

Nonintrusive Load-Shed Verification

An approach in which residential power meters monitor electricity use and analyze load enables electricity service providers to verify that registered appliances conform to their load-shed instructions.

Demand response, a cornerstone of smart-grid technology, lets consumers participate directly in energy markets by limiting their energy use during periods of emergency or peak demand. In a direct-control demand-response approach, an electricity service provider (ESP) offers consumers discounts or other incentives if they agree to let the ESP send load-shed instructions (LSIs) to specified appliances. For instance, an ESP might adjust the set points on an air conditioner's thermostat. Direct control can save consumers money and provide

ESPs with valuable tools for controlling energy generation costs and grid stability. But these benefits depend on the LSIs producing the expected response from appliances.

Load-shed verification (LSV) can improve reliability and eliminate freeloaders who accept incentives without implementing direct controls.

However, this generates many trust challenges because the consumer owns and operates the appliance and because effective demand response depends on the integrity of the appliances' responses to LSIs.

To address these challenges, we've implemented an algorithm based on a *nonintrusive*

load monitoring (NILM) learning phase that runs during an initialization period at the ESP. The result is a distributed NILM algorithm—a *nonintrusive load-shed verification* (NILSV) algorithm deployed on the residential meter. We built a prototype and conducted experiments in a residence to illustrate NILSV's promise along with some of its challenges.

LSV Challenges

Developing a trust model for direct control of residential consumer appliances is a well-recognized challenge.^{1,2} Designing and deploying approaches wherein the ESP has a secure, remote, and tamper-resistant verification agent in each appliance will be difficult owing to the appliances' diversity. To fully address the trust issue, a solution must secure not only the communications path and the load controller hosting the trusted agent but also the load's control connections. Securing these connections for the many consumer appliances to which control must be retrofit—such as HVAC (heating, ventilation, and air conditioning) systems, water heaters, and pool pumps—is largely impractical.

We can simplify this problem by using unidirectional authentication of control messages from the ESP to the appliance. This method enables several communication options, including public webpage messages (such as RSS feeds), FM radio broadcasts, and local broadcasts from

David C. Bergman, Dong Jin,
Joshua P. Juen, Naoki Tanaka,
and Carl A. Gunter
*University of Illinois
at Urbana-Champaign*

Andrew K. Wright
N-Dimension Solutions

Related Work in Nonintrusive Load Monitoring

At one time, electric power meters displayed analog data, which a visiting meter reader read periodically. Wireless communications reduced this burden by allowing automated meter reading (AMR) at first from a truck that drove near the meter, and later from a pole-top unit. Meters' wireless communication and computing and sensor capabilities will significantly improve with smart-grid efforts. These *advanced meter infrastructure* (AMI) capabilities can improve theft detection, outage management, and power-quality assessment, as well as AMR.

Demand Response

A significant AMI application is demand response, in which meters collect interval readings, transmit signals to appliances, and provide usage data to consumer portals to support power-use patterns that reduce electricity costs. These costs vary significantly over time because more expensive techniques must supplement a baseline of cheap generation techniques when demand peaks arise. If demand shifts from peak periods to off-peak periods, considerable savings are possible.

One strategy gives indirect control to a consumer. The electricity service provider (ESP) assigns a price for electricity in a given time interval; the customer uses this information to make decisions about power use. In direct-control strategies, the ESP sends signals to consumer appliances to alter their use, typically by limiting use during peak demand periods. Each approach has advantages and disadvantages. Indirect control puts consumers in charge but makes them responsible for demand-response actions. Direct control aids consumers by automating demand response but might provide more or less response than they want.

NILM

George Hart introduced the first NILM algorithm,¹ which determined each load's state by parsing the real power graph to find step transitions, clustering transitions by similar power changes, and classifying the clusters as individual appliances. The method assumed previous knowledge of the individual monitored appliances and an interactive training phase on the collection of appliances. The algorithm's accuracy was limited owing to difficulties detecting multistate appliances and differentiating loads

with similar power signatures.

Researchers have refined this technique and improved on it by using complex power, state tables, frequency analyses, and more sophisticated learning techniques, such as genetic algorithms.²⁻⁶ These techniques have allowed commercial NILM to classify small loads accurately without prior training or knowledge of the operating environment.

Unfortunately, these advanced NILM techniques involve high computational requirements and detailed sensor data. For example, the genetic algorithms that eliminate the interactive training phase take 10 minutes to run on a 3-GHz Pentium 4 machine, whereas the typical AMI meter runs under 100 MHz. Advanced techniques such as frequency analysis require meter readings at the millisecond scale,⁶ whereas AMI meters typically provide no better than one reading per second. Moreover, NILM techniques work best on large datasets, whereas AMI meters have little memory and AMI networks have limited bandwidth.

REFERENCES

1. G.W. Hart, "Nonintrusive Appliance Load Monitoring," *Proc. IEEE*, vol. 80, no. 12, 1992, pp. 1870-1891.
2. M.L. Marceau and R. Zmeureanu, "Nonintrusive Load Disaggregation Computer Program to Estimate the Energy Consumption of Major End Uses in Residential Buildings," *Energy Conversion and Management*, vol. 41, no. 13, 2000, pp. 1389-1403.
3. M. Baranski and J. Voss, "Genetic Algorithm for Pattern Detection in NIALM Systems," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, vol. 4, IEEE Press, 2004, pp. 3462-3468.
4. M. Baranski and J. Voss, "Detecting Patterns of Appliances from Total Load Data Using a Dynamic Programming Approach," *Proc. 4th IEEE Int'l Conf. Data Mining (ICDM 04)*, IEEE CS Press, 2004, pp. 327-330.
5. K.H. Ting et al., "A Taxonomy of Load Signatures for Single-Phase Electric Appliances," 2005; www.lucente.us/pubs/2005-IEEE-PESC-preprint.pdf.
6. K.D. Lee et al., "Estimation of Variable-Speed-Drive Power Consumption from Harmonic Content," *IEEE Trans. Energy Conversion*, vol. 20, no. 3, 2005, pp. 566-574.

residential meters. We can attain LSV by using the residential power meter to monitor electrical use and by analyzing load changes. When the meter detects an appropriate load change, it sends an LSV response message. Generally, residential users don't have a sufficient energy-management system to respond to an LSI calling for a specific load-shed level.

Therefore, we must base direct control on individual appliances' instructions.

A promising approach to this problem uses NILM technology (see the "Related Work in Nonintrusive Load Monitoring" sidebar).³ Nonintrusive load monitoring (NILM) is a direct-control approach that takes aggregate electric-metering data, such as the data

in Figure 1, and extracts a collection of appliance load profiles and use periods for these appliances that would account for the data. For instance, if an appliance is known to draw 1,000 Watts, a jump of this amount might indicate that the appliance was turned on, whereas a drop of this amount might mean it was turned off.

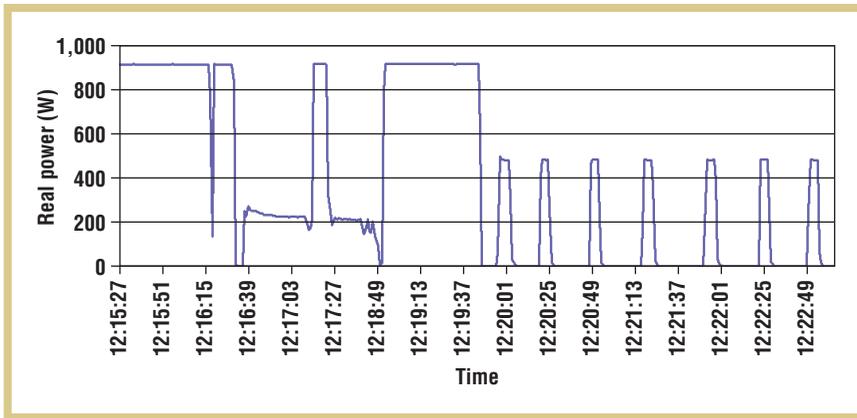


Figure 1. A typical power graph. Nonintrusive load monitoring takes aggregate electric-metering data and extracts a collection of appliance load profiles and use periods for these appliances that would account for the data.

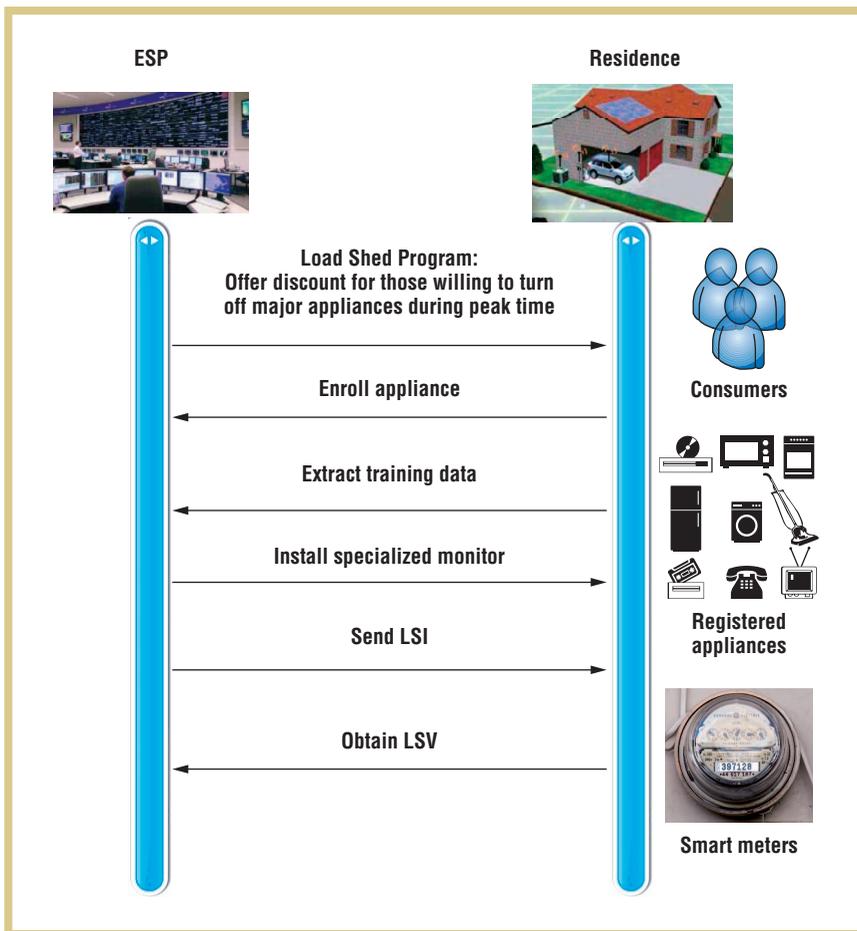


Figure 2. The nonintrusive load-shed verification (NILSV) process. Consumers enroll in load-shed programs and allow electricity service providers (ESPs) to install specialized monitors at home. During load-shed operation, the ESPs instruct enrolled appliances to switch off, and success is verified at smart meters.

household an LSI to turn off a 1,000-W appliance. In a predefined time frame, the meter will respond with an LSV indicating the appliance’s transition. The correct behavior is on \Rightarrow off or off \Rightarrow off. We call this NILSV. NILSV’s key challenge is handling the large amounts of detailed data and complex calculations that aren’t naively suited to typical residential meters’ low bandwidth, computing, and sensor capabilities.

NILSV Requirements

Assume ESP *E* offers an incentive program for the direct control of appliance type *A*. *E* has information about *A*’s load profile behavior through its own testing and data from *A*’s vendor. Customer *C* contacts *E*, asking to be enrolled in the program in exchange for direct control of a specific appliance *a* of type *A* in *C*’s home. *C* or *E* configures *a* to receive appropriate LSIs from *E* while operating in *C*’s residence.

We don’t concern ourselves with exactly how this happens, but we don’t assume that this configuration alone is sufficient to verify that *a* receives and reacts to the LSIs. This is the NILSV problem. How do we implement an LSV protocol to verify whether *a* responds correctly to LSIs, using only aggregate meter data from *C*’s residence that a typical residential meter senses and a typical advanced meter infrastructure (AMI) network reports?

We must first refine the NISLV problem. For instance, we must clarify the “typical” residential meter and AMI network’s technical characteristics. If the meter captures complex power at millisecond intervals and transmits it over a broadband network to a super-computer, the problem is probably a

Ideally, when residential consumers register an appliance for direct control, that appliance can receive LSIs but isn’t required to provide confirmation of

compliance. NILM conducted through the residential meter can confirm that the appliance has acted on the LSIs. For instance, suppose an ESP sends a

Figure 3. NILSV architecture. The system contains a centralized, computationally intensive module at the ESP and a distributed, resource-limited module at smart meter.

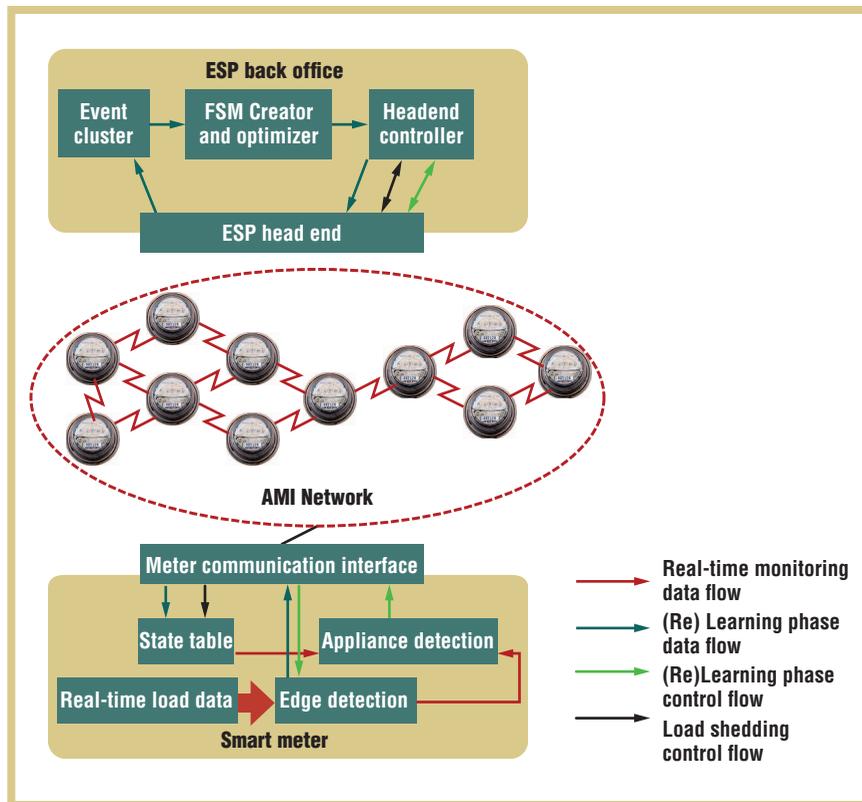
special instance of NILM. However, if we assume that the meter is a processor running at a few hundred megahertz and possessing no more than a few kilobytes of memory, and that the AMI network can transmit only a few hundreds of kilobits of data each second, then existing NILM techniques can't straightforwardly address the problem.

For instance, we assume the processor is an Atmel AT32UC3A (www.atmel.com/dyn/resources/prod_documents/32058S.pdf) running an operating system that accommodates software and data updates via the AMI network. We also assume that the AMI network is comparable to ZigBee, with a bandwidth of 250 kbps to the backhaul point. So, the meters' combined communication with a backhaul point can't exceed a bandwidth of 250 kbps. We assume that the meter can obtain real power readings on a per-second basis. For this purpose, we use TED 1000 meters from Energy Inc. We assume that the ESP has substantial computing power and memory; however, a given program might contain millions of meters and appliances.

Ideally, the NILSV protocol should provide reasonable assurance of the response rate it achieves if it sends out an LSI. For instance, NILSV testing might show that 90 percent of appliances in operation will properly respond to an LSI. The NILSV should also help identify appliances that don't respond to LSIs because they're unintentionally misconfigured or intentionally ignoring LSIs (freeloaders).

Our Approach

Our NILSV approach consists of a learning phase that compresses data from the meter and sends it via the



AMI network for detailed analysis on a back-end system and a monitoring phase, which uses this analysis to design a simple state-based agent that performs LSV on the meter. Figure 2 shows this process; Figure 3 describes the NILSV architecture.

The training phase is responsible for identifying the appliances in the home. We collect learning data by installing a monitor in the consumer's smart meter to scan for discrete step events—a computationally inexpensive process. In our tests, this reduced the necessary training data the meter needed to report by about 99 percent, allowing transmission over the low-bandwidth AMI network. When training data reaches the ESP's back end, the learning algorithm first clusters events based on similar power usage. The learning algorithm inputs the clustering data into a module that performs advanced analysis using a genetic algorithm to produce an optimized finite-state machine (FSM). The FSM is represented as a static table, which is

then deployed in the meter at the consumer's residence.

During the monitoring phase, the meter takes readings and conducts real-time edge detection to create a dynamic table. Upon encountering an edge, the meter uses the static table and a simple matching algorithm to update which appliances are on and off in the house and record the results in a dynamic table. By keeping the dynamic table up to date, the meter will always be prepared to respond to LSV requests from the back end for the enrolled appliance. The goal is to respond to LSV requests for a specific appliance rather than profile all appliances in use. Because appliances meriting direct control typically have larger loads, the state tables can ignore appliances below an energy threshold, reducing them to background noise.

In addition to updating the dynamic table, the meter also uses the edge- and appliance-detection modules to identify when to reenter the learning phase. The learning phase will be conducted

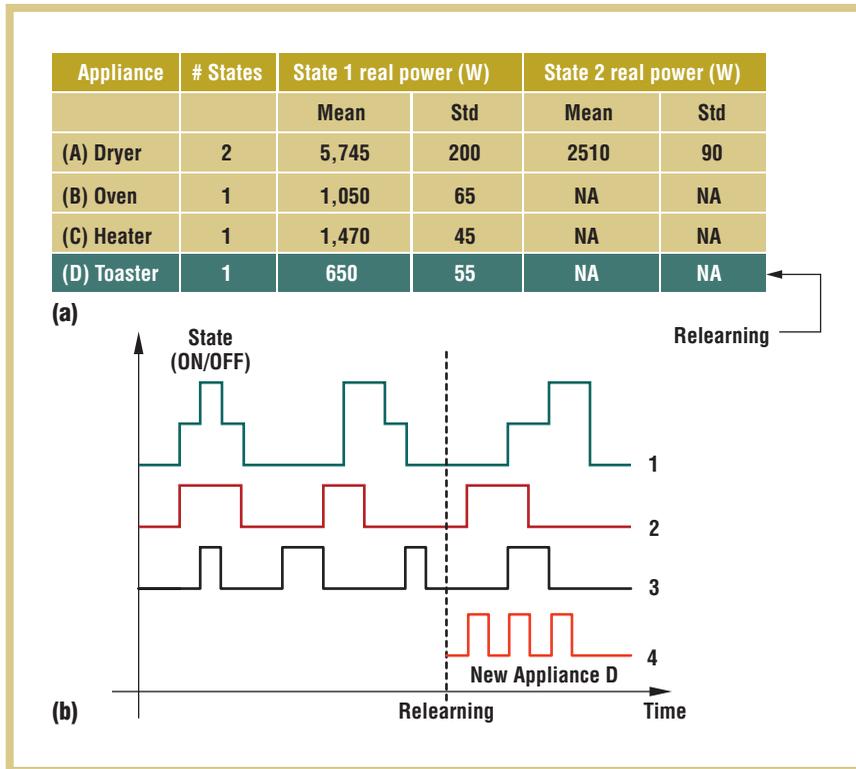


Figure 4. Sample (a) static and (b) dynamic tables. The static table stores the accurate state information for detected appliances learned at the ESP. The smart meter uses the dynamic table to track appliances' states in real time.

event's total power change. Continuously running the knapsack algorithm is computationally intensive, but incremental analysis is prone to error propagation.

To save computational power while minimizing error propagation, we developed a hybrid design that runs a modified knapsack algorithm on each edge event.⁴ This detection algorithm works well if the static table is accurate, each appliance has discrete finite states, and no two appliances consume the same power.

Building the Static Table

The first step in back-end data analysis is to establish clusters of on and off events to identify the appliances. The clustering algorithm accomplishes this by taking an interval of data and grouping like events by their respective power changes.

The learning phase must then identify grouped power changes to classify large appliances and group each appliance's power change. The most basic appliance model is the on/off model; however, not all appliances can be explained by just on/off states. To identify appliances, we implemented an algorithm based on genetic algorithms and dynamic programming that builds FSMs from edge-event clusters.^{6,7}

When to Learn

The learning schedule consists of meter-initiated reactive learning and controller-initiated proactive learning. When either detects the need to learn, it informs the other, and the meter transmits data until the controller indicates it has sufficient data to build a static table. If the meter finds it has no table (for instance, on initiation), be-

periodically if the appliance-detection module finds it necessary or if the controller asks the meter to do so. For instance, relearning might be necessary when the consumer introduces a new appliance or when the meter has difficulty identifying the correct appliance from an edge event.

The Distributed NILM Algorithm

This algorithm addresses the NILSV problem by detecting a specific appliance using a residential meter.⁴ We aim to split the computation between the meter and the back-end system with a well-chosen level of communication between the two.

Edge Detection

In both the learning and monitoring phases, the meter's first goal is to detect abrupt changes in the power readings corresponding to a large appliance turning on or off. The algorithm ignores minor changes, allowing the meter readings collected every second to be represented using only the power

values at large power changes and the corresponding event time. This process compresses the data because change events are rare compared to the original periodic power measurements.

Building the Dynamic Table

The appliance-detection module

- identifies appliances' current states from real-time edge events and creates the static appliance table using the back-end analysis, and
- detects static-table errors, triggers the meter to enter the relearning phase, and eventually receives an up-to-date static table.

Two common methods to detect appliance states are a *knapsack algorithm* and *incremental analysis*.⁵ A knapsack algorithm finds the combination of appliances whose total power is maximized under the constraint that the total power is less than the current observed power. Incremental analysis determines which appliances changed their states, on the basis of each edge

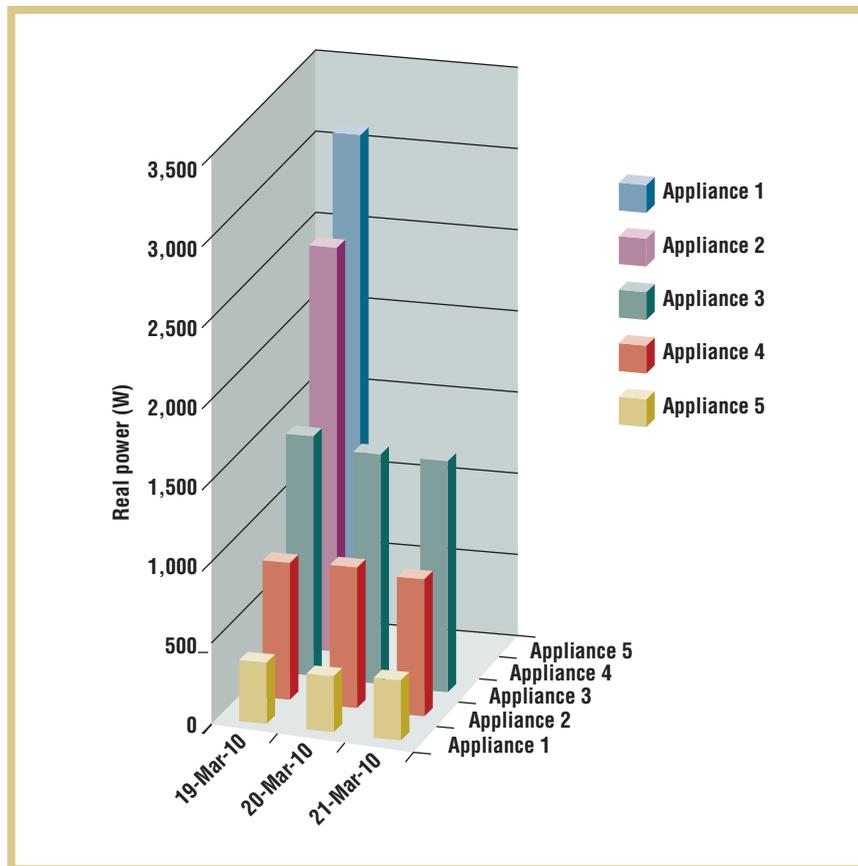
Figure 5. Detected appliances from 19 to 21 March. Our prototype system identified three loads regularly throughout the experiment.

believes the consumer has added a new appliance, or detects an error in the static appliance table, it sends a learning request. If the controller believes that an LSV is incorrect, it can issue a learning request. The meter should remove appliances no longer present in the static table to increase detection efficiency. Generally, the meter has difficulty detecting an appliance's removal without conducting advanced analysis. To eliminate removed appliances from the static table, the meter will periodically reenter the learning phase if the elapsed time since the last learning event exceeds a threshold.

Figure 4 shows sample static and dynamic tables before and after relearning. The back end starts building appliance tables from edge-event data on requests from the meter (reactive relearning) or periodically (proactive relearning). We assume the algorithm that builds the table finds similar appliance profiles over multiple days because residents will be using LSV candidate appliances regularly. Therefore, the ESP examines the tables created from multiple datasets. If it finds an appliance whose state-transition profile differs from those of the previously detected appliances, it assumes the consumer has added a new appliance. The controller ends its learning period when it doesn't find new appliances for a specified time period. The ESP can adjust the learning time and frequency to avoid network congestion.

Experimentation

Proving the viability of NILSV and our distributed NILM algorithm requires significant experimental analysis in a wide range of contexts. The following experiments demonstrate what's possible and issues we must address. We concentrate on verifying the static and



dynamic tables' accuracy. To generate the static table, we implemented the algorithm in Java and ran it on a high-powered desktop workstation to avoid resource limitations. We implemented the dynamic-table-building code on an AVR32 simulator for the Atmel UC3A0512 chip to simulate the smart meters' processing constraints.

In our LSV test scenario, the homeowner purchases an appliance and registers it with the ESP for load shedding. The ESP has already independently conducted tests to verify the appliance's signature and determined the load fingerprint that should be in the static table. The ESP then sends LSIs to the house to turn the appliance on and off, and NILSV determines whether the appliance follows the commands.

We conducted an experiment using data collected from a typical residential home in Urbana, Illinois. We measured the aggregate power using TED 1000. TED 1000 is nonintrusive and cost ef-

fective; when installed in the breaker panel, it can take one reading per second with an accuracy of ± 2 percent. The appliance to identify was a Honeywell heater operating at 1,500 W. To control the device, we installed Insteon controllers on the heater, allowing a computer to control a set schedule of on and off events. We scheduled the heater to turn on and off during three time periods corresponding to different activity levels in the home—the middle of the night, early morning to afternoon, and evening. For each test set, our ESP software generated LSIs while the heater was on, instructing it to turn off.

We tested the static table by running the learning phase on separate days' data and verifying that the heater was present and the same appliances were running from day to day. We tested the dynamic-table-building algorithm by noting the table's accuracy both when the heater was on and when it was off.

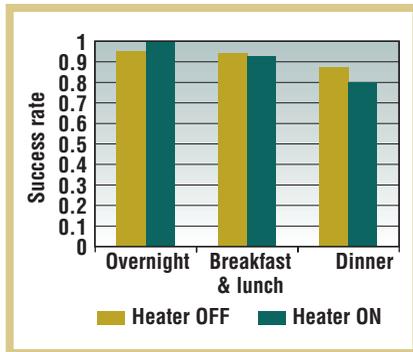


Figure 6. The heater detection success rate. The results show how well the heater was successfully identified in the test home during three different periods in a day.

The ESP would interpret a confirmation of the heater being off as a successful LSI response and would interpret feedback that the heater was on as evidence of a nonconforming appliance. We ran both tests throughout the week of 19–27 March 2010 while the family was living in the house.

We ran the static-table-building algorithm in simulation over all our collected data. Figure 5 shows the identified loads' power consumption during our experiment's first three days. Although 19 March had two extra unique appliances, the experiment confirmed that the learning phase identified the other three loads regularly throughout the rest of the experiment. The experiment confirmed that the learning phase can run for a shorter period of time while still successfully identifying the correct load signatures and accurately maintaining the dynamic table. Even if the learning phase missed the two larger loads in the initial learning phase, when the meter did detect them, it would notice an error in the static table and initiate a relearning phase.

We tested the dynamic tracking by applying the static-table-building algorithm's output from 19 March to the week's data. Figure 6 summarizes the results. During the overnight and early morning-to-afternoon tests, the dynamic table maintained more than

90 percent accuracy. However, during the evening tests, it dropped to approximately 80 percent for identifying when the heater was on and 85 percent for identifying when the heater was off. Testing confirmed the added noise during the evening tests reduced the algorithm's accuracy.

Figure 7 illustrates the test home's real-time power measurements during a 10-minute period in the evening. The heater was scheduled to be on from 6:30 to 6:35 p.m. During that time, a periodic load with roughly the same power signature occurred. This load turned on and off four times throughout the 10 minutes. The algorithm successfully identified the heater when it was on; however, after it turned off at 6:35, our algorithm still classified the periodic load as the heater. This scenario illustrates the difficulty in determining the appliance's state when multiple appliances share the same power signature.

Scalability Analyses

For NILSV to be viable, the proposed methods must be scalable in bandwidth and processing power on AMI networks. Throughout experimentation, the test home yielded an average of 1,000 step events, or approximately 15 Kbytes of data per day. Assuming a high-speed backhaul network, bandwidth is limited primarily between the meters and backhaul points. With 4,000 meters per backhaul point, the bandwidth requirement would be approximately 60 Mbytes ($4,000 \times 15$ Kbytes) of traffic per day or 5.5 Kbps of bandwidth.

Assuming the use of a network such as a ZigBee operating at 2.4 GHz and yielding 250 kbps, the algorithm consumes only 2.2 percent of network bandwidth. Even assuming a slower ZigBee network with only 30 kbps of bandwidth and a three-hop mesh network yielding an effective bandwidth of 10 kbps, the algorithm can still support relearning for every node every day using 55 percent of the total net-

work bandwidth. Of course, these are worst-case estimates because learning reduces the bandwidth requirements. Using the 10-kbps ZigBee network with each node learning one day a week, the algorithm would require 7.9 percent of network bandwidth; relearning one day per month would require 1.8 percent.

Regarding the computational cost, the primary bottleneck is the learning phase in the ESP back office. Although it depends on the parameters, we needed less than a minute in our tests to build a static state table from one day of data on a typical desktop PC. Thus, one machine can handle approximately 1,440 (24×60) homes per day. Only three ($4,000/1,440$) processor cores per backhaul point are sufficient, even when learning every day, assuming learning is scheduled to fully utilize computational resources.

NILSV Benefits

The industry is overwhelmingly heading toward demand-response programs in which the ESP directly controls home appliances. Such control will become increasingly important with the adoption of plug-in electric vehicles, which will add significant load to the grid. ESP-approved load-control devices and appliances will receive signals from the ESP and respond to verify status and compliance with load-shed requests. Typically, these home-area-network (HAN) devices will receive load-shed commands sent through the AMI network to the meter, which will then be relayed into the home via a short-range ZigBee wireless gateway in the meter. LSI responses will follow the reverse path. We call this the *meter gateway architecture* (MGA).¹

This MGA vision is fraught with problems. The ESP must trust all the load-control devices and appliances in the customer's home to behave properly. Accordingly, the proposed solutions generally envision cryptographic keys in every HAN device. With appropriate tamper-resistance techniques

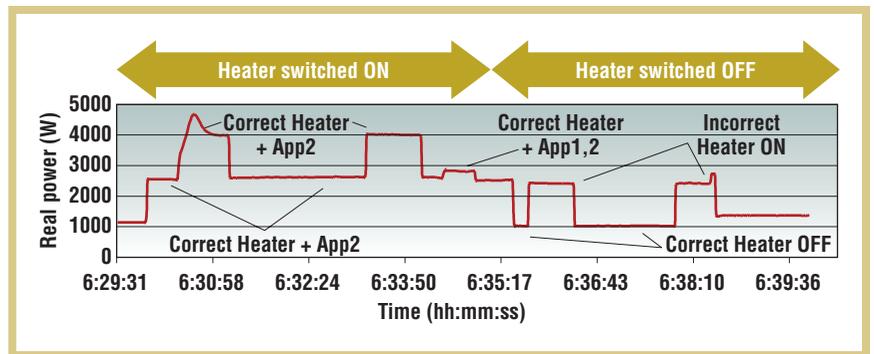
Figure 7. A heater detection scenario.
The results show a detailed appliance detection analysis on an evening power-use graph.

and key management infrastructures, such systems can deliver a high degree of assurance that load-control devices receive and reply to LSIs as intended. However, tamper resistance and cryptography can add significant manufacturing costs to devices as well as system-management complexity. Most ESPs will likely support only a few specific devices for their service area. This will limit demand-response programs' acceptance and inhibit interoperability for devices belonging to a consumer who moves to another service area. Again, in most cases, the loads to control are large appliances retrofitted with load-control devices.

Even with tamper resistance and cryptography ensuring that load-control devices are behaving properly, extending this trust to include the load itself is much harder. For example, many HVAC systems are controlled with a simple 24-volt AC signal. The sophisticated security in the demand-response thermostat is meaningless if the control wires don't connect to the intended load. In summary, the MGA seems likely to result in complex, expensive, inflexible, and insecure solutions with limited consumer acceptance.

By decoupling load control from load-shed verification, NILSV enables simpler, more flexible demand-response architectures. The meter is the only trusted device at the residence. Because it's already trusted to deliver total billing data, adding NILSV functions to it contributes minimal incremental risk. Load-control devices don't require the ESP's trust, tamper resistance, or private keys.

Because power measurements at the meter determine compliance with load-shed messages, NILSV confirms both the integrity of load-shed messages and responses and the load's behavior.



Load aggregators can deliver load-shed messages through the meter, various broadcast channels, the consumer's broadband connections, or other channels that require only one-way communication. Load-shed messages must be authenticated, but this is much simpler than authenticating responses because there are fewer load-shed-message senders than response senders. For load-shed messages delivered via the consumer's broadband, the delivery and authentication mechanism could be as simple as a Web server delivering pages authenticated via HTTPS.

The load-control devices' simplicity will more readily facilitate the development of a standard messaging protocol. These load-control devices, which are usable with any ESP's demand-response program, will become common consumer electronics, available at retail stores.

With an appropriate load-shed language wherein an LSI specifies only a goal, such as "limit device X to Y percent of maximum load until time T," and leaves that goal's implementation up to the load-control device, demand-response programs can be flexible and accommodate a variety of load-control devices. The only requirement is that the load-control device's response to load-shed messages is measurable by NILSV at the meter. Demand-reduction programs can compensate consumers for participation only when LSIs produce actual load reduction. NILSV allows for simpler, cheaper load-control devices with greater variety and simpler messaging and management that will

likely lead to more flexible demand-response programs and greater consumer adoption.

NILSV monitors demand-response compliance without extra communication overhead between appliances and meters. Our proposed method addresses smart-grid issues including the low-end metering hardware and the constrained bandwidth links between the meters and their head ends.

Our method uses the meter as a trusted party and eliminates any communication with untrusted appliances. Doing this circumvents many potential integrity and confidentiality problems. However, we must conduct more research to determine the accuracy of the NILM algorithms at the ESP before the solution is ready for use.

The biggest problem we encountered was when appliances had the same power signature. Because this was the only metric distinguishing the two appliances, the system classified them as one appliance. The algorithm then generated the appliance table incorrectly. We plan to address this problem.

Furthermore, we must assess the efficiency of the algorithm on the meter and ascertain the computing power necessary on the back end to process the data. We also need to examine the average amount of time the meters need to be in the learning phase. For these analyses to be accurate, they'll need to employ a large set of real-world data.

the AUTHORS



David C. Bergman is a design engineer at Advanced Micro Devices, where he works on design verification of x86 processors. His research interests include large-scale simulation and power-grid security. Bergman has an MS in electrical and computer engineering from the University of Illinois at Urbana-Champaign. Contact him at dcborgma@illinois.edu.



Dong Jin is a PhD student in the University of Illinois at Urbana-Champaign's Department of Electrical and Computer Engineering. His research interests are in large-scale computer and communication systems security, including power grids and the Internet. Jin has an MS in electrical and computer engineering from the University of Illinois at Urbana-Champaign. Contact him at dongjin2@illinois.edu.



Joshua P. Juen is an MS student in the University of Illinois at Urbana-Champaign's Department of Electrical and Computer Engineering. His research interests are computer security and privacy, including the design of secure systems, anonymous Internet browsing, and location-based privacy. Juen has a BS in electrical engineering from Central Michigan University. Contact him at juen1@illinois.edu.



Naoki Tanaka is a PhD student and research assistant at University of Illinois at Urbana-Champaign's Department of Computer Science. His research interests include risk-aware authorization based on economic models. Tanaka has a Master of Information Science and Technology from Osaka University. Contact him at tanaka5@illinois.edu.



Carl A. Gunter is a computer science professor at the University of Illinois at Urbana-Champaign and director of the Illinois Security Lab and the Center for Health Information Privacy and Security. His research interests are programming-language semantics, formal analysis of networks and security, and privacy, including privacy issues for the power grid and healthcare. Gunter has a PhD from the University of Wisconsin. He's a member of IEEE and the ACM. Contact him at cgunter@illinois.edu.



Andrew K. Wright is chief technology officer at N-Dimension Solutions, where he guides the company's technical strategy for developing cybersecurity products for the electric-power sector. His research interests include cybersecurity for industrial and control systems, network and smart-grid security, and applied cryptography. Wright has a PhD in computer science from Rice University. Contact him at andrew.wright@n-dimension.com.

Finally, we must produce a viable NILM scheme and improve appliance classification. Currently, we use only real power to identify appliances. Classifying appliances on the basis of time and frequency of use and other less-deterministic measurements is an interesting possibility. 

REFERENCES

1. M. LeMay et al., "An Integrated Architecture for Demand Response Communications and Control," *Proc. IEEE Hawaii Int'l Conf. System Sciences (HICSS 08)*, IEEE CS Press, 2008, p. 174.
2. A. Lee and T. Brewer, *Smart Grid Cyber Security Strategy and Requirements*, tech. report, US Nat'l Inst. Standards and Technology, 2009.
3. G.W. Hart, "Nonintrusive Appliance Load Monitoring," *Proc. IEEE*, vol. 80, no. 12, 1992, pp. 1870–1891.
4. D.C. Bergman et al., "Distributed NILM," Apr. 2010, Manuscript. **//It is under review, notification on Nov 19//**
5. M. LeMay, J.J. Haas, and C.A. Gunter, "Collaborative Recommender Systems for Building Automation," *Proc. 42nd IEEE Hawaii Int'l Conf. System Sciences (HICSS 09)*, IEEE CS Press, 2009, pp. 1–10.
6. M. Baranski and J. Voss, "Genetic Algorithm for Pattern Detection in NIALM Systems," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, vol. 4, IEEE Press, 2004, pp. 3462–3468.
7. M. Baranski and J. Voss, "Detecting Patterns of Appliances from Total Load Data Using a Dynamic Programming Approach," *Proc. 4th IEEE Int'l Conf. Data Mining (ICDM 04)*, IEEE CS Press, 2004, pp. 327–330.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.