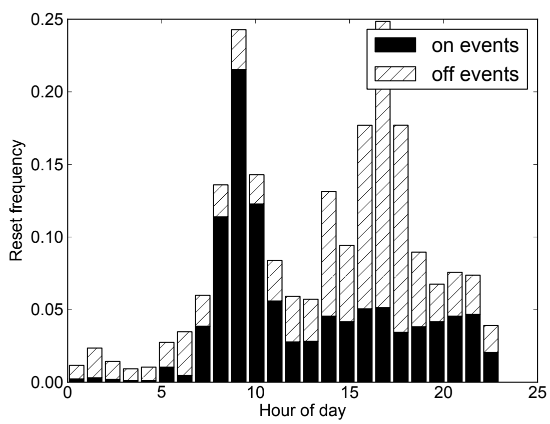


Figure 3: Not all network dynamics are caused by link and node failure or mobility; some devices were turned off or unplugged during nights. These events correlate strongly with working hours.

Once data loss due to LBR connectivity problems and meter unpluggings is filtered out, we can observe the overall data yield in the month of April, shown in Figure 4. The 5^{th} percentile yield across all devices was over 99% nearly every

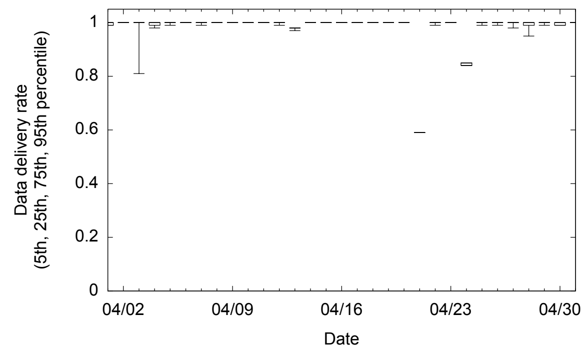


Figure 4: Observed data yield distribution in April, with losses due to mote shutdowns removed. Losses due to database maintenance are clearly visible as days with lower yield but little variation. For many days, the 5^{th} percentile yield was above 99%. Boxes show 25^{th} and 75^{th} percentiles, whiskers show 5^{th} and 95^{th} percentiles. Data were delivered using UDP datagrams with no end-to-end mechanisms.

day, and the best value was 99.91%; the median was 98.7%. Because we did not deploy meters with an eye towards connectivity, certain meters had consistently poor connectivity; however the deployment was dense enough that the size of this set was only 12 meters, or less than 3% of meters deployed. These numbers refer to drop rates of the best-effort UDP datagrams sent from meters to the database and compare favorably with published estimates of Internet-wide packet loss [5, 33].

4.3 Network Dynamics

Running a routing protocol in a large network over a year gives us a rich set of data with which to observe link dynamics. First, we have application-layer data reported by the motes to the data repository. These UDP packets are sent every 20 seconds, traversing several network segments. These contain the local time, global time, sequence number, current default next hop and link estimate, and the current path cost estimate. These data provide the ultimate metric of application-layer performance. Second, HYDRO routers apprise the LBRs of their link-state at regular intervals so that they can build source routes back into the network. We stored a snapshot of this state every five minutes, which consists of the top four entries in each router's neighbor table, along with corresponding link estimates (ETX). These estimates are kept fresh by a periodic exploration process each router conducts to sample neighboring links.

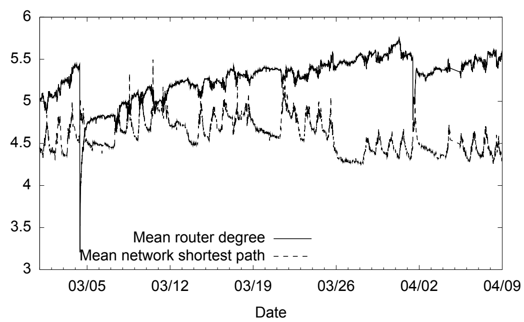
Using these data, we are able to confirm several previously proposed hypotheses about link dynamics: *intermittent links are prevalent*, *their uses reduced routing stretch by a factor of 2*, and *networks experience significant diurnal and weekly variations.*

4.3.1 Daily Variations

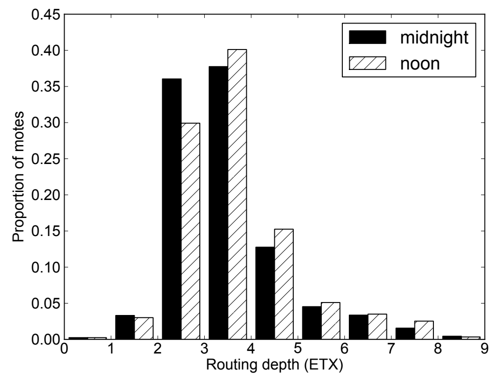
Figure 5(a) shows a basic look at the collected topology for March. In most of these analyses we focus on March and April when the set of meters was stable and there were infrequent database or connectivity outages. The dashed line shows the mean shortest path between all pairs of nodes in

Loss cause	Description
Network loss	Packets dropped by the link layer or routing protocol due to exceeding the number of retransmissions or hop limit.
Device shutdowns	Devices attached to power strips were frequently turned off overnight.
Building shutdowns	The facilities organization disconnected power for maintenance, resulting in a 14-hour outage and subsequent network reinitialization.
LBR connectivity	The LBRs were networked through an Ethernet VPN, and connected to the IPv6 Internet with a tunnel. Both of these network components were subject to failure.
Software faults	We observed a software failure rate of approximately 10 nodes per month, or one every 3.5 mote-years. This could be improved through better software engineering.
Database failures	Months of data were lost when MySQL tables were corrupted during a software upgrade.
Meter attrition	Over a period of time, some devices were removed from the system by maintenance workers or occupants. Approximately 100/455 of our meters disappeared over the course of the study.

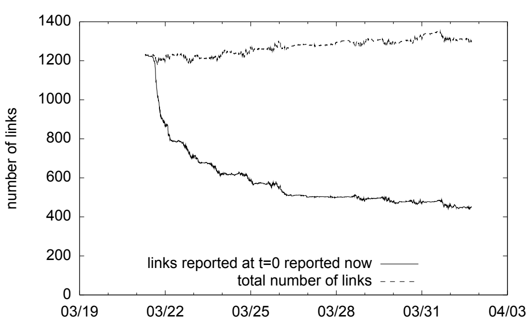
Table 2: Missing data in real deployments are caused by many factors beyond the control of the deployment staff.



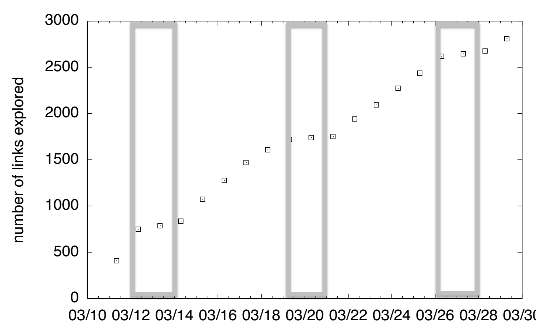
(a) HYDRO quickly builds a rough link-state database, and then slowly improves its view as individual routers explore new links during this 1.5 month-long section of the deployment. Routes become approximately 10% longer during the workday, and then relax at night. Reported degree gradually increases while path length decreases, as HYDRO discovers new links. Furthermore, a subset of meters are only on during working hours. The network was reinitialized on 3/5/2011.



(b) Compared to the topology at midnight, routes during the day are approximately 10% longer. This shift occurs each day as changing noise conditions cause the routing protocol to evict grey links and search for better candidate parents.



(c) The link set is actually quite dynamic. At the start of the analysis we snapshot the link set reported at that time, and compare the size of its intersection into the future. Less than half of the links appear after a week. Without this dynamic set of links, the network diameter increases by a factor of 2.



(d) The set of links undergoing exploration continues to grow, rather than stabilize. The network is quite dense, and each router has many more potential default routes than can fit in the neighbor table. The exploration process halts on weekends (in grey) when better conditions no longer force the routing protocol to adapt.

Figure 5: Running a routing protocol over a long period of time in a real environment confirms many hypotheses about link behavior over time.

the network, while the solid line shows the mean reported router degree. Since HYDRO reports only a partial subset of neighbors it is using for routing, this degree is not the "true" degree of the network. The weekend to weekday distinction is clearly visible, with path lengths remaining stable on weekends but increasing by 5-10% on weekdays. Figure 5(b) shows this effect more clearly, contrasting node routing depth in terms of ETX at midnight with depth at noon; the shift to deeper ranks is clearly visible.

4.3.2 Routing Churn

There are numerous possible explanations for these daily variations. Many studies suggest that there are links with bimodal behavior [29], alternating between "good" and "bad" states with various frequencies. One difference with these studies is that we do not have ground truth data about these links. Rather, we have link data filtered through the actions of a routing protocol. However, it is interesting to consider the actions of the protocol by examining the links it selected in a real-world setting. Figure 5(c) shows the result of an analysis designed to uncover how much churn there is in the nodes' next hop choices. At the start of the analysis (chosen arbitrarily after the network had been initialized for several weeks), we form a set containing all links reported by the network. At each subsequent time step, we find the size of the intersection of the set of reported links at that time with the original link set.

We find that the protocol consistently reports a set of about 500 links, which we call the **static links**. The remainder of the original set of links disappear after about a week. Since HYDRO removes links which are not useful for routing as well as poorly-performing links, there is a subset of links that are both consistently both useful and reliable. However, HYDRO also discovers and reports a constantly changing set of other links, which we call the *churn links*. The total number of links reported by the network (from all nodes) is consistently around 1200; each router was configured to report a maximum of 4. This indicates that the density of this deployment provided each router with ample choice of next-hop, although not all were consistently available.

Figure 5(d) addresses whether the churning set of links is of a fixed size or is dynamic. At the beginning of our analysis we remove all static links and at each time step add all new links which are not part of the static set. HYDRO consistently tries new links; the number of links it tries does not stabilize in the time frame of the experiment. Over 20 days it tries almost 3000 discrete links; we infer that this churn is caused by changes in noise floor and interference conditions, since *the exploration process comes to a virtual halt on weekends.*

The final question was whether the effort to discover and track these variable links was "worth it." That is, does the set of churn links improve routes to the LBRs? The answer is a resounding "yes." If we were to consider only the links in the static set, *i.e.*, those continuing to appear after a week, *the network diameter increases by a factor of 2.1: from an ETX of 4.6 to 9.9.*

4.4 Networking Takeaways

We chose not to use TCP, opting instead for UDP, to use only the best-tested code and because the reliable delivery by the raw network was sufficient. The introduction of so many parallel TCP flows in a large deployment was an untested proposition. Furthermore, even TCP's end-to-end acknowledgements are not a panacea for data reliablity problems unless much care is taken managing send buffers on the mote to guarantee all data is delivered in the face of connection resets and timeouts. Transport- and application-layer protocol designs currently in development (for instance, CoAP [26]) should be evaluated to ensure that they provide appropriate mechanisms for caching between network segments, especially if they are designed to work over UDP. *It is an open question of how to merge the buffering and end-to-end demands of TCP with the constrained resources of embedded systems.*

Furthermore, we have largely validated earlier studies about the importance of grey links to routing, We infer that the churn links which are coming and going from the routing table must have varying quality. Even in a very dense deployment, failing to take advantage of links that are not consistently high-quality would have resulted in paths twice as long as we experienced. Even in a deployment with no mobility, we must maintain a continuous process of routing table maintenance and discovery to achieve good performance.

5. ENERGY SCIENCE

A mark of good system design and construction in a sensor networking deployment is the collection of scientific data that enable better visibility into a phenomenon that could not previously be observed. In this section, we document the domain-specific lessons derived from the preparations taken to collect the data, including the calibration of meters for accurate collection of data, stratified appliance sampling for guiding the deployment strategy, and share insights gained from the data.

5.1 Accuracy Analysis

For energy meters, calibration transforms measurements taken by the device into engineering units usable for scientific comparison. This process presents a number of challenges. First are accuracy requirements: in the MELs regime, loads seldom consume more than $300W$, with a large proportion below $60W$. We are not aware of a documented testing regimen that covers low load levels. For example, the California Public Utilities Commission requires expensive, utility-grade electric meters to be accurate to within 2% of load from $60W$ to $3.6kW$, ignoring the lowest range. [30] A second requirement is simple, yet individualized calibration: there are differences among metering devices caused by variations in the manufacturing process, but the calibration equations need to be simple enough to be computed on the devices themselves. Thus, each device needs to be tested, calibrated, and programmed separately. Last, to ensure that the process of calibrating all 455 meters is not cost- or time-prohibitive, it should be quick and automated.

We used these goals to guide the design and development of an automated calibration process. It uses eight different resistive loads from $2W$ to $100W$, actuated by a computer-controlled relay to create 21 calibration points between $0W$ and $300W$. A reference power meter [10] is used for ground truth. Additionally, the process is designed to handle five meters plugged in concurrently, accelerating the calibration process. Each calibration run takes 5 minutes, reducing the time per meter to 1 minute.

The process of developing a calibration function revealed a couple of key insights. First after analyzing raw data from several hundred meters, the raw values exhibited highly linear behavior *but only over limited domains*. In fact, a single linear function over the entire range of loads cannot meet our accuracy target. Therefore, *single-point calibration is insufficient*. Instead, a piecewise linear function using multi-point calibration as shown in Figure 6 may be used, consisting of three portions, where the first only ensures that the meter returns a zero reading at zero load, and the boundary between the second and third segment is chosen to minimize the overall error in the calibration. Although more segments in the function would decrease error, there is risk of overfitting as well as unnecessary complexity. The calibration coefficients are stored on each meter so that future readings are provided in engineering units (mW) for easier analysis.

The second insight is that despite larger percentage error at lower load levels, absolute error remains quite low when examined across all meters. The error of this calibration procedure for the entire population of 455 meters is shown in Figure 7. The plot shows cumulative distribution functions at four load levels. At low loads, we achieve absolute errors of less than $1W$ for virtually all of the meters calibrated. Additionally, more than 75% of meters are within 2% of the measured load at $60W$, the standard for “utility-grade” metering, with improved accuracy as the load increases. These results demonstrate that inexpensive metering hardware can be calibrated both quickly as well as reasonably accurately.

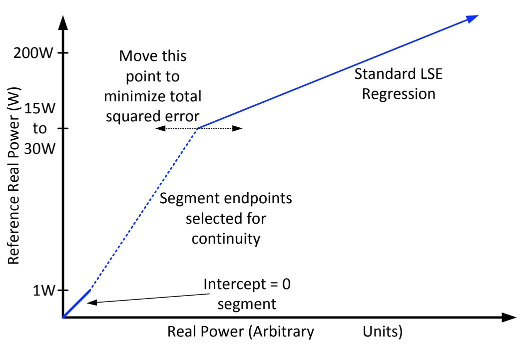


Figure 6: Selection of piecewise linear calibration coefficients.

5.2 Energy Results

The detailed metering of MELs enabled by this work provides an opportunity to understand MELs energy use in a way previously not possible. In a sub-metered building, which is still rare, MELs energy use is typically a single number for the entire building because only the primary end-uses (*i.e.*, heating, cooling, ventilation, lighting and water heating) are metered, and MELs consumption is found by subtraction from the building total. On the other hand, in this study, the fine resolution in device type and time resolution allows energy to be divided by device type, operational mode, or a variety of other parameters.

Section 2.2 described the methodology for selecting a stratified sample of devices for metering. Here we provide the quantitative justification for that experimental design decision. Figure 8 presents the count as well as annual energy

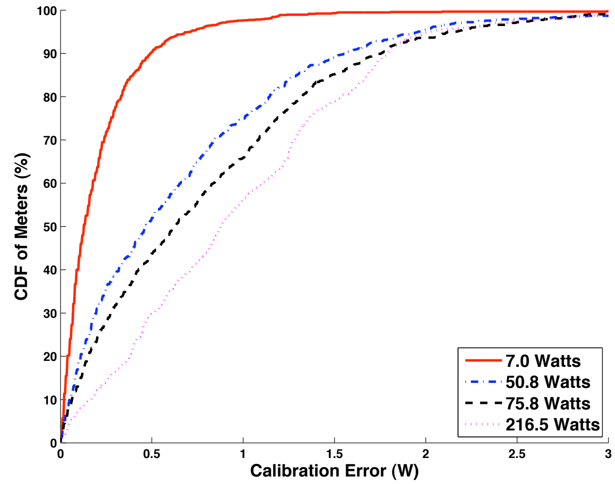


Figure 7: Error of calibrated meters at four load levels.

consumption of devices in the top seven energy-consuming categories and all other devices for the commercial study building.

Energy estimates for the entire population are projected from the metered sample of devices using sample probability weights, and energy is projected from the metering period to the entire year. Computers clearly use both the most energy overall and the most per unit, whereas the Other category of devices shows the opposite behavior. Because the building is primarily office space, displays, imaging, networking and miscellaneous (*i.e.*, task) lighting are the next largest energy users. Space heating and fans make up most of miscellaneous HVAC, and the appliances are primarily refrigerators found in break rooms and a few offices. The energy breakdown shows that information technology equipment consumes over 75% of annual MELs energy but comprises less than half of total devices, emphasizing that *IT devices should be disproportionately metered more when studying MELs energy use in offices*. This finding corresponds well with surveyed sources of these data [32].

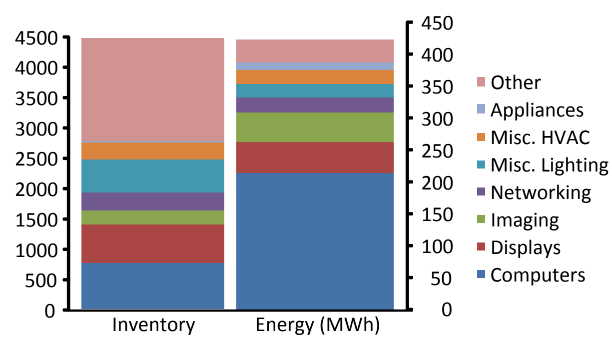


Figure 8: Comparison of the device inventory to annual energy use by device category.

Another dimension in which to view the data is temporally. The average weekday load shape divided by category is shown in Figure 9, providing insight into typical energy usage patterns. The largest energy-consuming category, computers, uses about 60% of its energy outside working hours at this facility. Considering that only 10% of the computers in this facility must remain on at all times, this presents an enormous opportunity for energy efficiency via power management on computers. In fact, this facility could save over 100,000kWh ($\approx$ \$15,000) annually without loss of access or productivity by reconfiguring operating systems [1] or installing widely-available software [2, 15, 22] for this purpose.

Additionally, it is clear that significant energy is used in most other device categories outside of working hours. Timer-controlled plug strips could shut off power to office devices except appliances and network equipment during non-working hours. If used, they provide a relatively inexpensive solution that would save 30% of non-computer energy use. Even though these changes have a negligible cost and despite the popularity of energy efficiency in commercial settings, there is still huge potential for saving money and energy in the MELs end-use category.

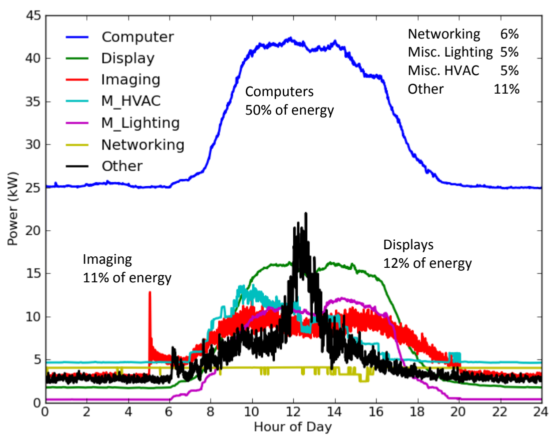


Figure 9: Device category average load shapes.

A cost effective way to save energy is to ensure that only energy-efficient products are used in a building. For IT equipment, price and efficiency are not tightly correlated. Consider computer displays. A comparison of the on-mode power of the metered devices shows that the average power is 33W for displays smaller than 24". However, the 10th percentile on-mode power of 24" displays is only 20W. If all of the smaller monitors were replaced with 20W 24" displays during the next upgrade cycle, users would not only have greater screen area, but total display energy use would decrease by 40%.

6. RELATED WORK

Our study of a large-scale, long-lived energy metering wireless sensor network benefits from previous efforts in two key areas: WSN deployment science and studies of routing and link behavior in low-power, lossy networks.

Though a range of studies document the design, development, and evaluation of large-scale and occasionally long-lived WSNs [3, 6, 9, 18, 21, 34], there are a notable few that provide systematic guidance based on difficulties experienced in deploying large-scale, long-lived WSNs. We designed our system and methodology considering these valuable insights. For example, Langendoen, *et al.* "advocate the manifold use of statistics" to have frequent state reports available for potential debugging efforts [17]. In our study, node and LBR state are continuously logged. Each data packet provides a number of different time counters for *post hoc* assessments; further, routing state and periodic snapshots of LBR link-state allows reconstruction of network dynamics. These data have been invaluable for diagnosing protocol behavior and connectivity issues. Barrenetxea, *et al.* draw from years of experience deploying networks in the Swiss Alps to provide guidance through the entire development, testing, and deployment process [4]. They thoroughly summarize issues described in previous deployments augmented with their own experiences to provide a "hitchhiker's guide" to outdoor WSN deployments. In the development phase, they advocate "keeping it simple" and avoiding complexity wherever possible, particularly to circumvent unexpected interactions between software components especially in the communications stack. Many of our design decisions reflect this parable, including our selection of a well-tested routing protocol. Hnat, *et al.* provide a similar guide aimed towards indoor deployments [11]. A particular insight is that despite common belief, deployment in buildings still presents significant connectivity and access challenges. In our case, the ability to cope with a consistently varying noise floor by using an adaptive routing protocol as well as our range of remote debugging and reconfiguration services enabled us to address these challenges.

Adherence to the lessons in the literature coupled with thorough planning and understanding of the deployment environment allowed us to avoid many of the issues that plague WSN deployments, resulting in a successful long-lived and large-scale deployment. It is this combination of long life, large scale, and good planning that sets us apart from previous work in plug-load energy metering. The PowerNet project deployed 85 wireless electricity meters for 3 months to examine IT-related electricity consumption [16]. Also, Jiang, *et al.* deploy 39 plug-load meters in an office space for several months and look at disaggregation of appliances as well as decomposition on different parameters [15]. However, both of these studies lack the rigor in sample design and long time-scale needed to understand overall MELs usage in their respective settings. Additionally, neither describe calibration processes to ensure the accuracy of the collected data.

There is also a body of work related to the understanding of link dynamics. Zhao and Govindan characterize the size and behavior of "grey zones" of intermittent performance in low-power links, while Son, *et al.* extend this understanding to multiple contending senders [27, 37]. Srinivasan, *et al.* synthesize these results into a more detailed understanding of these link dynamics, and finally propose a metric called β which is used to characterize the temporal variation in links [28, 29]. Finally, Ortiz attempts to quantify the value of variable links and multiple channels for reliability when routing is also an option [24]. These studies informed our understanding of the HYDRO protocol, and also allowed us to infer what was happening inside of our network from the external signals it presented to us, like routing state. They also confirm the importance of using a protocol which expects significant temporal variation from the links in use.

7. CONCLUSION

Our results present several new findings and insights. For instance, the design decision to explore and use links that become unstable at certain parts of the day was essential to achieving low path costs. Using these links results in path costs half of what would be achieved using exclusively very stable links. In general, we found that although there was almost no mote mobility, there were still dynamics caused by a changing noise floor and a shifting set of devices which are on the network: the interior of a commercial building is indeed a dynamic environment. We also confirm the value of point-to-point routing in a real sensor network deployment.

There are also several takeaways which should inform future protocol design. It is still somewhat unusual to connect large subnetworks of devices directly to the broader Internet but gave us host of advantages in terms of debugging and visibility; this may cause researchers in the embedded space to examine transport protocols which can provide better end-to-end guarantees than either UDP or TCP while continuing to impose a very low burden on the low-power and lossy network segments. Incorporating reliability at every layer of design is common practice in other domains; we expect it to become common here as well.

The result of this work is to allow deployments at larger scales. We have presented results from @scale; one of the largest deployment in terms of mote-years yet published. We were successful in achieving our science goals while testing several hypotheses about network dynamics, and reinforced emerging design practice on the construction of this type of network.

Acknowledgements

We would like to thank Ken Lutz, Albert Goto, and Xiaofan Jiang of Berkeley and Iris Cheung, Judy Lai, and Margarita Kloss of LBNL for their help with the design, manufacture, calibration, and deployment of the @scale system. Alice Chang and Shelley Kim provided able assistance in calibrating and programming the meters; Sara Alspaugh of Berkeley contributed Figure 2. Ray Woodfin and Joann Nguyen of Megaforce Corporation provided support in manufacturing these devices. This work was supported by the DOE Building Technologies Program and the National Science Foundation under grants CPS-0932209 and CPS-0931843.

8. REFERENCES

[1] Advanced configuration and power interface specification – revision 4.0a. http://www.acpi.info.

[2] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta. Somniloquy: augmenting network interfaces to reduce pc energy usage. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, NSDI'09, Berkeley, CA, USA, 2009. USENIX Association.

[3] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier*, 46:605–634, 2004.

[4] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 43–56, New York, NY, USA, 2008. ACM.

[5] J.-C. Bolot. End-to-end packet delay and loss behavior in the internet. *SIGCOMM Comput. Commun. Rev.*, 23:289–298, October 1993.

[6] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Gună, M. Corrà, M. Pozzi, D. Zonta, and P. Zanon. Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment. In *In Proc. of the 8th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks*, 2009.

[7] S. Dawson-Haggerty. Design, implementation, and evaluation of an embedded ipv6 stack. *Masters Thesis, University of California at Berkeley*, 2010.

[8] S. Dawson-Haggerty, A. Tavakoli, and D. Culler. Hydro: A hybrid routing protocol for low-power and lossy networks. In *First IEEE International Conference on Smart Grid Communications*, October 2010.

[9] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler. Trio: Enabling sustainable and scalable outdoor wireless sensor network deployments. In *IEEE SPOTS*, pages 407–415, 2006.

[10] Electronic Product Design, Inc. Plm-1 power meter. http://epd.com/power_meters.html#PLM-1.

[11] T. W. Hnat, V. Srinivasan, J. Lu, T. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse. The hitchhiker's guide to successful residential sensing deployments. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, New York, NY, USA, 2011. ACM.

[12] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 81–94, 2004.

[13] J. Il Choi, M. A. Kazandjieva, M. Jain, and P. Levis. The case for a network protocol isolation layer. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 267–280, New York, NY, USA, 2009. ACM.

[14] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity ac metering network. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 253–264, Washington, DC, USA, 2009. IEEE Computer Society.

[15] X. Jiang, M. Van Ly, J. Taneja, P. Dutta, and D. Culler. Experiences with a high-fidelity wireless building energy auditing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2009. ACM.

[16] M. Kazandjieva, O. Gnawali, B. Heller, P. Levis, and C. Kozyrakis. Identifying energy waste through dense power sensing and utilization monitoring. Technical report, Stanford University, 2010.

[17] K. Langendoen, A. Baggio, and O. Visser. Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture. In *In Int. Workshop on Parallel and Distributed Real-Time Systems*, 2006.

[18] C.-J. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao. Racnet: A high-fidelity data center sensing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, New York, NY, 2009.

[19] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, New York, NY, USA, 2004. ACM Press.

[20] J. Martocci, P. D. Mil, N. Riou, and W. Vermeylen. RFC 5867: Building automation routing requirements in low-power and lossy networks, 2010.

[21] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai. Canopy closure estimates with greenorbs: sustainable sensing in the forest. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2009. ACM.

[22] S. Nedevschi, J. Ch, S. Ratnasamy, and N. Taft. Skilled in the art of being idle: reducing energy waste in networked systems. In *In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, 2009.

[23] B. Nordman and M. Sanchez. Electronics come of age: A taxonomy for miscellaneous and low power products. http://eetd.lbl.gov/ea/nordman/docs/taxonomy.pdf.

[24] J. Ortiz and D. Culler. Multichannel reliability assessment in real world wsns. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, New York, NY, USA, 2010. ACM.

[25] T. Schmid, Z. Charbiwala, Z. Anagnostopoulou, M. B. Srivastava, and P. Dutta. A case against routing-integrated time synchronization. In *Proceedings of the 8th ACM*

Conference on Embedded Networked Sensor Systems, New York, NY, USA, 2010. ACM.

[26] Z. Shelby, K. Hartke, C. Bormann, and B. Frank. Constrained Application Protocol (CoAP). Internet-Draft draft-ietf-core-coap-07, Internet Engineering Task Force, July 2011. Work in progress.

[27] D. Son, B. Krishnamachari, and J. Heidemann. Experimental Analysis of Concurrent Packet Transmissions in Low-Power Wireless Networks. Tech Report ISI-TR609, Viterbi School of Engineering, University of Southern California, 2005.

[28] K. Srinivasan et al. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. Tech Report SING-06-00, Department of Electrical Engineering, Stanford University, 2006.

[29] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, New York, NY, USA, 2008. ACM.

[30] State of California Division of Measurement Standards. Device enforcement program manual, 2008. `http://www.cdfa.ca.gov/dms/programs/devices/DeviceProgramManual2008.pdf`.

[31] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. A macroscope in the redwoods. *ACM Sensys*, 2005.

[32] U.S. Department of Energy. *Buildings Energy Data Book*. 2009.

[33] A. Wang, C. Huang, J. Li, and K. W. Ross. Queen: Estimating packet loss rate between arbitrary internet hosts. In *10th International Conference on Passive and Active Network Measurement*, volume 5448 of *Lecture Notes in Computer Science*. Springer, 2009.

[34] T. Wark, C. Crossman, W. Hu, Y. Guo, P. Valencia, P. Sikka, P. Corke, C. Lee, J. Henshall, K. Prayaga, J. O'Grady, M. Reed, and A. Fisher. The design and evaluation of a mobile sensor/actuator network for autonomous animal control. In *Proceedings of the 6th international conference on Information processing in sensor networks*, New York, NY, USA, 2007. ACM.

[35] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. *USENIX OSDI*, 2006.

[36] T. Winter et al. draft-ietf-roll-rpl-19: RPL: IPv6 routing protocol for low power and lossy networks. (in progress), 2011.

[37] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, New York, NY, USA, 2003. ACM.