This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

Pervasive Energy Monitoring and Control through Low-Bandwidth Power Line Communication

Sean Barker[†], David Irwin, and Prashant Shenoy
Bowdoin College[†]
University of Massachusetts Amherst

***Abstract*—The Internet of Things (IoT) is growing rapidly, with increasingly sophisticated networking, sensing, and actuation functions embedded into everyday devices. One important IoT application is managing a building's energy usage by monitoring and controlling its electrical devices. Many existing IoT-enabled devices operate through low-cost, convenient power line networks, using protocols such as X10 and Insteon for communication. However, as these technologies have traditionally targeted low-bandwidth device control, they are often not readily suited to higher bandwidth uses such as continuous energy monitoring. In this paper, we consider the challenge of leveraging existing low-bandwidth PLC networks for energy monitoring, and present several techniques that enable reliable, high-resolution monitoring in such networks. As a case study, we consider the popular Insteon protocol, and show that intelligent polling and event detection methods can reduce the bandwidth requirements and undetected power events in a real-world Insteon network by 50% or more versus naive methods. Our techniques have been employed in a real IoT-enabled smart home, which has collected much of the data publicly released in the UMass Smart* energy dataset.**

***Index Terms*—Smart home, home automation, energy, sensor, actuator, power-line communication, Insteon**

I. Introduction

The Internet of Things (IoT) is experiencing rapid growth, with many companies embedding networking, sensing, and actuation functions in everyday household devices. Prominent examples include the Nest thermostat, Belkin's WeMo line of IoT devices, Philips' Hue light bulb, and Samsung's Smart Washing Machine and Dryer. One important IoT application is the granular management of building energy consumption by monitoring and controlling the energy usage of these IoT-enabled devices. In doing so, "smart" buildings are capable of using *demand-side energy management* to reduce their overall energy consumption and peak power usage, while also better aligning consumption with renewable generation [1]. Demand-side management requires buildings to *continuously monitor* devices' power usage, and *remotely control* when and how much energy they consume.

Since buildings do not generally have wired Ethernet in power outlets, thermostats, or wall switches, most devices must rely on various other types of networking mediums and protocols. One popular networking option uses Power Line Communication (PLC), which leverages the physical electrical wiring of the building to send and receive messages.

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Since all buildings (smart or otherwise) contain extensive existing electrical wiring, networks that operate over the power line infrastructure offer key benefits, such as rapid adoption, minimal infrastructure cost, and easy integration into existing building environments. However, while remote control and monitoring in smart buildings are closely coupled, two disjoint sets of technologies have evolved to perform these tasks.

- **Control**. Power line based Home Automation (HA) protocols, such as X-10 and Insteon, were designed explicitly for remote device control and are widely used in smart buildings. The protocols enable programatic actuation of outlets and switches (i.e., turning on or off) using a building's power line for communication. Since remote control typically only requires infrequent, brief commands, these protocols offer little bandwidth, and cannot support direct monitoring of power at high resolutions. However, they are often more reliable than wireless communication, owing to specific design choices that sacrifice bandwidth for reliability (e.g., Insteon devices act as message repeaters).
- **Monitoring**. In contrast to PLC protocols like Insteon, wireless network substrates and protocols have received more attention for energy monitoring. Researchers have developed numerous techniques to enable high resolution wireless monitoring of a device's energy consumption at various spatial and temporal dimensions. Several past efforts focus on outlet- and switch-level monitoring using wireless technologies, such as Z-Wave, ZigBee, and WiFi. WiFi is especially attractive, since it enables devices to connect directly to the Internet in buildings that already support a WiFi network. Of course, wireless protocols that support high-bandwidth monitoring are also capable of supporting low-bandwidth control commands. Under ideal conditions, wireless communication offers enough bandwidth to monitor and control the energy usage of hundreds of outlets at high resolution, e.g., every second.

Unfortunately, in practice, since outlet and switch boxes are embedded in walls and may be behind large appliances or in the extremities of a building, wireless communication often exhibits interference that severely degrades its performance and reliability. In addition, commercial off-the-shelf devices that support wireless monitoring and control via WiFi, such as the Belkin WeMo Insight Switch, have only been widely available in the last few years. In contrast, PLC-based HA devices are widely deployed, having been available for many years (X10 since the 1970s and Insteon since 2005).

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

Thus, while monitoring and control systems are important for managing smart buildings, they have evolved independently using disjoint protocols and standards. In particular, even though low-bandwidth PLC-based protocols have been deployed and used extensively for building control, they have not been used for high resolution energy monitoring, likely due to their extreme bandwidth limitations. In this paper, we examine how to unify a smart building's monitoring and control substrates into a single PLC infrastructure. Our goal is to enable reliable, high resolution energy monitoring and control across every device in a smart building, using only a low-bandwidth power line network, such as Insteon. While prior work attempts to mitigate the reliability issues of wireless communication [2], [3], we take the opposite approach by focusing on exploiting HA protocols to perform high resolution monitoring, which is motivated by the extensive deployment of PLC-based HA protocols for control. Despite the lack of prior work considering HA protocols for monitoring, these protocols have a number of advantages that make them attractive.

Commercial Availability. HA products have been commercially available for many years for a variety of devices, including appliances, lamps, wall switches, and outlets. In contrast, wireless energy monitoring and control was viewed as a research topic only a few years ago [4]. Thus, fewer wireless outlet and switch meters are commercially available, and the ones that are, such as the Z-Wave Smart Energy Switch or Belkin's WeMo Switch, only recently went on the market.

Open Standards. HA protocols are open standards that vendors can integrate directly into devices. In contrast, wireless devices often use proprietary protocols, such as Z-Wave, that complicates using them with third-party devices.

Backwards Compatibility. Since many smart buildings already use power line based HA protocols, our approach will augment existing deployments with monitoring functions. For instance, due to their maturity, HA protocols have already been adopted in many early demand-side management field trials.

Reliability. HA protocols' use of power line communication does not suffer from the interference problems that hinder wireless communication at large scales, and, thus, provides a reliable foundation for remote monitoring and control.

Unobtrusiveness. To reduce interference, wireless meters are usually not embedded in walls, but installed externally, e.g. by plugging into outlets, which makes them obtrusive. In contrast, HA devices are embedded into "normal" outlets and switches.

Despite the benefits, using power line based HA protocols for energy monitoring poses significant challenges.

- **Scalability.** HA protocols were not designed to support continuous monitoring traffic. For example, the MAC layers for HA protocols do not employ "standard" features, such as collision avoidance. Thus, monitoring the energy usage of even tens of devices at high resolution is challenging. Thus, a key challenge is *scaling* HA protocols to monitor many devices despite their limitations.
- **Accuracy.** HA protocols are capable of monitoring power state changes for switches and low resolution power usage for outlets. Thus, another key challenge is *accurately* translating switch state change events and coarse outlet power data into high resolution power measurements.

Contributions. In this paper, we make the following contributions. First, we detail and experimentally quantify Insteon's limitations for monitoring. Second, we develop a simple a model that describe's Insteon's reliability at different query rates. Third, we design a set of techniques that leverage a building-wide power meter to provide high resolution energy monitoring on top of the existing control capabilities in HA protocols such as Insteon. In particular, we focus on two such techniques: i) learning switch power by correlating state changes with changes in building-wide power usage, and ii) continuously monitoring outlet power usage via "smart polling" that judiciously uses the minimal bandwidth available. Our techniques have been employed for high-fidelity energy monitoring and control in a real smart home, which includes 62 HA-enabled wall switches and power outlets. Our system serves as the foundation for our publicly-released UMass Smart* energy dataset [5].

II. Overview

The primary drawback in using power line based HA protocols for energy monitoring at large scales is their extreme bandwidth limitations. While more recent power line based protocols, such as HomePlug, provide plentiful bandwidth (adapters capable of over 1Gbps are now available), they have not traditionally been used for HA. Instead, these new protocols target high-bandwidth data from general Internet traffic and multimedia devices, such as televisions. HomePlug is not typically embedded into standard outlets, switches, or devices due to cost, power/heat, and form factor constraints. While HomePlug Green PHY (or HomePlug GP) was recently introduced to mitigate these constraints, it is not in wide use. We focus on Insteon in our deployment, since it is a mature technology that extends the original X10 HA protocol with greater reliability and more bandwidth.

Figure 1 depicts our architecture, which includes Insteon-enabled wall switches (Insteon SwitchLincs) and outlets (Insteon iMeters). We assume a building "operating system" (OS) running on a server implements our monitoring and control system by interacting with these Insteon-enabled devices. Our techniques could be implemented within any building OS, including Microsoft's HomeOS [6], Apple's HomeKit, openHAB, BOSS [7], etc. To support control, the OS sends wall switches and outlets commands via a Power Line Modem (PLM) to alter their power state, such as turning devices on or off. To support monitoring, Insteon-enabled wall switches send asynchronous *notifications* to the OS whenever someone toggles the switch, while the OS must explicitly *query* Insteon-enabled outlets for their power usage. To understand Insteon's limitations for monitoring, we describe the protocol below and develop a simple model that captures its reliability.

A. Insteon Protocol

In the Insteon protocol, senders broadcast messages over a building's power line, while receivers listen for messages and send acknowledgements upon receipt. The protocol limits transmissions to brief intervals near where the alternating current (AC) crosses zero, which occurs twice every 16.6 ms

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

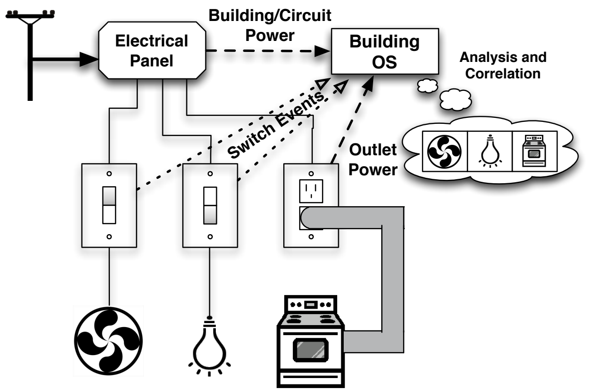

Fig. 1. Our monitoring and control architecture.

at 60Hz AC, since electrical noise impairs communication and is at minimum during the zero crossing. However, due to harmonic noise from power supplies or signal attenuation over long distances, devices may still not receive every message.

Thus, to increase reliability and range, all Insteon devices also act as repeaters that automatically repeat messages they hear a fixed number of times, based on a configurable *hops* field. Additional hops effectively increase each message's length by a factor of (1+*hops*). The simple broadcasts and hops alleviate the need for complex routing protocols to transfer messages. The protocol also avoids flooding and collisions when repeating messages, since all devices synchronize retransmissions using the 60Hz AC power line frequency—each transmission begins exactly 800 microseconds before the zero crossing and ends exactly 1023 microseconds after the zero crossing. Thus, when repeating, all devices transmit the same data at exactly the same time for the same number of times, which serves to further amplify and strengthen every transmission against electrical interference.

The Insteon protocol supports two types of messages: 10 byte standard messages and 24 byte extended messages, which require 6 and 13 zero crossings to transmit, respectively. Since there are 120 zero crossings per second with 60Hz AC power, a standard message takes 50ms to transmit and an extended message takes 108.33ms, with no additional hops. While Insteon's maximum theoretical bandwidth is 2880bps, in practice, devices typically use three hops and acknowledgements, which reduces the maximum bandwidth by 16× to 180bps. In addition to repeated messages, a sender that does not receive an acknowledgement within a specified timeout will retransmit a message up to five times. Thus, for noisy lines that require retransmissions, actual bandwidth may be much less than 180bps with three hops.

Insteon also uses 900MHz wireless communication to supplement the power line and increase its reliability and range, while also enabling it to cross phases in multi-phase power systems. The wireless communication mirrors the powerline communication in that all messages broadcast over the power line are also broadcast wirelessly at precisely the same time.

Finally, note that Insteon *does not* prevent multiple devices from sending different messages at the same time, although, since all devices act as repeaters, they inherently wait for in-transit messages to end before starting a new transmission. While repeating messages avoids flooding and collisions due to the synchronized retransmissions, Insteon has no collision avoidance mechanism, akin to exponential backoff in Ethernet, to prevent multiple devices from sending messages at the same time, thereby causing a collision and the loss of both messages. At high message rates, this lack of backoff combined with its static number of multiple hops and retransmissions results in repeated collisions, causing bandwidth to collapse. Insteon likely does not employ any collision avoidance mechanism because it was originally designed for low bandwidth control, where collisions are highly unlikely. In contrast, high bandwidth monitoring dramatically increases the likelihood of collisions by using near the maximum available bandwidth.

B. *Protocol Limitations*

Insteon sacrifices its already limited bandwidth to increase reliability. However, the available bandwidth affects both the maximum rate the OS can query each outlet's power usage as well as the percentage of asynchronous switch notifications and control commands lost due to bandwidth saturation and collisions. Setting the query rate for outlets presents a tradeoff: a rate too high will saturate the available bandwidth and result in the loss of either asynchronous switch notifications or control commands, while a rate too low will result in coarser and less accurate outlet power data. To understand this tradeoff, we experiment with our own smart home deployment by varying the rate of outlet queries, and then determining both the percentage of queries lost (Figure 2) and the percentage of switch notifications lost (Figure 3). For each data point, we issue outlet queries at the specified interarrival time on the x-axis for 10 minutes, while turning wall switches on and off 50 times, such that the time between toggling the switch is uniformly random between 0 and 20 seconds. We also perform a similar experiment in isolation in a separate building with no other devices attached to the power line.

Each power outlet query includes three standard Insteon messages and one extended message: a standard query message from the PLM to the outlet requesting the current power, an extended response message from the outlet to the PLM with the outlet's current average power usage, and a standard acknowledgement for each message. We use the default number of three hops for the initial message in all experiments. Note that altering the number of hops does not significantly alter the results, as we can only control the number of hops for the initial message sent from the PLM; the two acknowledgements and the extended message response always use three hops and originate from device firmware that we cannot change. Based on the Insteon protocol specification, each outlet query should take 4*(0.05+0.05+0.1083+0.05) = 1.0333 seconds, including the original message and the three additional hops.

Below, we use the specification to model the percentage of outlet queries we expect to receive, and the percentage of switch notifications we expect to lose for different query rates. We construct a simple model of the probability of losing a switch notification (S_{lose}) as a function of the interarrival time of outlet queries (T_i) and the length of an individual

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

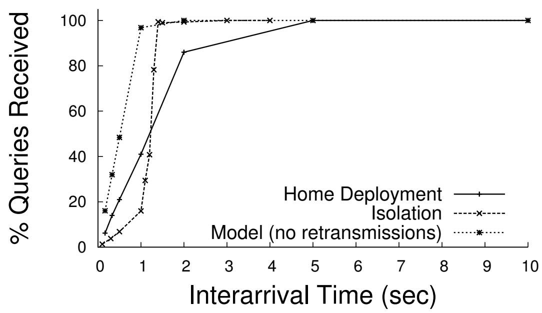

Fig. 2. Insteon does not support high outlet query rates.

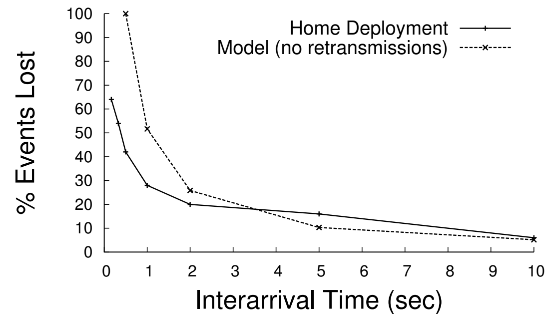

Fig. 3. Switch events may collide with outlet queries.

query ($T_q = 1.0333$). For simplicity, our model assumes that when transmissions collide, the transmission from the device physically closer to the PLM is successful, which is likely to be the case in practice. Our model also assumes one retransmission due to power line noise, and no additional retransmissions due to collisions. Thus, the model divides the length of an individual query (T_q) by two times the interarrival time between queries (T_i), where the two in the denominator approximates the effect of one extra retransmission.

$$S_{lose} = \frac{T_q}{2 * T_i} = \frac{0.5166}{T_i} \tag{1}$$

We also model the probability of receiving a query $Q_{receive}$ below. If we issue queries at an interval greater than the query length, then we expect to receive every query. For intervals less than the query length, we expect queries to collide.

$$Q_{receive} = \begin{cases} 1 & : T_i > 1.0333 \\ \frac{T_i}{T_q} = \frac{T_i}{1.0333} & : T_i < 1.0333 \end{cases}$$

Figure 2 shows that, as expected, issuing queries faster than the 1.0333 seconds it takes to complete them rapidly degrades network performance. In isolation, our results show an abrupt drop in the percentage of outlet queries received once the interarrival time hits the protocol's saturation point at 1.0333 seconds. The actual drop is more sudden than our model, since the model is in isolation and does not account for multiple retransmissions of a lost message, which immediately collapses the available bandwidth. Our deployment also shows more query losses than our model before the saturation point, which is likely due to i) additional losses from powerline noise due to other devices and ii) collisions with switch notifications and the resulting retransmissions. Figure 3 shows the percentage of switch notifications lost during the same experiment. We lose slightly fewer switch notifications after the saturation point at 1.0333 seconds than our model predicts. This indicates that, as expected for these infrequent switch notifications, multiple retransmissions of lost switch notifications (which our model does not capture) serve to slightly increase the percentage of successfully transmitted notifications.

C. Observations

Our results highlight the limitations of using Insteon for monitoring the energy usage of many devices. To illustrate, consider a simple approach to querying outlets that issues one query every 10 seconds to a new outlet in round-robin fashion. Thus, given N outlets in a building, this approach can query each outlet once every $10 * N$ seconds. Since our home deployment has thirty Insteon-enabled outlets, we are able to measure each outlet's power once every five minutes.

A five minute data resolution is not effective at monitoring the energy usage of most types of devices. As we show in prior work [8], [9], many non-linear electronic devices, such as LCD televisions, exhibit rapid and significant changes in power, e.g., >100W every second, when turned on. Other high-power resistive and inductive devices also exhibit complex patterns of power usage that change every second. Further, since many devices, such as a microwave or toaster, have operating times much less five minutes, this approach cannot detect their operation. In fact, the only devices this simple approach can accurately detect are low-power resistive devices, which exhibit highly stable power usage, that are left on for more than five minutes. The only prominent low-power resistive devices are incandescent light bulbs, which are slowly being phased out. Of course, additional outlets will further decrease the power data resolution that can be supported.

Even when employing such a low query rate, the probability of losing a switch notification or a control command is still near 5%. Since the building OS issues control commands, losing them does not present a significant issue, since it can recognize their loss at application-level and resend them. However, our simple approach to monitoring would have no way to detect a lost switch notification, so it will miss 5% of them at this query rate. Thus, our results motivate a more efficient approach to monitoring outlet power usage that judiciously controls the number of outlet queries.

Finally, while Insteon wall switches issue asynchronous notifications whenever an occupant toggles a switch, these switches generally do not have energy monitoring functions embedded into them. However, even if switched loads had energy monitoring functions, it would not be an efficient use of bandwidth to query switched loads for their energy usage. As we discuss, since switched loads by definition are capable of toggling between a discrete number of well-defined power states, it is more bandwidth-efficient to infer the power usage of switched loads by correlating changes in switch state, which are reported asynchronously, with changes in building power recorded by a centralized smart meter. Thus, we develop techniques for correlating state changes

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

with a building-wide "smart" power meter, which is typically available on modern buildings. These techniques are effective, assuming the available bandwidth is not saturated by the outlet monitoring above, and consumes minimal bandwidth.

Thus, we develop techniques to both i) accurately correlate switch events with changes in building-wide power to infer each switched loads power usage, and ii) efficiently poll outlet power usage to balance bandwidth and accuracy.

III. AutoMeter Deployment

Our goal is to develop a system that reliably controls and monitors each electrical device in a building at a high resolution. While our system enables demand-side energy management, as we discuss in Section I, our original motivation for designing it was much more practical: to enable research into building energy-efficiency by collecting and analyzing fine-grained energy data from each of a building's devices. We have publicly released much of the data we have collected as part of this effort as the Smart* home energy dataset [5], which has been downloaded over 2500 times to date and been used as the basis for a variety of different research projects, including benchmarking Non-Intrusive Load Monitoring algorithms [10], [11], comparing home energy usage across countries [12], and developing data privacy techniques [13], [14], [15]. Our particular approach to designing our monitoring system stems from previous (unsuccessful) attempts to reliably gather such fine-grained data at large scales using various wireless protocols, e.g., ZigBee, Z-Wave, WiFi, etc. These attempts led us to value Insteon's high reliability over the potential for higher bandwidth via wireless communication.

Here, we describe our AutoMeter prototype, which uses the switch learning and smart polling techniques described in the next section to enable high resolution energy monitoring with low-bandwidth PLC protocols. We deployed our prototype in a 3-bedroom, 2-bath house. The house has 34 wall switches, which control lights and exhaust fans. We replaced 30 of these mechanical wall switches with 20 Insteon SwitchLinc Relays and 10 Insteon SwitchLinc Dimmers. The 30 switches control 24 loads, since the house has two 4-way switches and two 3-way switches. As discussed below, we use an eGauge meter to monitor other switches. We use 30 Insteon iMeters to monitor plug loads in the home. The iMeters monitor all but 12 of the home's permanent plug loads. The unmonitored loads, e.g., night lights, electric toothbrushes, etc. consume little power in aggregate. We use an eGauge meter in the home's electrical panel to measure building-wide power consumption each second, which transmits data over the power line to a gateway server. Since eGauge is also able to monitor power for the home's circuits using additional CTs, we use it to monitor loads not connected to SwitchLincs or iMeters, including a clothes dryer, garbage disposal, dishwasher, basement lights, HRV duct heater, and the electrical components of the gas furnace, including an exhaust fan.

We implement AutoMeter's controller on a low-power and compact Raspberry Pi server. The server attaches to an Insteon PLM, which plugs into a standard outlet, via USB. Our software leverages the open-source plmtools package, which listens on the PLM's USB serial connection to send and receive binary data from the Insteon network. The package includes the `plmsend` and `plmcat` programs to send and receive raw Insteon messages using the PLM.

We wrote an Insteon monitoring daemon using these programs to 1) detect and record asychronous notifications broadcast on the power line whenever a switch turns on, off, or dims, and 2) monitor outlet power usage by using our smart polling techniques described in the next section. Since the switch notifications do not encode the dim percentage (from 0% to 100%) for dimmable switches, our daemon issues a status query to determine it whenever the dim level changes. Unfortunately, the commercial software that supports the iMeter is not designed for constant monitoring, since users must manually enter daily events that specify iMeter query times, which must be at least one minute apart. Thus, we reverse-engineered the iMeter protocol and modified the `insteon` command-line program in the plmtools package to support querying iMeter power, as well as sending asynchronous messages, i.e., not waiting for a reply from plmcat. The modifications allow us to issue iMeter queries at arbitrarily fast rates; we use this functionality for the experiments in the previous section.

Our fork of `plmtools` (`plmtools-imeter`), turns the iMeter into an easily scriptable meter which can be queried using a simple, one line Linux command. We have also extended plmtools in several other ways, such as adding more robust error handling (which is important given the potential for powerline packet collisions), human-readable descriptions of observed packets in real-time, and the decoupling of packet deliveries from receipts. The latter enhancement allows, for example, a single process to receive and process all incoming packets, while other processes asynchronously dispatch commands over the power line. This is useful when simultaneously listening for interrupts (such as from Insteon wall switches) and dispatching commands (such as iMeter queries).

AutoMeter's controller stores a timestamped record of each switch event and plug meter power usage in a SQLite database. We store the devices's name and the event timestamp, in addition to either the on-off-dim state between 0 and 100 or the plug power consumption. The controller also fetches the second-level eGauge data from eGauge's webserver, and stores it in the database. The controller uploads its SQLite database to a centralized off-site MySQL database each morning for long-term storage. We have released AutoMeter's underlying software as open-source for others to use, including the updated plmtools package.[1]

IV. High-Fidelity Monitoring and Control

AutoMeter combines two techniques—switch event correlation and smart polling—to perform accurate high resolution energy monitoring using low-bandwidth power line protocols. Switch event correlation infers the power usage of load switch events (which generally do not have embedded power readings) with readings from a building-wide smart meter. In doing so, switched loads can be accurately monitored without consuming any of the available bandwidth for explicit

[1] Code available at http://traces.cs.umass.edu/index.php/Smart/Tools

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

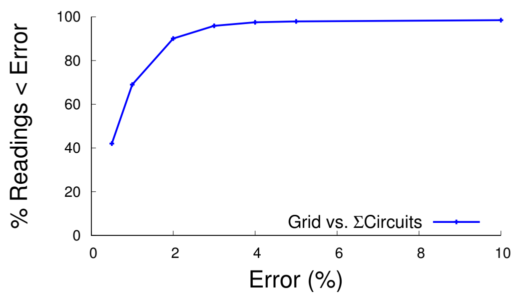

Fig. 4. Error between our deployment home's aggregate electricity data and the sum of all the individual circuits

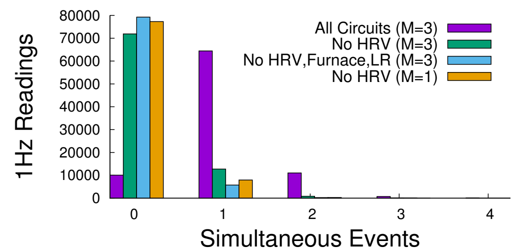

Fig. 5. Histogram of concurrent power events in our home.

polling. Thus, with the bandwidth limitations of HA power line protocols, inferring the power usage of switched loads is preferable, even if switched loads had energy monitoring functions built into them. For all general devices, which may not necessarily switch between discrete power states, smart polling intelligently queries the power usage of a (potentially large) set of devices to optimize the use of low-bandwidth power line communication. Below, we present and evaluate multiple approaches for each technique.

A. Switch Event Correlation

We first consider the problem of learning the power of switched loads. While individual switched loads (primarily lights) do not consume much power in isolation, they are the 12th largest load in our deployment home in aggregate. Further, if we remove summer-only loads, e.g., A/Cs, fans, then the switched loads are the 5th largest load. In addition, recent estimates attribute 5-10% of home energy and 20-50% of building use to lighting [16], which is the primary switched load. Finally, correlating loads that switch between discrete power states with building-wide power data is more efficient than querying their power usage using our smart polling techniques. Smart polling is better-suited for devices with variable power usage or that contain internal switches that cannot be controlled externally.

As described previously, switched loads generate asynchronous notifications when toggled or dimmed, but do not report actual energy usage. In principle, learning a switched load's power consumption should be straightforward: simply record the change in the building-wide power data whenever a switch changes state. However, learning switch power is complicated by two issues: 1) power sensing and timing errors may occur in the building meter, and 2) multiple power events may occur within the building meter's monitoring granularity. In particular, we observe frequent timing errors that delay new power readings due to communication problems. To illustrate, Figure 4 shows the error between the aggregate energy usage for our home (sensed via instrumenting the incoming power lines) and the sum of the of the power usage of all the home's circuits (sensed separately via instrumenting each individual circuit). The graph quantifies the extent of the relative error in the meter. For example, in this case, nearly 10% of the per-second readings for the entire home and the sum of the circuits are more than 2% of each other, while nearly 1% of readings are more than 4% of each other. These errors occur for two primary reasons. First, the sensors themselves have a 1% error, as rated by the manufacturer. Second, since the sensors are physically different, the timing of processing and releasing the readings may differ.

While we also observe multiple simultaneous power events, monitoring building-wide power every second mitigates their impact. For example, Figure 5 shows a histogram of the number of per-second readings that fall within a concurrent power event across all circuits our home on a single day. In this case, we define an event as a change in power greater than 10W, with an associated margin M of either 0 seconds or 3 seconds that determines the duration (or 'length') of an event. If a change occurs at time T and the margin is M seconds, then any other change in power is concurrent if it occurs between $T + M$ and $T - M$. Figure 5 shows that the vast majority of per-second readings, e.g., $x = 0$ or $x = 1$, are not part of concurrent events (with a threshold of 10W and a margin of 3 seconds). While the number of readings that fall within concurrent events ($x \geq 2$) is approximately 10,000 for $M = 3$, most of these are caused by a small number of highly variable loads. If we remove the HRV (the most variable load), the number of readings falls by more than 10$\times$ to approximately 900, while removing the furnace and living room loads results in an additional 3$\times$ reduction to roughly 300 readings. Thus, simultaneous events are a more significant issue for coarser monitoring intervals.

Below, we discuss both *proactive* and *reactive* techniques to learn switch power consumption that is robust to both coarse data and data errors. As noted above, while simultaneous events are not a significant issue, our techniques must take into account the likelihood of sensor error, particularly our reactive technique.

Proactive Switch Learning. In our proactive approach, we write a simple program to remotely toggle each switch one by one from the Insteon PLM, and observe the change in building power 2 seconds before and 10 seconds after toggling. For the experiment, we turn off most loads in the home to decrease simultaneous power events from other devices and data errors, which are proportional to the home's total load. Table I reports the power for each switch, as well as the switched load's rated power, and shows the approach is over 93% accurate on average across all loads. While the proactive approach is accurate, not all buildings will be able to shutdown most loads to reduce errors and cycle through every load in order to determine power usage. Thus, we explore a reactive approach that learns power usage over time based on collected data.

Reactive Switch Learning. The reactive approach also computes the change in building-wide power whenever a switch changes state; again, we use 2 seconds before and 10 seconds

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

Name	*Proactive*	*Reactive (#)*	*Actual*
kitchen:lights:dim	257W	290W (62)	260W
kitchen:sink	67W	69W (15)	65W
kitchen:lights1:dim	190W	192W (13)	195W
hall:lights1:dim	193W	39W (5)	195W
guest:lights:dim	255W	279W (10)	260W
guestbath:fan	51W	147W (36)	50W
guestbath:overheadlight	101W	100W (107)	100W
guestbath:sinklight	57W	60W (55)	60W
livingroom:dininglights:dim	128W	38W (25)	130W
livingroom:firelights:dim	148W	925W (8)	130W
livingroom:sideporch:dim	121W	850W (7)	130W
livingroom:lights1:dim	255W	361W (9)	260W
livingroom:lamp	17W	18W (61)	18W
frontporch:light	12W	185W (16)	20W
stairs:light1	72W	70W (20)	65W
masterbath:overheadlight	102W	100W (147)	100W
masterbath:fan	54W	110W (114)	50W
masterbath:sinklight	58W	59W (313)	60W
master:lights:dim	256W	26W (19)	260W
master:closet:a	12W	20W (57)	20W
master:closet:b	12W	9W (15)	20W
bedroom:lights:dim	254W	258W (21)	260W
bedroom:maincloset	18W	19W (14)	20W
bedroom:linencloset	22W	20W (38)	20W
bedroom:closet	22W	22W (35)	20W

TABLE I
TABLE OF SWITCH POWER USING PROACTIVE AND REACTIVE LEARNING VERSUS ACTUAL POWER CONSUMPTION.

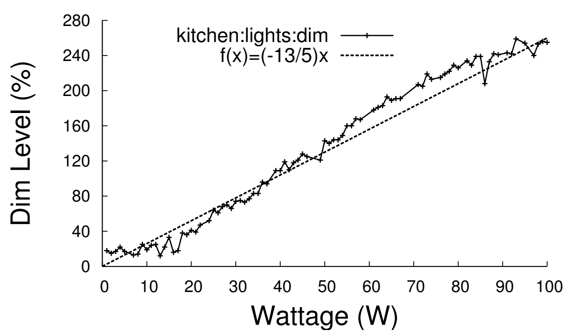

Fig. 6. Power usage is a linear function of a light's dim level.

after the change. We normalize the power step by a switch's dim level: if the dim level is 50% then we multiply the power step by two. We confirmed the linear relationship between power and dim level by recording the change in building-wide power as we vary the dim level from 1% to 100%, as shown in Figure 6. Due to power and timing errors or other loads changing their power consumption, the power step over the interval will not always correspond to a switch's power consumption. However, our premise is that over long periods with many events, the plurality of power steps will be near the actual power consumption of the switch, since building loads alter their power states at human time-scales, e.g., minutes to hours. Thus, our reactive approach groups every observed power step for each state change for a given switch into bins, e.g., 5-15W, 15-25W, 25W-35W, etc. We then select the bin with the most events, average its values, and record that value as the switch's power consumption.

Table I shows the results of the reactive approach for 2 weeks of data, as well as the total number of switch events in parenthesis.[2] We find that the reactive approach is accurate for switches with many events over the 2 week period, and less accurate for rarely-used switches. Interestingly, the approach is not accurate for the exhaust fan in each bathroom, since they nearly always change state at the same time as a 100W overhead light or a 60W sinklight. We are currently augmenting the technique to identify these correlated switches. Figure 7 shows a histogram of the number of events in each bin for three of the thirty switches. The data demonstrates how power and timing errors in the building data, as well as simultaneous power events, cause a wide range of power values for each switch event, which complicates learning switch power.

[2]The table has less than 30 switches, since we only learn a single value for each set of multi-way switches.

B. Smart Energy Polling

For devices other than switches, interrupts cannot be used, as the notion of an 'event' is difficult to define for any general (non-switch) device. Instead, the simple types of energy meters and outlets popular in HA deployments (such as the Insteon iMeter Solo) must be **polled** in order to monitor energy consumption, resulting in many devices competing for limited global bandwidth. Since many widely-used PLC networks such as Insteon have extremely limited bandwidth, we design several smart polling techniques to accurately monitor many devices without saturating the network. Our system is capable of several different types of polling approaches, which we describe in detail below and then evaluate in simulation to highlight the tradeoffs of using each approach.

Round-robin polling. The simplest approach is to continuously query monitored outlets in a round-robin fashion at a static query rate. While straightforward, this approach suffers the most from bandwidth limitations, since devices that rarely change state (such as lights) are polled at the same rate as highly variable devices (such as a washing machine).

Frequency-based polling. The second technique we consider, which we call frequency-based polling, is a slight modification of round-robin polling in which devices are polled at different rates based on their activity level or priority. Highly active or important devices are polled more frequently to more accurately capture their behavior, while more static (or less important) devices are polled less frequently. Here, we define a device's level of 'importance" as its frequency of energy state changes (i.e., power increases or decreases) that the device exhibits over a typical day. Given the state change frequency for each outlet, our system polls each outlet at a rate proportional to its frequency, scaled such that the system continuously polls at a fixed global query rate (as in round-robin). Note that other device orderings besides state change frequency are also possible, such as the maximum power of a device (i.e., prioritizing high-power devices such as heaters), or a custom ordering favoring specific devices (e.g., for a particular application).

Event-driven polling. Our third technique makes use of a centralized "smart" meter that monitors power for an entire building at high resolution. Such smart meters are widely available commercially, and are increasingly being installed by utilities: in 2011, nearly 500 utilities in the U.S. had collectively installed more than 37 million smart meters [17]. We use an eGauge meter installed in the electrical panel, which

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

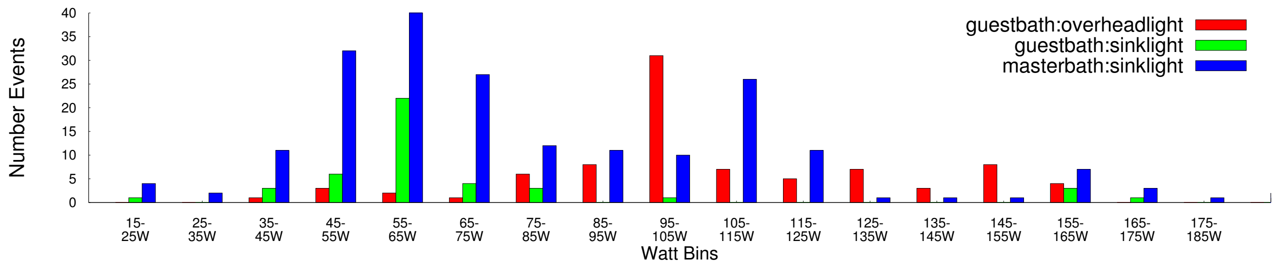

Fig. 7. Histogram of events for each 10W power bin for selected switches using the reactive learning technique.

measures building-wide average power usage as well as per-phase average voltage and frequency each second [18].

Given such a centralized meter, our third technique, which we call event-driven polling, analyzes live data from the smart meter to determine when to poll outlets on-demand as energy events occur. Since the meter records aggregate energy, an energy event stemming from any individual device will be reflected in the aggregate data. We assume, of course, that the centralized meter can be queried at the same rate as the meters (and in tandem). When an energy event is detected, the remaining task is to attribute the change to a specific device. To do so, we conduct a round of frequency-based polling of individual outlets, and stop once we find the matching energy change. As a result, in most cases, only a small subset of devices must be polled to identify the source of an event.

C. Smart Polling Evaluation

To evaluate the three polling approaches, we conduct a simulation study using real-world, device-level data gathered from our home deployment. Our sample dataset consists of 3 days of 1 Hz data from 22 distinct outlets. We replay this data while simulating polling at a variable rate ranging from 0.25 to 5 seconds. The polling interval also determines the duration of a poll (i.e., the delay between issuing a poll and receiving the response). For the round-robin and frequency-based approaches, we assume that polls are issued continuously as quickly as possible, given the polling interval.

Energy Breakdown. We first consider a classical monitoring application – determining the energy used by each device over the course of a day (e.g., for providing an energy breakdown). Figure 8 shows the average percentage difference between the energy use of each device in the monitored trace and its true usage over the course of a day (i.e., its usage error).

We see that in all cases (even with a simple round-robin approach), the usage error is quite low, remaining consistently under 10% for polling intervals up to roughly 2 seconds and only increasing slowly for slower polling intervals. This result is an encouraging (albeit simplistic) indication that even a highly bandwidth constrained network can provide an accurate coarse-grained breakdown of device energy usage.

The event-driven approach does slightly worse than the others, which is likely due to possible errors in assigning energy events to devices (e.g., a 30W change that is assigned to the wrong device, thus contributing to the usage error of the correct device that was not polled). We also evaluated a slight variation on our frequency-based polling approach in

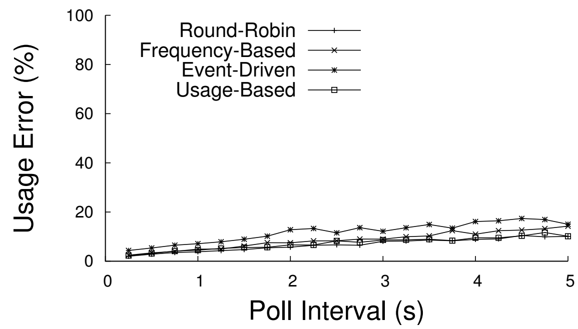

Fig. 8. Even low-frequency polling results in accurate per-device energy usage information.

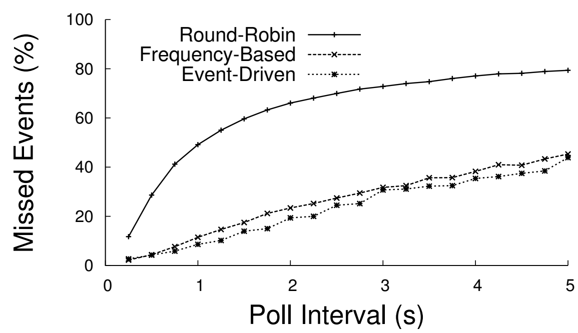

Fig. 9. Frequency- and event-driven polling capture more energy events than basic round-robin polling.

which the order of polls is based on the typical aggregate usage of the device rather than its number of state changes, as depicted by the "usage-based" polling label in Figure 8. While we would naturally expect this approach to improve on simple round-robin polling, the measured improvement is minimal.

Energy Event Detection. We next consider capturing 'events' in the monitored traces, where an event is defined as a 1-second power change of at least 10W. Events provide a useful measure of fine-grained monitoring accuracy (as opposed to a coarse-grained energy breakdown), and are important for a range of applications, including occupancy monitoring [19], [20]. For each polling approach, the percentage of events missed is shown in Figure 9 as the polling interval is varied.

Here we immediately see the limitation of the simplistic round-robin approach, which misses significantly more events (at least 2X) than the frequency-based or event-driven approaches. The poor performance of round-robin is largely

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

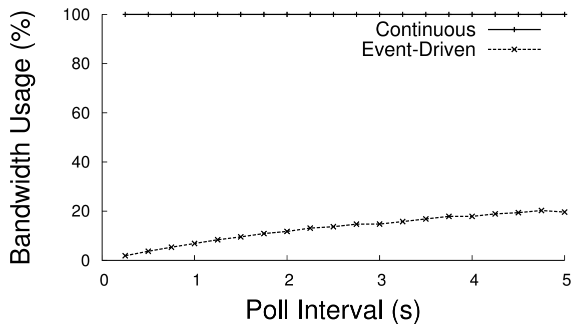

Fig. 10. Event-driven polling drastically decreases the bandwidth required to monitor a set of outlets.

due to a small number of highly-variable devices, such as a TV and washing machine, which exhibit numerous and rapid power changes when active. Frequency-based and event-driven polling compensate for these devices by polling them much more rapidly, which ensures that most events continue to be captured. While event-driven polling marginally outperforms the frequency-based approach, the strong performance of the frequency-based approach is significant since this approach does *not* rely on the presence of a building-wide smart meter.

Bandwidth usage. The key benefit of event-driven polling is that due to its on-demand nature, polling may be stopped completely when no events are occurring. To demonstrate this, Figure 10 shows the aggregate bandwidth consumption over the course of the trace period using the event-driven approach. While round-robin and frequency-based polling both poll continuously (effectively using 100% of available bandwidth assuming a minimized polling interval), the event-driven polling employs far fewer polls — less than 20% — to achieve the high accuracy seen in Figure 9. Thus, we expect the event-driven approach to scale well to large number of devices, and also preserves bandwidth for other messages (e.g., control messages) that may otherwise collide with polling messages. This result highlights the utility of combining a building-wide meter with intelligent outlet-level polling.

Note that the combination of event-initiated polling rounds and the frequency order within a round serves to opportunistically use bandwidth when needed to capture most events. For example, some types of devices issue almost continuous events when active, and thus require back-to-back polls of the same device to avoid missing many events. To illustrate, we consider a different time period in which a heat recovery ventilator (a highly oscillating load) is active for a significant period and compare the round-robin approach with the event-driven approach. The bandwidth usage and percentage of missed events for both approaches is shown in Figure 11.

Unlike in Figure 10, here, the event-driven approach does not save a significant amount of bandwidth, using roughly 95% of the total available bandwidth to repeatedly query the HRV. However, in doing so, the event-driven approach maintains less than 20% event loss up to a 2 second poll interval. In contrast, even with only a 1 second poll interval, the round-robin approach loses over 80% of all events (substantially worse than in Figure 9 in which the HRV was not active).

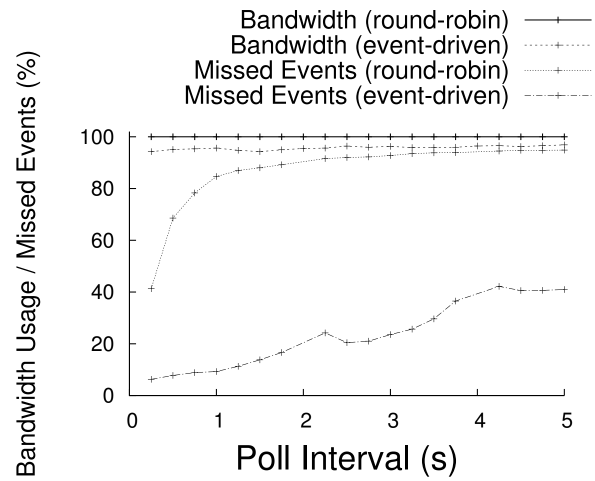

Fig. 11. Event-driven polling effectively utilizes bandwidth to monitor highly variable devices.

D. Scalability and Commercial Buildings

The evaluation presented above focuses on a home environment, in which the number of devices is relatively small compared to what would be found in a large commercial building. However, our techniques should still be useful in such an environment. The primary scalability issue in switch event correlation is the possibility of simultaneous events – however, as demonstrated in Section IV-A, simultaneous events are quite rare in practice. For very large buildings with many occupants, devices, and (potentially simultaneous) events, a power meter providing finer-grained data (e.g., floor-level or circuit-level readings, which are provided by some meters such as the eGauge) could serve to mitigate most of these issues by isolating events within switch subsets.

For smart polling, the fixed amount of bandwidth in a PLC network like Insteon limits the number of devices that could be feasibly monitored, even employing the techniques discussed. However, a subset of devices could still be effectively monitored in a large building by simply polling those devices and ignoring all others. Doing so would result in a reduction in efficiency for event-driven polling, since events caused by non-monitored devices would still trigger polling rounds, but the impact of these extra polling rounds would be limited by the number of monitored devices. Hence, monitoring a subset of the most important devices (e.g., larger, energy-intensive loads) should remain feasible. As with switch correlation, a finer-grained smart meter could also solve these challenges.

E. Summary

In AutoMeter, we demonstrate how to monitor a large set of devices at high resolution using only a low-bandwidth PLC network. In particular, our monitoring infrastructure makes use of switch event correlation to monitor switched loads without any polling at all, while other devices are queried using "smart polling" techniques that minimize the number of messages required without significantly sacrificing accuracy. We demonstrate that our event correlation is over 93% accurate

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2017.2703916

in our home at learning the consumption of wall switches, while smart polling decreases the bandwidth used to monitor a set of devices by over 80% and reduces the number of energy events lost by over 50%. While our specific AutoMeter deployment is built using Insteon components in a home environment, we expect our approach to work well in other low-bandwidth PLC environments as well.

V. Related Work

We know of no work that exploits existing low-bandwidth HA power line protocols for high fidelity energy monitoring. Prior work has largely focused on the higher-bandwidth power line protocols, such as HomePlug, which are not as appropriate for the relatively simple tasks of energy monitoring and device control [21], [22], [23]. High-bandwidth power line protocols are more appropriate for serving as part of the network backbone [23] and transmitting high-bandwidth data, such as video. The advantage of low-bandwidth protocols, such as X10 and Insteon, is that their cost is much lower [22], which is important when embedding simple energy monitoring and control into the hundreds of switches, outlets, and devices present in modern homes.

There is much prior work on both developing wireless energy monitoring sensors [24], [4] and deploying large networks of wireless energy monitoring sensors [3]. Prior work has also addressed the challenges of deploying and maintaining large wireless sensor networks (for energy monitoring and otherwise) [2]. Our work is complementary to this prior work on wireless monitoring. In part, our work serves to fill a void, since researchers have not focused on exploiting low-bandwidth power line protocols for simple energy monitoring and control. Further work is necessary to compare the advantages and disadvantages of using various wireless protocols versus low-bandwidth power line protocols in different environments and to determine the most appropriate implementation technique.

VI. Concluding Remarks

This paper discusses the challenges of enabling energy monitoring and control in smart buildings using low-bandwidth power line communication protocols, such as Insteon. In particular, we describe our own deployment and architecture for whole-house monitoring and control using Insteon, and empirically quantify the limitations of the Insteon protocol. We then present several techniques for switch event correlation and smart polling used in our deployment, which address the limitations of low-bandwidth PLC with focus on correlating switch events with a whole-house power meter and polling energy usage on highly active devices. Finally, we evaluate our techniques using data collected from our real-world home. In particular, we show that our smart polling techniques outperform naive polling approaches in both monitoring accuracy (increased by 2X) and total bandwidth consumption (decreased by 80%), thereby improving the feasibility and scalability of low-cost, low-bandwidth monitoring networks.

Acknowledgements. Portions of this article appeared in previous papers presented at the 2011 ACM Workshop on Embedded Sensing Systems For Energy-Efficiency In Buildings [25] and the 2012 Workshop on Data Mining Applications in Sustainability [5].

References

[1] J. Taneja, D. Culler, and P. Dutta, "Towards Cooperative Grids: Sensor/Actuator Networks for Renewables Integration," in *SmartGridComm*, 2010.

[2] T. Hnat, V. Srinivasan, J. Lu, T. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse, "The Hitchhiker's Guide to Successful Residential Sensing Deployments," in *SenSys*, November 2011.

[3] X. Jiang, M. V. Ly, J. Taneja, P. Dutta, and D. Culler, "Experiences with a High-fidelity Building Energy Auditing Network," in *SenSys*, November 2009.

[4] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler, "Design and Implementation of a High-Fidelity AC Metering Network," in *IPSN*, 2009.

[5] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes," in *SustKDD*, August 2012.

[6] C. Dixon, R. Mahajan, S. Agarwal, A. J. Brush, B. Lee, S. Saroiu, and V. Bahl, "An Operating System for the Home," in *NSDI*, April 2012.

[7] S. Dawson-Haggerty, A. Krioukov, J. Taneja, S. Karandikar, G. Fierro, N. Kitaev, and D. Culler, "BOSS: Building Operating System Services," in *NSDI*, April 2013.

[8] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, "Empirical Characterization and Modeling of Electrical Loads in Smart Homes," in *IGCC*, June 2013.

[9] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, "Powerplay: Creating virtual power meters through online load tracking," in *BuildSys*, November 2014.

[10] N. Batra, J. Kelly, O. Parsons, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava, "NILMTK: An Open Source Toolkit for Non-Intrusive Load Monitoring," in *eEnergy*, June 2014.

[11] L. Ong, M. Berges, and H. Noh, "Exploring Sequential and Association Rule Mining for Pattern-based Energy Demand Characterization," in *BuildSys*, November 2013.

[12] N. Batra, M. Gulati, A. Singh, and M. Srivastava, "It's Different: Insights into Home Energy Consumption in India," in *BuildSys*, November 2013.

[13] X. Liao, D. Formby, C. Day, and R. Beyah, "Di-PriDA: A Privacy-preserving Meter Querying System for Smart Grid Load Balancing," in *IEEE Security and Privacy*, May 2014.

[14] I. Leontiadis, R. Molva, and M. Onen, "Privacy Preserving Statistics in the Smart Grid," in *DASEC*, June 2014.

[15] M. Jelasity and K. Birman, "Distributional Differential Privacy for Large-Scale Smart Metering," in *IH&MMSec*, June 2014.

[16] "Energy Use for Lighting." http://www.dmme.virginia.gov/DE/ConsumerInfo/HandbookLighting.pdf.

[17] "U.S. Energy Information Administration, Frequently Asked Questions, How Many Smart Meters are Installed in the U.S. and who has them?." http://www.eia.gov/tools/faqs/faq.cfm?id=108&t=3, 2011.

[18] "eGauge Energy Monitoring Solutions." http://www.egauge.net/, 2012.

[19] W. Kleiminger, C. Beckel, T. Staake, and S. Santini, "Occupancy Detection from Electricity Consumption Data," in *BuildSys*, November 2013.

[20] D. Chen, S. Barker, A. Subbaswamy, D. Irwin, and P. Shenoy, "Non-Intrusive Occupancy Monitoring using Smart Meters," in *BuildSys*, November 2013.

[21] N. Roy, D. Kleinschmidt, J. Taylor, and B. Shirazi, "Performance of the Latest Generation Powerline Networking for Green Building Applications," in *BuildSys*, November 2013.

[22] P. Pannuto and P. Dutta, "Exploring Powerline Networking for the Smart Building," in *IP+SN*, April 2011.

[23] R. Murty, J. Padhye, R. Chandra, A. Chowdhury, and M. Welsh, "Characterizing the End-to-End Performance of Indoor Powerline Networks," in *Harvard University Technical Report*, 2008.

[24] S. Lanzisera, "The "Other" Energy in Buildings: Wireless Power Metering of Plug-in Devices." Environment Energy Technologies Division Seminar, Lawrence Berkeley National Labs, June 17 2011.

[25] D. Irwin, A. Wu, S. Barker, A. Mishra, J. Albrecht, and P. Shenoy, "Exploiting Home Automation Protocols for Load Monitoring in Smart Buildings," in *BuildSys*, November 2011.

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.