nPlug: An Autonomous Peak Load Controller

Tanuja Ganu, Deva P. Seetharam *Member, IEEE*, Vijay Arya *Member, IEEE*, Jagabondhu Hazra, Deeksha Sinha, Rajesh Kunnath, Liyanage Chandratilake De Silva, Saiful A. Husain, and Shivkumar Kalyanaraman *Fellow, IEEE*

***Abstract*—The Indian electricity sector, despite having the world's fifth largest installed capacity, suffers from a 12.9% peaking shortage. This shortage could be alleviated, if a large number of deferrable loads, particularly the high powered ones, could be moved from on-peak to off-peak times. However, conventional Demand Side Management (DSM) strategies may not be suitable for India as the local conditions usually favor inexpensive solutions with minimal dependence on the pre-existing infrastructure. In this work, we present a completely autonomous DSM controller called the *nPlug*[1]. nPlug is positioned between the wall socket and deferrable load(s) such as water heaters, washing machines, and electric vehicles. nPlugs combine local sensing and analytics to infer peak periods as well as supply-demand imbalance conditions. They schedule attached appliances in a decentralized manner to alleviate peaks whenever possible without violating the requirements of consumers. nPlugs do not require any manual intervention by the end consumer nor any communication infrastructure nor any enhancements to the appliances or the power grids. Some of nPlug's capabilities are demonstrated using experiments on a combination of synthetic and real data collected from plug-level energy monitors.**

Our results indicate that nPlug can be an effective and inexpensive technology to address the peaking shortage. This technology could potentially be integrated into millions of future deferrable loads: appliances, electric vehicle (EV) chargers, heat pumps, water heaters, etc.

***Index Terms*—Smart Plug, Demand Response, Peak Loads, Scheduling, Multiple Access**

I. INTRODUCTION

AS OF NOVEMBER 2011, the Indian electricity sector, despite having the world's fifth largest installed capacity of 185.5 GW, suffers from a 12.9% peaking shortage and 10.3% energy shortage [2]. The situation could worsen with the current trends in population and income growth, industrialization, and urbanization. Electricity consumption is expected to increase substantially in the coming decades as well [3].

Considering that electricity cannot easily be stored in large scale, peak shortage can be alleviated by increasing supply or by reducing demand. Supply can be increased through the use of "peaker" power plants that operate on fast-starting fuels. Peaker plants operate only during the peak, for a small fraction of time, so their electricity is inherently expensive. It is estimated that if India were to add peakers to the existing generation portfolio, the average supply cost might increase by over 35% [4].

Clearly, there is a significant role and potential for demand side management (DSM) programmes in India. The Government of India, through new Energy Conservation legislation, is also seeking to implement a host of such programmes within the country [5]. However, conventional DSM strategies may not be suitable for India as the local conditions usually favor only inexpensive solutions with minimal dependence on the pre-existing infrastructure [6]. To address this need, we developed an autonomous DSM system based on smart plugs called nPlugs [1] that "sit" between deferrable loads and wall sockets. An nPlug senses line voltage and frequency to infer the load level and supply-demand imbalance in the grid respectively. It processes the sensed data using resource-efficient data mining algorithms to identify the peak/off-peak periods and imbalance conditions of the power grid. It runs the attached load(s) during user-specified time intervals while avoiding unfavorable grid conditions (peak load hours and supply-demand imbalance conditions) as much as possible. As a result, each nPlug runs a decentralized load rescheduling algorithm that contributes to peak load reduction by distributing the loads over time. The benefits of our approach are:

Network free - Since nPlugs don't require any network infrastructure for sensing or control, they can be completely autonomous. This makes nPlugs particularly appropriate for locations where communication infrastructure is underdeveloped. For instance, in India, as there is spectrum crunch to serve the data/voice communication needs of humans, there may not be sufficient bandwidth to support machine-to-machine communications that would be required by centralized DSM solutions. Even wired networks may not be widely applicable as only 11.3% of Indian households have access to Internet [7].

Location-specific load management - nPlugs sense the line voltage to determine whether the incoming feeder is congested or not. As the line voltage reflects the load on the local transformer and load on the grid that feeds that transformer, nPlugs can alleviate the local load levels even if the overall grid is not congested. It is important to note that the gains from decentralized demand reduction could add up and alleviate grid-level load issues as well.

Brownfield innovation - nPlugs don't require any changes to the grid or to the appliances that they manage. This approach is particularly suitable for a mature system like the power grid and for the millions of appliances already in use.

Manuscript received October 8, 2012; revised March 18, 2013.

T. Ganu, D.P. Seetharam, V. Arya and J. Hazra are with IBM Research, India (e-mail: tanuja.ganu@in.ibm.com).

D. Sinha is a student at IIT Mumbai, India. This work was done while she interned at IBM Research, India.

R. Kunnath is with Radio Studio, India.

S. A. Husain and L. C. De Silva are with Universiti Brunei Darussalam.

S. Kalyanaraman is a Senior Manager at IBM Research, India. He is also a visiting professor at Universiti Brunei Darussalam.

Digital Object Identifier 10.1109/JSAC.2013.130705.

[1]An earlier version of the paper was presented at ACM e-Energy conference 2012 [1]. This paper extends the work on nPlug Software Components and presents additional analysis and experimental results.

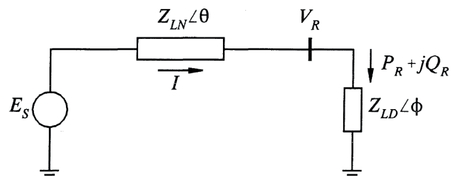


Fig. 1. A two-bus power system

Incremental adoption - since each nPlug has the potential to alleviate peak load, nPlugs can be introduced in small batches into the grid. This reduces the initial investment as well as the risks of introducing new technology into a pre-existing infrastructure.

No policy changes required - Since nPlugs don't depend on differential pricing schemes or smart meters, deploying them doesn't require any regulatory approvals. Any customer who is willing to contribute to peak load reduction can do so by simply plugging a deferrable load into an nPlug.

Inexpensive solution - every hardware and software component in nPlugs are based on careful analysis of cost-performance trade-offs. The prototype we have built costs about USD 30 in small volumes (< 100 units) and we estimate nPlugs in large volumes ($> 100,000$ units) would cost about USD 15.

The rest of the paper is organized as follows. Section II presents the power systems basis for inferring grid load and supply-demand imbalance by sensing line voltage and frequency. The details of nPlug hardware design is described in Section III. The data mining algorithms used to identify peak and off-peak periods as well as load scheduling algorithms are presented in section IV. Experimental evaluation of our algorithms is presented in Section V. Section VI presents related work and finally section VII concludes with a discussion about future work.

II. Power Systems Background

This section uses the power systems theory to explain why the line voltage and frequency measured at a consumer premise can serve as good indicators of grid load and supply-demand imbalance respectively.

Inferring grid load from voltage. Figure 1 shows a simple "power distribution system" wherein a load is connected to a source through a transformer. $\tilde{E_S}$ is the source voltage, $\tilde{V_R}$ is the load voltage, $\tilde{Z}_{LN}$ is the transformer impedance, and $\tilde{Z}_{LD}$ is the load impedance (all quantities are vectors). We will now see how the magnitude of load voltage V_R decreases with increasing load.

The current flowing through the line and load, $\tilde{I}$ is given by $\tilde{I} = \frac{\tilde{E_s}}{\tilde{Z}_{LN}+\tilde{Z}_{LD}}$ where $\tilde{Z}_{LN} = Z_{LN}\angle\theta = Z_{LN}cos\theta + jZ_{LN}sin\theta$ and $\tilde{Z}_{LD} = Z_{LD}\angle\phi = Z_{LD}cos\phi + jZ_{LD}sin\phi$.

Here θ is phase angle between reactive and resistive components of the transformer impedance while ϕ is the phase angle between the load current and voltage. Now the magnitude of current I is given by

$$I = \frac{E_S}{\sqrt{(Z_{LN}cos\theta + Z_{LD}cos\phi)^2 + (Z_{LN}sin\theta + Z_{LD}sin\phi)^2}}$$

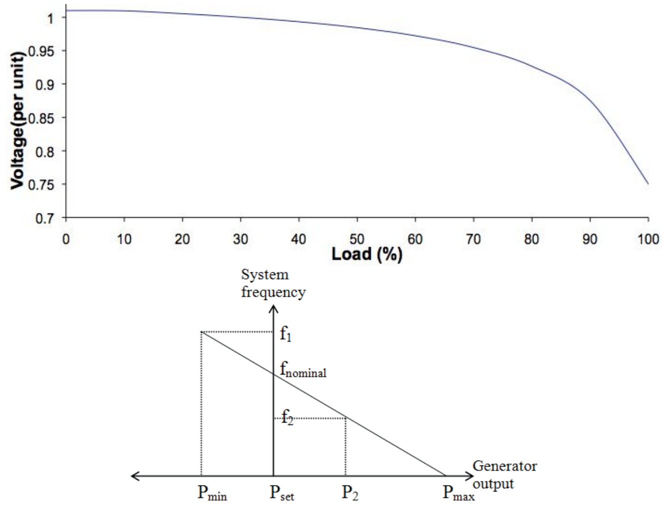


Fig. 2. (top): Load-voltage characteristics of a distribution system, (bottom): Load-frequency control characteristics

Therefore the magnitude of load voltage V_R is:

$$V_R = Z_{LD} \times I \tag{1}$$

Since the source voltage E_S and transformer impedance $Z_{LN}\angle\theta$ are generally constant, the load voltage V_R is essentially a function of the magnitude of load impedance Z_{LD} and the power factor $\cos\phi$. To minimize reactive power consumption, appliances are usually designed to have high power factor (0.9 to 1). Thus from Eq.(1) we see that the load voltage V_R is dominated by the magnitude of load impedance Z_{LD}. As the load increases (i.e. impedance decreases), the load voltage V_R decreases and vice versa.

To illustrate this connection between load voltage and load level, we simulated a transformer system with the following initial conditions: source voltage (E_s) of 1.05 per unit, line impedance (Z_{LN}) of 0.04 per unit and load impedance (Z_{LD}) of 1 per unit. As shown in Figure 2(top), load voltage doesn't falls until the load on the transformer exceeds 60-70% of its capacity and beyond that voltage drops rapidly. This is not surprising because with the increase in load, auto transformers try to maintain a constant load voltage by changing their tap positions. Beyond 60-70% load, tap positions cannot be adjusted to control load voltage anymore which leads to the fast voltage dip. This voltage dip could be sensed locally for decentralized demand management in distribution system and it forms the basis of nPlugs. In section V, we plot the variation in voltage V_R measured at a household and show how it drops during peak hours.

Though above experiment shows local voltage sensing is a good indicator of overload in distribution system, further investigation is required to understand how voltage levels would be impacted by automatic voltage controllers (such as switching capacitor banks, static and dynamic VAR (Volt Ampere Reactive) compensators, etc) because these controllers dynamically change source voltage and maintain a constant load voltage. However, since such voltage controllers are expensive, they are not installed on most of the distribution substations in India and in many other developing countries.

Detecting supply-demand imbalance from frequency. Conventionally, the grid frequency is regarded as an indicator of imbalance between generation and demand. During imbalance, the output of each generator is automatically adjusted to meet the demand. This changes the system frequency according to the *load-frequency* characteristics of the generators as shown in Fig. 2(bottom). The plot shows that when the load on generation is higher than P_{set} (the generation needed to support a fixed load), the frequency drops. On the other hand, if it is less than P_{set}, the frequency rises.

Although frequency is a good indicator of imbalance, our measurements show that frequency may not be sufficient to identify grid load levels accurately. One possible reason for this is that in anticipation of increased demand, the generation is ramped up to keep the frequency close to nominal levels. Moreover, frequency indicates grid-wide supply-demand imbalance. On the other hand, voltage can indicate load levels at the local transformer and feeder.

Although the power systems theory explained above is well-known, to the best of our knowledge, none of the existing systems learn the voltage/frequency patterns to derive load schedules that can help reduce peak loads.

III. nPLUG HARDWARE DESIGN

Figure 3(top) shows an initial prototype of nPlug. The hardware design is based on cost-performance trade-offs. Figure 3(bottom) shows the hardware modules of nPlug, which are user controls, frequency and voltage sensing circuits, relay, real time clock and power supply.

User Interface. nPlug is equipped with buttons for entering scheduling preferences and for overriding nPlug's scheduling decisions, and a 32-character (16x2) LCD.

Controller, Memory and Storage. The current design uses a Microchip PIC24FJ128GA010 16 bit microcontroller. This 4MIPS controller has 128KB of Program Memory, 8KB of RAM and a SPI flash memory interface. A 4 Mb Flash memory with SPI Interface is used to store the end user preferences, the sensing history as well as the outputs from the learning and scheduling modules

Voltage Sensing. Line voltage is sensed by measuring voltage across a resistive divider (built with 1% tolerance resistors) between phase and neutral. The divider is sized in such a way that the dynamic range of microcontroller's 10-bit Analog to Digital Converter (ADC) can handle the entire input voltage range (110 V - 350 V) Moreover, as the ADC input requires only negligible current, low power, high value resistors are chosen so that the current used for measurement is minimal.

Frequency Sensing. Frequency is sensed using a current-limiting resistor directly connected from the phase to the microcontroller input. The protection diodes in the microcontroller I/O pin act as rectifiers limiting the voltage in both directions converting the mains signal into a trapezoidal waveform. Frequency is determined by counting the number of zero-crossing positive transitions (occur only once per cycle) in one second. A time interval of one second is used for computing frequency since it remains fairly constant over this period. The measurement interval of one second is derived from an asynchronous timer which counts at 100 microsecond resolution. This gives an accuracy of 0.01 hertz.

Real Time Clock (RTC). Scheduling decisions are based on the accuracy of the clock. Since there is no network interface for a nPlug to synchronize its clock with true time, an accurate yet power-efficient RTC (DS1307) that is driven by a 10 ppm crystal and powered by a coin cell battery (CR 2032) is included onboard. The battery needs to be replaced once every three years or so.

Relay. A relay is required to turn the attached appliances ON and OFF. This is achieved by means of commonly available mechanical relays that can handle 30A at 230V. As the relay must be operated only when the line voltage is close to zero, frequency measurement interrupt whose positive edges are at the zero-crossings is used to time the opening and closing of the relay.

Power Supply. Different components in nPlug require different range power supplies. Relay, RTC and microcontroller require 12 V, 5 V and 3.3 V power supplies respectively. A low cost primary side regulated CC/CV switch-mode regulator is used for generating the 12 V, and 5 V and 3.3 V are generated using LM317 regulators.

IV. nPLUG SOFTWARE COMPONENTS

Figure 3(bottom) shows the high level architecture of an nPlug that has four software components explained in the following sections: (i) UI manager, (ii) Data manager, (iii) Analytics module, and (iv) Load scheduler.

A. UI Manager

The UI Manager accepts following user-specified constraints: 1. *Earliest start time*: the earliest time at which an appliance can be switched on; 2. *Latest end time*: the latest time at which the appliance must finish running; and 3. *Duration*: the duration for which the appliance must be powered. For example, a residential consumer who leaves for work at about 8 AM may specify that her insulated water heater must be run for 30 minutes between 4 AM and 7 AM. In addition, the user is allowed to specify an optional *Hold time*: the minimum time an appliance must be run at a stretch when turned ON. This can be used for devices such as washing machines, dishwashers, and storage-based appliances (inverters/PHEVs) which may not be required to run continuously and their operational duration could be interrupted based on grid constraints.

B. Data Manager

The data manager works as an interface between the hardware sensors and storage. nPlugs sense the grid at regular time intervals to measure line voltage and frequency. The sensed data is preprocessed and saved in the data storage for analysis by the analytics module. Due to memory and processing constraints of nPlug hardware, there are limitations on the data volume that it can handle. Therefore the data manager compresses the sensed data prior to storage. The data is compressed using the Piece-wise Aggregate Approximation (PAA) technique [8] that is simple enough to compute even on a microcontroller. PAA compresses the sensed data by segmenting the data sequence into fixed-length sections and

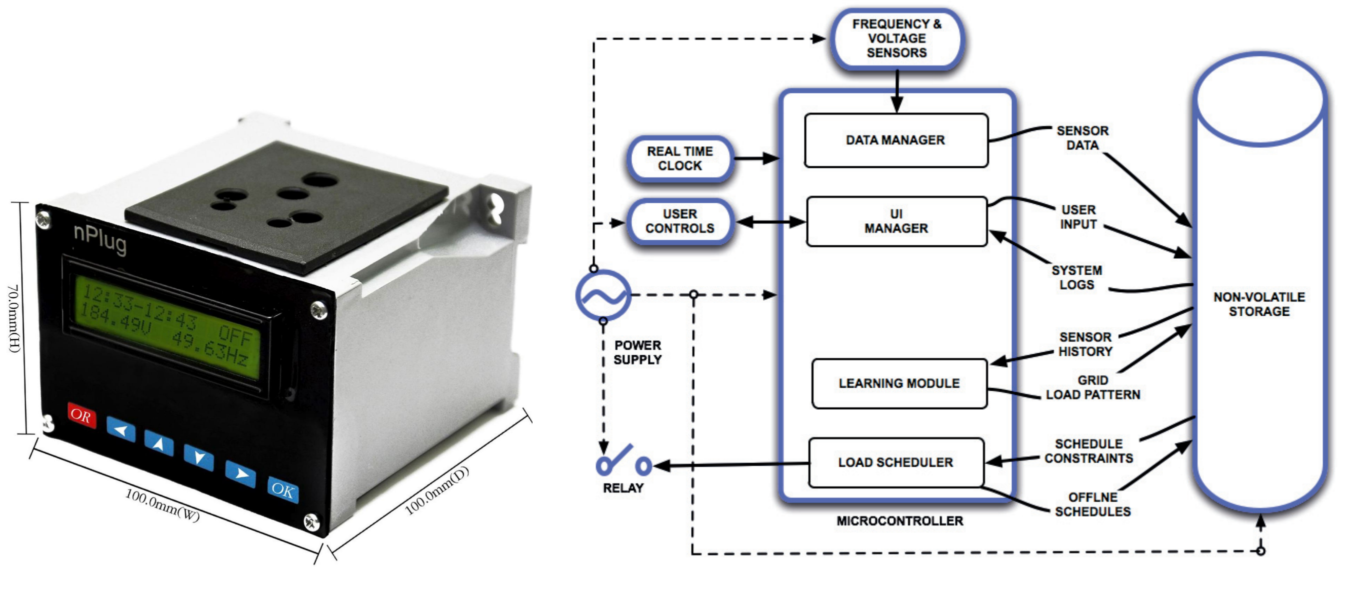


Fig. 3. nPlug prototype (left), system overview (right)

calculating the mean value of these sections. Given a time series V with n data points, $V = \{v_1, v_2, \ldots, v_n\}$, PAA divides the series into the segments of length w and creates a compressed series $V' = \{v'_1, v'_2, \ldots, v'_m\}$ of length $m = \frac{n}{w}$, where

$$v'_i = \frac{1}{w} \sum_{j=(i-1)\times w+1}^{i\times w} v_j \quad \forall i \in \{1 \ldots m\}$$

Thus PAA compresses the original data by a factor of w. PAA attempts to preserve the similarities in the original time series and allows data analysis on the compressed representation instead of the original. Furthermore, PAA supports stream processing that is beneficial in the resource-constrained environments such as nPlug. In nPlug, we use $w = 300$ that provides sufficient dimensionality reduction and still retains granular (5 minutes interval) information for further data analysis. Figure 4 shows the voltage time series measured at an indian household for a day at every second (blue) and the corresponding PAA compressed time series (red).

C. Analytics

The analytics module uses the sensed voltage and frequency data to identify (i) peak and off-peak periods and (ii) situations of supply-demand imbalance.

1) Inferring peak and off-peak periods: nPlugs learn the peak and off-peak periods of the power grid by analyzing the voltage time series data collected and stored by the data manager. This information is then used to make scheduling decisions for the deferrable load attached to the nPlug.

The peak and off-peak periods are identified using two steps. Firstly, the stored PAA data is transformed into a more meaningful symbolic representation i.e. low, medium, or high load by using an approach similar to Symbolic Aggregate Approximation (SAX) [8]. The SAX representation is used when the time series exhibits a Gaussian distribution. In order to discretize/label a time-series with k alphabets, the SAX approach defines $k-1$ break points $\beta_1, \beta_2, \ldots, \beta_{k-1}$

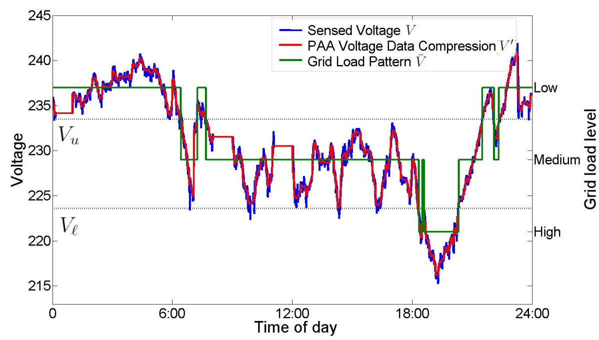


Fig. 4. Sensed voltage time series (blue), with PAA data compression (red) and grid load pattern (green).

in the Gaussian curve producing k equal-sized areas under the curve. All values within an interval (β_i, β_{i+1}) are coded with the symbol corresponding to the interval. However the voltage time series is skewed and does not follow a Gaussian distribution. Therefore we use domain knowledge and identify lower and upper break-points using the following heuristic: $V_\ell = \min(V') + 0.3 \times (\max(V') - \min(V'))$ and $V_u = \min(V') + 0.7 \times (\max(V') - \min(V'))$. Thus values $\leq V_\ell$ are classified as high load, values $\geq V_u$ as low load, and values in between as medium load level. The resulting 3-alphabet time series is called as the grid load pattern $\bar{V}$. Fig 4 shows the grid load pattern in green the break points V_ℓ and V_u using dotted lines.

Let $\bar{V}^1, \ldots, \bar{V}^c$ denote the grid load pattern for previous c days. Considering PAA data compression window, $w = 300$ (i.e. window length of 5 minutes), we get 288 data points in grid load pattern $\bar{V}$ for each day. In the second step, a median grid load-pattern $\hat{V}$ for a 24-hour period is computed by considering the grid-load pattern of previous c days, where each entry at time t is the median of previous c entries at the same time, that is $\hat{v}_t = \mathit{median}_{i=1}^{c}(\bar{v}_t^i)\ \forall t = 1 \rightarrow 288$. All time periods of high load in the median grid load pattern are regarded as peak periods and the balance as off-peak periods.

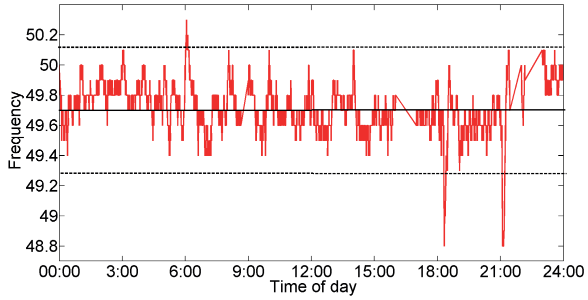


Fig. 5. Sensed frequency time series for a day at a household. The mean, f_μ, and 2-SD thresholds, f_ℓ and f_u, are indicated using the solid and dashed lines respectively.

2) Inferring supply-demand imbalance: To ensure grid and appliance safety, nPlugs avoid scheduling appliances during periods of supply-demand imbalance. Unlike peak load, the supply-demand imbalance situation does not repeat periodically every day. The imbalance is mostly due to unplanned or sudden change in demand or supply and can be detected by using the line frequency, as discussed in Section II. nPlug learns the normal operating range of grid frequency by analyzing the sensed frequency data and identifies the imbalance as an outlier. We use the 2-SD (two standard deviation) statistical test [9] to compute the thresholds of normal operating frequency. The lower and upper operating thresholds, f_ℓ and f_u, are computed as $f_\ell = f_\mu - 2 \times f_\sigma$ and $f_u = f_\mu + 2 \times f_\sigma$.

where f_μ and f_σ are the mean and standard deviation of sensed frequency data. Since these can be computed in an online manner on a microcontroller, it is not required to store the entire frequency time series data. In order to reduce sensitivity to the extreme outliers that can change f_μ and f_σ, values beyond 3-SD are discarded from computations. At every sampling time interval, nPlug senses the line frequency, f_t, and if it is less than f_ℓ, it is identified as the situation of supply demand imbalance. Otherwise, f_ℓ is updated using f_t. Figure 5 shows the frequency time series along with thresholds f_ℓ and f_u.

D. Decentralized Scheduling

This section discusses various strategies used by individual nPlugs to schedule deferrable loads by taking into account user specified constraints as well as grid load conditions. The scheduling algorithms used by each nPlug contribute to load-leveling and reduction of the aggregate peak load without any centralized control. In this work, we focus on a ON-OFF load control model, which allows nPlugs to *defer* the load offered by appliances by turning them on or off. The model applies to appliances such as water heaters, dish washers, washing machines, and storage-based appliances. Other load control models which allow nPlugs to *reduce* the load offered by an appliance via power/current control are under study for future work.

An nPlug receives the earliest start time S_t, latest end time E_t, the operational duration d, and the hold time h of the appliance from the end user. For loads whose operational duration cannot be split, $h = d$. The time between S_t and E_t is treated as divided into discrete time intervals each of width τ. Let $n = (E_t - S_t)/\tau$ be the total number of available time slots. Let $H = h/\tau$ be the number of contiguous slots that the appliance needs to run at a stretch after being switched on. Let $D = d/\tau$ be the total number of slots needed by the appliance to finish work. We now discuss three scheduling schemes that may be used by nPlugs: (i) Off-peak scheduling, (ii) Randomized scheduling, and (iii) GSMA Scheduling.

1) Off-peak Scheduling: This is a plain vanilla scheduling scheme wherein an nPlug attempts to avoid peak time intervals if possible. As discussed in section IV-C1, nPlugs learn the peak time intervals adaptively by sensing the grid. Hence the set of feasible start times to schedule the appliance are all slots $\in [S_t, E_t - d]$ *excluding* the set of peak time slots, where the device can be run for $D = d/\tau$ continuous slots. nPlugs use a simple rule-based approach wherein the appliance is scheduled at the earliest possible time slot that provides minimum overlap between the operational slots and the peak time slots.

2) Randomized Scheduling: Although Off-peak scheduling is useful, it may cause coordinated peaking during off-peak hours if several nPlugs use the same rule to shift loads to common time slots. Randomized scheduling attempts to distribute the loads uniformly across time. Each nPlug picks a slot uniformly at random among all slots $\in [S_t, E_t - d]$ and schedules the appliance at the start of the slot. Peak time slots may also be excluded if necessary. Given sufficient number of time slots, randomized scheduling yields a uniform demand distribution across the off-peak slots and a commensurate reduction in the peak load.

The performance of randomized scheduling can be seen by comparing the loads introduced by both randomized and optimal centralized scheduling schemes over time. Let m be the total number of all customer appliances that need to be scheduled between S_t and E_t and ℓ be the load introduced by each appliance. An optimal scheme will schedule loads back-to-back and introduce a constant load of $\mu^* = \frac{m}{(n/D)}\ell = \frac{mD\ell}{n}$ on the grid during each time slot between S_t and E_t.

For the randomized scheme above, appliances start in slots $\in [1, n - D]$ uniformly at random. Let $x_t^j = 1$ if the jth appliance starts in the time slot t, 0 otherwise. Therefore $\mathsf{Pr}(x_t^j = 1) = \mathsf{E}[x_t^j] = 1/(n - D)$. Let L_t be the total load introduced at any time slot t, $\forall t > D$.

$$L_t = \ell \sum_{i=t-D}^{t} \sum_{j=1}^{m} x_i^j \;\; \therefore \mu_t = \mathsf{E}[L_t] = \ell \sum_{i=t-D}^{t} \sum_{j=1}^{m} \mathsf{E}[x_i^j] = \frac{mD\ell}{n-D}$$

The random load L_t and its mean μ_t can be compared by using Chernoff bound.

$$\text{For } \delta \geq 0, \;\; \mathsf{Pr}(L_t > (1+\delta)\mu_t) < \Big(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\Big)^{\mu_t}$$

The above probability decreases exponentially with number of appliances. For e.g., if $m > 50$, even for $\delta = 0.2$, it hits 0. This implies that $L_t \approx \mu_t$. μ_t in turn is close but slightly larger than $\mu^* = \mu_t(1 - \frac{D}{n})$. Therefore for small D/n, the difference between randomized and optimal scheduling is small. If the hold time $H < D$, then instead of generating one random start time to schedule the load for D consecutive time slots, the nPlug can generate D/H random numbers to schedule each block of load for H consecutive time slots. In this case the performance of the scheme improves as $H/n < D/n$.

3) GSMA (Grid-sense multiple-access) Scheduling: Both off-peak and randomized scheduling schemes above help reduce peak loads. However they cannot respond to load fluctuations as they are agnostic to the running capacity of the grid. In GSMA-scheduling, which is inspired from multiple-access protocols in packet networks, multiple nPlugs continuously sense the grid and attempt to acquire service in the presence of varying load. The nPlugs use voltage sensing to determine if the load on the grid is low or high (i.e. if spare capacity is available or not). If the sensed voltage is sufficiently high, the appliance is switched ON, otherwise the nPlugs back-off and attempt to schedule the appliance at a later stage. The length of each time slot τ is assumed to be long enough so that if appliances are switched ON or OFF in the previous time slot, the altered grid capacity can be sensed in the next slot.

Algorithm 1 presents a GSMA-based *probabilistic negative linear back-off* (PNLB) algorithm used by the nPlugs. In PNLB, the contention between multiple nPlugs is resolved in two steps. Firstly, if at time slot t, an nPlug wishes to sense the grid, it uses a *contention window* of length $w_c(t)$ and senses the grid at time slot $t + r$ where r is chosen uniformly at random $\in [0, w_c(t) - 1]$. Secondly, after sensing the current voltage v_c during a time slot, each nPlug switches on the appliance with a probability p that is proportional to the currently available grid capacity. $w_c(t)$ and p are given by Eq.(2) where V_ℓ and V_u are the safe operating voltage thresholds of the grid inferred from the sensed data (section IV-C1). The first step mimics the behavior of the optimal scheduling scheme (section IV-D2) and the second step ensures that users react to varying grid load whenever possible.

$$w_c(t) = \max\{1, \frac{n-t}{D}\}, \qquad p = \begin{cases} 0 & \text{if } v_c < V_\ell \\ 1 & \text{if } v_c > V_u \\ \frac{v_c - V_\ell}{V_u - V_\ell} & \text{otherwise} \end{cases} \tag{2}$$

Algorithm 1 Probabilistic Negative Linear Back-off (PNLB)

Input: $S_t, E_t, d, h, \tau, V_u, V_\ell, f_\ell$
1: $n = (E_t - S_t)/\tau$, $D = d/\tau$, $H = h/\tau$, $t \leftarrow 0$, $t_r \leftarrow D$ % Residual time t_r is set to D
2: **if** $t \geq n - t_r$ **goto** step 15 % If enough time available to schedule the appliance
3: $w_c \leftarrow \frac{n-t}{t_r}$ % set the contention window
4: $r \leftarrow$ **randint**$(0, w_c - 1)$
5: $t \leftarrow$ **wait**(r, t) % wait for r time slots
6: $(v_c, f_c) \leftarrow$ **sense** % sense the grid voltage and frequency
7: **if** $(v_c < V_\ell)$ **then**
8: $p \leftarrow 0$
9: **else if** $(v_c > V_u)$ **then**
10: $p \leftarrow 1$
11: **else**
12: $p \leftarrow (v_c - V_\ell)/(V_u - V_\ell)$
13: **end if**
14: **if** **rand**$(0,1) < p$ **and** $f_c > f_\ell$ **then**
15: **switch**(ON) % acquire service with probability p
16: $t_{wait} \leftarrow \min(H, t_r)$
17: $t \leftarrow$ **safewait**(t_{wait}, t) % switch ON for t_{wait} time slots
18: $t_r \leftarrow t_r - t_{wait}$ % update residual time t_r
19: **switch**(OFF);
20: **if** $t_r > 0$ **then**
21: **goto** step 2
22: **else**
23: **exit**
24: **end if**
25: **else**
26: $t \leftarrow$ **wait**$(1, t)$; **goto** step 2
27: **end if**

To understand PNLB, observe that given n slots and m appliances, the minimum number that need to use the grid in each slot so that all finish on time is $k = \frac{m}{(n/D)}$. However if the load on the grid is high in the first few slots and low later, then $< k$ can use the grid at first and $> k$ later. When the algorithm starts, the residual service time $t_r = D$ and the contention window $w_c = n/D$, so that $k = m/w_c$ nPlugs attempt to acquire service in the first slot on average. If the grid capacity is high so that $p \approx 1$, then about k will begin service. If the capacity continues to remains high, about k more will acquire service in the next slot. If the capacity decreases, then $m' = k(1-p)$ users may fail and use a smaller contention window $w'_c(t') = (n - t')/D$, so that m'/w'_c will attempt to acquire service in a future slot $t' > t$. As time progresses, w_c gradually decreases so that the remaining users sense the grid at a fair rate to finish on time. If an appliance has failed to acquire service in all slots, it is switched ON before its finish time.

When $H < D$, an nPlug attempts to sequentially acquire service D/H times, each time scheduling the appliance for H consecutive slots. After each run of appliance, its residual service time t_r as well as the remaining slots to acquire service again i.e. $(n-t)$, both decrease by H. The contention window w_c is updated accordingly so that the nPlug senses the grid at the fair rate to finish on time.

After an appliance is switched ON, nPlugs switch to "safe-wait" state wherein they sense the grid voltage and frequency regularly so that they can switch OFF the appliance if necessary to ensure grid reliability. This "safewait" period ensures that nPlugs can back-off if the grid gets overloaded after the appliance is turned ON.

Fairness and Aggressiveness. In PNLB, the number of nPlugs that sense the grid to acquire service at each time slot is set to mimic the centralized scheduling scheme (section IV-D2) to achieve optimal load leveling. The rate of sensing is controlled by the contention window $w_c(t)$ which is a function of the residual service time and remaining time slots available until finish time. Therefore an nPlug that requires more service or has fewer time slots remaining senses more often than an nPlug that may have either less residual service time or more available time slots. In this model, each nPlug's waiting time to complete service varies and is also affected by the randomness and length of the operational durations involved. In section V, we show experimentally that our model of contention window leads to similar waiting times on average as the operational duration or hold times decrease.

4) Rebound Effects. : Rescheduling customer loads over time may lead to rebound effects wherein existing peaks may grow larger and reoccur at alternate times. We now argue that nplugs do not lead to new increased peaks in the presence of background loads (which may not controlled). Nplugs attempt to schedule loads during off-peak slots between the start and end times specified by customers. Typically this duration will include the original times of appliance use. Randomized scheduling essentially adds an additional layer of load uniformly across all off-peak slots between these start and end times. Since the peak slots are excluded and loads are uniformly spread out, this is unlikely to results in new and larger peaks.

PNLB senses the grid as well as controls the rate at which appliances acquire service. Therefore it is sensitive to background load and attempts to distribute load between the start and end times such that valleys of low background load are filled and the total load remains uniformly below the overload threshold. However since appliances have a hard deadline, a peak may occur when a partial set of appliances with similar end times may have missed a chance to acquire service and are therefore scheduled close to their end times. However, such a peak is likely to be much smaller compared to a peak formed by all the appliances being scheduled together.

5) General GSMA scheduling and its performance: PNLB can be regarded as variant of "GSMA/OA(overload avoidance)" along with specified service deadlines, i.e. nPlugs acquire service at a certain rate in order to avoid overload whenever possible, as well as try to finish on time. Also during safewait state, they relinquish service if necessary to ensure grid reliability. However nPlugs do not proactively use any overload-detection(OD) protocol to actively drop-off in case they exceed capacity *after* acquiring service. The capacity can exceed even with OA in place due to the following reasons: Firstly, when nPlugs attempt to acquire service with a certain probability, the random number of these that actually acquire service may be more than the average. Secondly, since it may not be possible for nPlugs to determine in advance the voltage drop that will result from their appliance, the cumulative load introduced by nPlugs that actually acquire service may exceed capacity of the grid. The benefit of an OD-protocol is that it can allow only some nPlugs to drop-off instead of all, to reach the operating capacity.

In order to understand the performance of PNLB and general GSMA-based variants for demand dispatch, we now relax the requirement that appliances need to be serviced before a deadline and study the asymptotic performance of two best-effort GSMA variants: (i) p_j-persistent GSMA, and (ii) (p_j, p_ℓ)-persistent GSMA. We assume that the system has a total capacity to serve about k nPlugs/appliances simultaneously and a total of $m(t)$ nPlugs contend for service at any time t. We assume that nPlugs can sense the running capacity of the system $c(t)$, that is the number of nPlugs currently being serviced.

The p_j-persistent GSMA follows OA as in PNLB. At each time slot, after sensing that the system has free space, unserved nPlugs attempt to acquire service with probability p_j. If successful, they get served for a fixed number of slots and leave the system. The (p_j, p_ℓ)-persistent GSMA follows both OA and OD. As before, at each time slot, after sensing that the system has free space, unserved nPlugs attempt to acquire service with probability p_j. If successful however, they enter a 'temp' state and begin to receive service. At each time slot, the 'temp' nPlugs sense the grid load to check if the current capacity is $\leq k$. If so, they all move to 'joined' state where

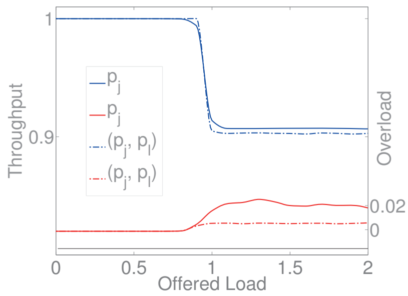


Fig. 6. Asymptotic performance of p_j-persistent and (p_j, p_ℓ)-persistent schemes for demand dispatch

they continue to receive service until finished. If not, each 'temp' nPlug leaves the system with probability p_ℓ. Note that both protocols do not prioritize nPlugs in any manner.

It is straightforward to see that at any time t, the optimal value of $p_j(t) = \max\{0, [k - c(t)]/m(t)\}$. Similarly, the optimal leaving probability for 'temp' nPlugs is $p_\ell(t) = \max\{0, [c(t) - k]/c_\ell(t)\}$, where $c_\ell(t)$ denotes the number of 'temp' nPlugs in the system. Figure 6 shows the asymptotic performance of both the protocols as a function of offered load, with optimal values of p_j and p_ℓ. The performance is measured using two metrics: (i) *Throughput* which gives the ratio of capacity used for service excluding any excess, over the capacity used by the optimal scheme and (ii) *Overload* which gives the ratio of excess capacity used over the capacity used by the optimal scheme. We see that the asymptotic throughput of both the protocols reaches close to 90%. As expected, the p_j-persistent protocol yields a slightly larger overload and hence a slightly better throughput. The plot shows that by choosing right values p_j and p_ℓ, the performance of decentralized scheduling schemes such as PNLB can be made close to that of centralized ones. Future work will modify the persistent GSMA protocols above into those that use a varying contention window and infer the optimal values of p_j and p_ℓ automatically.

Differences with networking protocols The above protocols differ from CSMA protocols used by the MAC-layer to share a communication channel in the sense that collisions could be tolerated to a certain extent. In networks, if more than one node attempts to acquire service, all the nodes fail due to a collision. However for a grid that can serve about k appliances, if $k + \delta$ acquire service, then some of the δ users can drop-off while the others can continue running.

V. Evaluation

In this section, we present the experimental evaluation of the nPlug algorithms. For this evaluation, we used data from an ongoing project [10], where plug-level energy monitors have been instrumented in a few homes in Bangalore and Chennai in India in order to collect the consumption profiles of household appliances. In addition to reporting the energy usage, these monitors report the line voltage and frequency every second. This time series is used by the nPlug analytics module to infer peak periods and detect supply-demand imbalance. We use voltage, frequency, and energy usage time series data collected in 2011 and 2012 for our experiments.

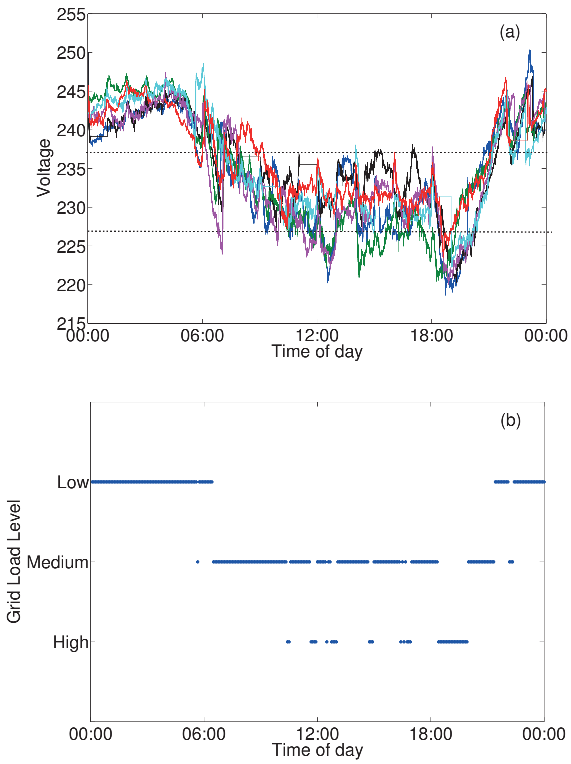


Fig. 7. (a) Voltage time series for 7 days at an Indian household and (b) the inferred grid load pattern

A. Inferring grid load using local sensing

Figure 7(a) plots the raw voltage time series corresponding to seven days (18th-25th Feb 2011), as sensed by a smart plug at one of the sockets in a household in Bangalore, India. The plot shows that (i) the line voltage varies over a wide range from $218 - 250$V and (ii) the voltage time series exhibits a similar trend every day with some differences. The voltage remains high at night when the load on the grid is low. It decreases after about 6AM in the morning when appliances are generally switched on. During the day it decreases and fluctuates as loads may have increased or decreased, mostly remaining within a range. It decreases further in the evenings after about 6PM when people generally return from work and electricity is used for lighting and other appliances. Thus the plot indicates that the times of high and low local voltage match well with the regular times of low and high load on the grid respectively.

Figure 7(b) plots the corresponding median grid load pattern that is inferred by the analytics module after the voltage time series is compressed using PAA (sections IV-B and IV-C1). The voltage values between $V_\ell = 228$V and $V_u = 238$V are classified as medium load, while those below and above are classified as high and low load respectively. The time period from 6:45 to 8:30PM is classified as one of the peak periods while 10:30PM to 6AM is classified as an off-peak period.

In order to understand how the local voltage measured at a household in Bangalore varies with the aggregate grid load, we analyzed the power data from KPTCL[2] that publishes the aggregate grid load in the state of Karnataka (where Bangalore is located). Figure 8(a) plots the average hourly local voltage along with the average hourly grid load for one of the sample days. We see that the local voltage is negatively correlated with the grid load i.e. as the hourly load rises, the hourly voltage drops. The Pearson's correlation coefficient between these two time series is $\rho = -0.8197$. We computed the correlation coefficient for all 7 days of voltage and grid data. The correlation coefficient varies from -0.7654 to -0.8993 with an average of -0.8197. The plots show that nPlugs could use local voltage as a load indicator to schedule appliances at coarser time scales using off-peak or randomized scheduling and contribute to peak load reduction.

When nPlugs use GSMA based scheduling approaches, they sense the local voltage continuously and schedule appliances under varying load conditions. In order to understand how loads introduced by different households affect the locally sensed voltage, we analyze the load and voltage measurements on same and different phases from neighboring households which are powered by the same transformer in Chennai, India.

We analyze the effect of AC (air-conditioner) loads as these are heavy and therefore contribute to a noticeable change in voltage. In Chennai, transformers and households are generally 3-phase wherein different sockets in a household may be on different phases. Fig. 8(b) shows the instantaneous load (watts) and voltage measured at the same socket that powers the AC. We see a clear inverse relationship between load and voltage. After each compressor cycle, when the load drops, the voltage rises. The voltage measured at the socket is also affected by background loads which may have been running in other homes on the same phase.

Fig 8(c) shows how the voltage varies on three different sockets in a household when socket 1 powers an AC. The blue curve shows the voltage on socket 1, while the red curve shows the voltage on socket 2(without load) on the same phase in the household. Although the voltage on sockets 1 and 2 vary in the same manner, they differ by a small amount (~ 2 volts) when the AC load is high. When the load reduces, both voltages are almost the same. Thus the proximity of load to the sensing point affects the measured voltage as there may be some losses on the wiring within the household as well. Lastly, the grey curve shows the voltage on a 3rd socket on a different phase that powers a refrigerator. We do not observe any noticeable effects of AC load on the voltage at socket 3 or the effect of refrigerator load on the voltages at sockets 1 and 2. Thus loads on different phases may not impact each other's voltage significantly.

Fig 8(d) shows how the voltage varies on three different sockets on the same phase when the first socket powers an AC, the 2nd socket is at the same household (without load) and the third socket is at a neighboring household(without load). Before the AC is switched on, all three voltages are approximately the same. As the AC's compressor cycle begins, all voltages drop and all voltages decrease as load increases. When the load rises, although the voltages vary together, there are minor differences between the voltages values ($\sim 2 -$

[2]http://110.234.115.69:8282/LoadCurveUpload/lcdownloadFl.asp

3 volts). We also see a short event when all three voltages increase and drop (marked by 3rd and 4th grey lines from left), which is not associated with the ACload. This may have been caused by a background loads on the same phase.

In conclusion, the locally sensed voltage variations are a function of loads at different locations in the distribution grid and their proximity to the sensing point. In general, the local voltage may remain low when the phase, the local transformer, the substation, or other grid assets may be overloaded.

B. Inferring supply-demand imbalance

The grid frequency decreases when the demand is higher than supply and vice versa. The frequency is continuously controlled by the grid operator by dispatching generators. However, under critical conditions, when there is a shortage of generation, nPlugs could help by not scheduling more load. We see that the frequency measurements vary in a very narrow range and exhibit a Gaussian-like distribution. About 95% of values lie within the 2-SD thresholds and the balance are classified as outliers by the analytics module. The 2-SD thresholds inferred from the data are close to the standard operating thresholds in India which are 49.4Hz and 50.1Hz. Therefore the frequency measurements could be used to detect imbalance conditions.

We observe that frequency shows low correlation with load and the average correlation over 7 days is -0.17. Frequency varies in very short time scales with load and may not be a good indicator of load at coarser time scales.

C. Decentralized Scheduling

In this section, we present the results of Monte-Carlo simulations conducted to evaluate the performance of decentralized scheduling schemes discussed in section IV-D: (a) Randomized scheduling and (b) GSMA-based PNLB. We compare the performance of these schemes against no scheduling (i.e. no nPlugs) and optimal centralized scheduling using direct load control. The performance is measured using three metrics: *Peak to Average Ratio*, *% Throughput*, and *% Over-utilization* (defined in section IV-D5).

We consider the following deferrable appliances in our experiments, which are commonly used in cities across India: (a) water heater (2.5KW) (b) washing machine (0.8KW) (c) water pump (2.5KW, used to pump up supplied or ground water) (d) Inverter (0.7 KW, used for power backup). In addition we consider (e) PHEV loads (1KW - 1.7KW) that may appear in future. These appliances are scheduled within the user specified time periods by randomized scheduling and PNLB to reduce peak loads.

We present the results for three scenarios documented in Table I: (1) Single peak from water heaters (2) Single peak from water heater plus varying background load (3) Multiple peaks from different appliances plus varying background load. (4) Multiple peaks from with varying background load as well as varying generation capacity with PHEV loads. In each case, the grid threshold capacity is set large enough so that the optimal centralized scheme can schedule the appliances without violating the threshold capacity. In addition the grid is assumed to have a spare generation capacity that is 50% of the threshold capacity. The throughput and overload of schemes is measured with respect to the threshold capacity.

In order to establish the correspondence between voltage and grid capacity, we assume that the grid voltage varies between $V_{min} = 225$V and $V_{max} = 255$V with a safe operating region from $V_\ell = 228$ to $V_u = 249$V. The voltage of the grid at any time t is computed as

$$v(t) = V_{min} + (1 - \frac{P(t)}{P_{max}}) \times (V_{max} - V_{min}) \qquad (3)$$

where $P(t)$ is the load in the grid at time t and P_{max} is the threshold capacity. Note that the above method to compute voltage from grid load may not be completely realistic. The voltage change in a household is a function of both load in the household as well as the grid and is hard to estimate. Moreover it depends on several factors such as the distance of the household from the transformer and so on. Therefore our experiments evaluate the performance of scheduling algorithms assuming the simplified model of grid load mentioned in (3). For GSMA-based PNLB, the time slot length τ, that is used to sense the grid at regular intervals, is set to 1min.

Scenario 1 The first scenario is designed to capture common domestic demand patterns observed in major Indian cities during the early morning hours. Since most households switch on their heater for about 30 min, this demand induces a peak during morning hours [11], [12], [13]. Figure 9(a) shows such a peak when 100 water heaters are switched on between 6:15 and 7AM. To reschedule this load, note that users are generally insensitive to the exact time at which the heaters are switched on as long as hot water is available by a certain time. Also since water heaters have insulation, water once heated remains useable for a few hours. Therefore we assume that users specify 4:00 AM as the earliest start time, 7:00 AM as the latest end time, and duration as 30min. For this scenario, the threshold capacity P_{max} is set to 42.5KW since this is the minimum capacity needed to operate 100 water heaters for half an hour each, so that all the heaters finish their operation within 3 hours. Figure 9(b) and (c) show the results for one sample run of randomized scheduling and PNLB. The average and standard deviations for 20 runs are shown in Table II.

We see that randomized scheduling provides a good distribution of the load and as expected some overload. PNLB mimics the behavior of the optimal centralized scheme but with a small peak towards the end. This occurs since appliances that were not scheduled earlier are switched on towards the end so that all appliances finish on time. The throughput of PNLB remains above 90% with low overload. Thus the peak load reduces significantly by using nPlugs with randomized scheduling or PNLB.

Scenario 2 Having established the benefits of decentralized scheduling, we now evaluate the performance of scheduling schemes in the presence of varying background load. The background load corresponds to the domestic loads that are either non-deferrable or ones that do not use nPlugs. This load is assumed to have a mean amplitude of 50% of the peak load.

Figure 9(d)-(f) shows the results when water heaters are scheduled by nPlugs in the presence of varying background load. Again, we observe that randomized scheduling distributes the load uniformly over time. However it does not

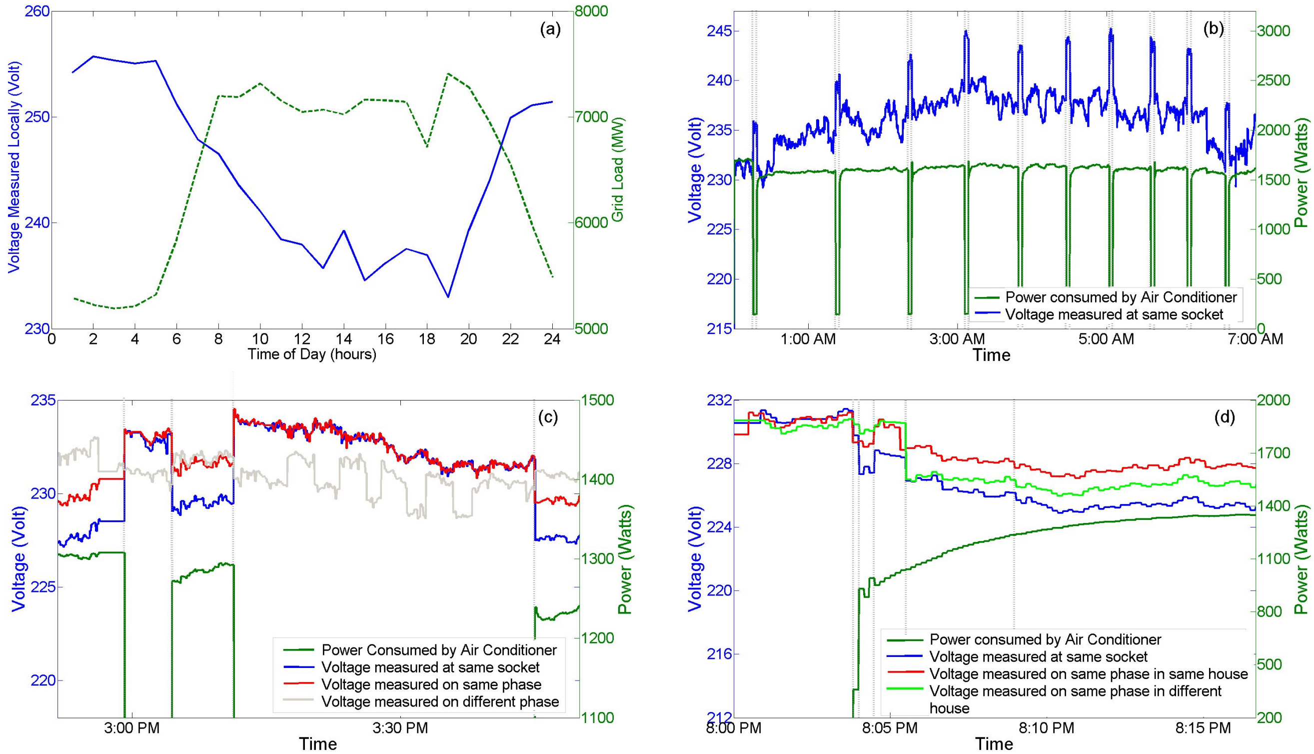


Fig. 8. (a) Correlation of grid load and local voltage, (b),(c),(d): Power consumed by AC and its effect on Voltage at different sockets on same and different phases at two neighboring households.

TABLE I
SUMMARY OF EXPERIMENTAL SCENARIOS (THE USER PREFERENCES ARE SPECIFIED AS [EARLIEST START TIME, LATEST END TIME, OPERATIONAL DURATION(MIN), <HOLD TIME> (MIN)])

Scenario No.	Appliance types	Number of appliances	Additional variable load	User Preferences
1	Water Heaters	100	No	[4:00, 7:00, 30]
2	Water Heaters	100	Yes	[4:00, 7:00, 30]
3	Water Heaters, Water Pumps, Washing Machines, Inverters	200	Yes	[4:00, 7:00, 30], [5:00, 7:00, 20] [6:00, 8:15, 40], [6:40, 8:15, 25]
4	Four types of Storage Appliances (PHEVs, Inverters etc.)	200	Yes	[20:00,6:00,420,30],[20:00,4:00,400,20] [20:00,6:00,480,15],[20:00,4:40,360,25]

TABLE II
SUMMARY OF MONTE-CARLO SIMULATIONS FOR DIFFERENT DECENTRALIZED SCHEDULING ALGORITHMS BASED ON THREE METRICS: *Peak to Average Ratio*, *% Throughput*, AND *% Over-utilization above threshold generation capacity)*.

Scenario No.	Peak to Average Ratio			% Throughput			% Over-utilization		
	w/o nPlug	Random	PNLB	w/o nPlug	Random	PNLB	w/o nPlug	Random	PNLB
1	1.78±0.1	1.48±0.2	**1.41±0.1**	23.9±1.7	82.3±6.7	**93.2±2.1**	62.2±2.3	13.1pm1.1	**4.1±0.8**
2	1.79±0.2	1.51±0.1	**1.33±0.1**	78.2±1.9	87.3±3.4	**94.1±2.4**	19.7±1.6	5.3±1.2	**2.1±0.7**
3	1.78±0.1	1.52±0.1	**1.29±0.1**	80.1±1.8	88.2±3.8	**94.3±2.3**	12.4±2.6	4.3±1.2	**1.2±0.8**

TABLE III
PERFORMANCE OF PNLB AND PNLB WITH PREEMPTION (PNLB+) FOR STORAGE-BASED APPLIANCES WITH LONG SERVICE TIMES WITH VARYING LOAD AND VARIABLE GENERATION, BASED ON THREE METRICS: *Peak To Average Ratio*, *% Throughput* AND *% Over-utilization*.

Scenario No.	Peak to Average Ratio			% Throughput			% Over-utilization		
	w/o nPlug	PNLB	PNLB+	w/o nPlug	PNLB	PNLB+	w/o nPlug	PNLB	PNLB+
4	1.59±0.03	1.56±0.16	**1.31±0.11**	83.8±2.1	84.1±1.7	**92.3±1.6**	10.2±0.8	9.4±1.3	**2.2±0.7**

efficiently use available grid capacity since it does not sense the running load in the grid. On the other hand PNLB that uses a GSMA-approach, senses the running load and therefore uses the varying capacity more efficiently, thus yielding a better throughput and lower overload.

Scenario 3 The third scenario is designed to mimic the demand pattern in metropolitan cities where the use of multiple high power electrical appliances is more common [11]. Figure

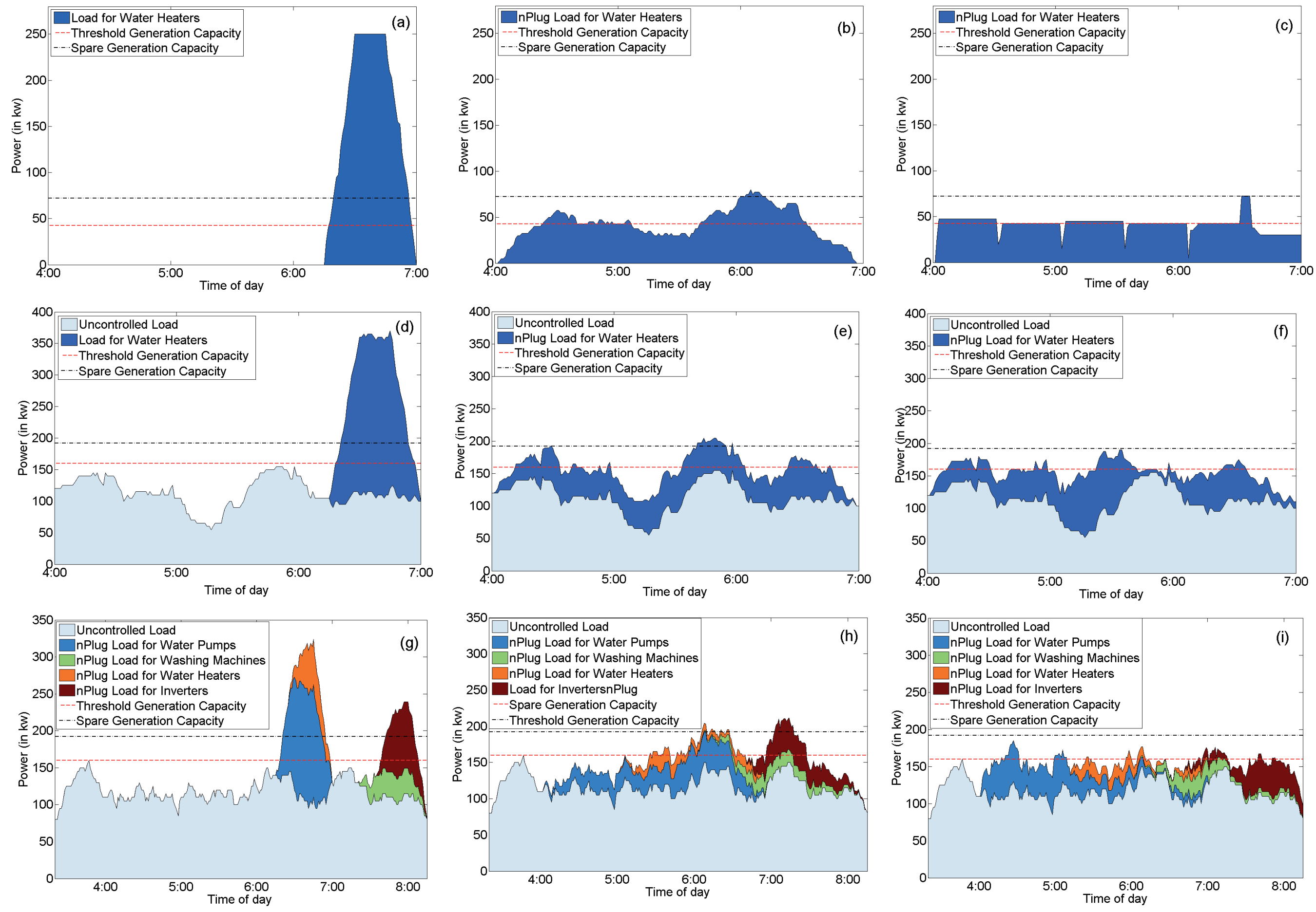


Fig. 9. Performance of different scheduling schemes: rows top to bottom: Scenarios 1 to 3. Columns left to right: Base case without nPlug, Randomized scheduling, and PNLB. (a)-(c): Scheduling of of 100 water heaters using nPlug. (e)-(f): Scheduling 100 water heaters in the presence of varying background load. (h)-(i): Scheduling 200 appliances of different types of appliances in the presence of varying background load. Randomized scheduling and PNLB contribute to peak load reduction and load-leveling.

9(g) shows a demand pattern that was constructed by considering appliance ratings and commonly occurring appliance mix in metropolitan households where multiple appliances are switched on simultaneously resulting in multiple peaks.

For scheduling using nPlugs, different appliances are assumed to have overlapping start and end times and different operational durations as shown in Table I. Figure 9(h)-(i) plot the results for one sample run of randomized scheduling and PNLB. Table II presents the mean and standard deviations over 20 runs. We see that both schemes contribute to peak load reduction and load-leveling even when different appliances with different user constraints are attached to nPlugs. PNLB allows nPlugs attached to different appliances to use the available capacity efficiently even as appliances are switched on and off and the grid load varies.

Scenario 4 The fourth scenario is designed to mimic future demand and supply patterns in the presence of storage based loads such as PHEVs/inverters, variable generation (e.g. wind, solar), and variable background loads. Figure 10(a) shows a demand pattern that was constructed by considering four different storage appliances of varying service durations in the presence of varying load and variable generation. Since the load is not responsive to grid conditions, we observe multiple overload peaks. Figure 10(b) shows the results when the storage appliances are controlled by nPlugs without preemption i.e. loads once scheduled cannot be interrupted. We see that situations of overload reduce. However since appliances have long operational duration, they become unresponsive after being scheduled and a few overload peaks still occur. Figure 10(c) shows the results when storage appliances are scheduled by PNLB with preemption by specifying the hold times (referred as PNLB+). We observe that all appliances are serviced by nPlugs without overloading the grid. Table III shows the performance as measured using peak to average ratio, throughput, and overload metrics.

Fairness In PNLB, the rate at which nPlugs sense the grid varies with the residual service time and the time remaining to finish service. In order to determine if the algorithm is fair to different users, we conduct experiments to estimate the waiting time before each appliance is serviced. We consider a scenario with long operational duration and use PNLB with preemption. We consider 200 appliances, each of 1kW powered by a grid of capacity 100 kW i.e the grid can support half the appliances at any point in time. In a fair scheduling scheme, each appliance stays on and off the grid for alternate durations of hold time, thus waiting for one hold time before each service. Table IV

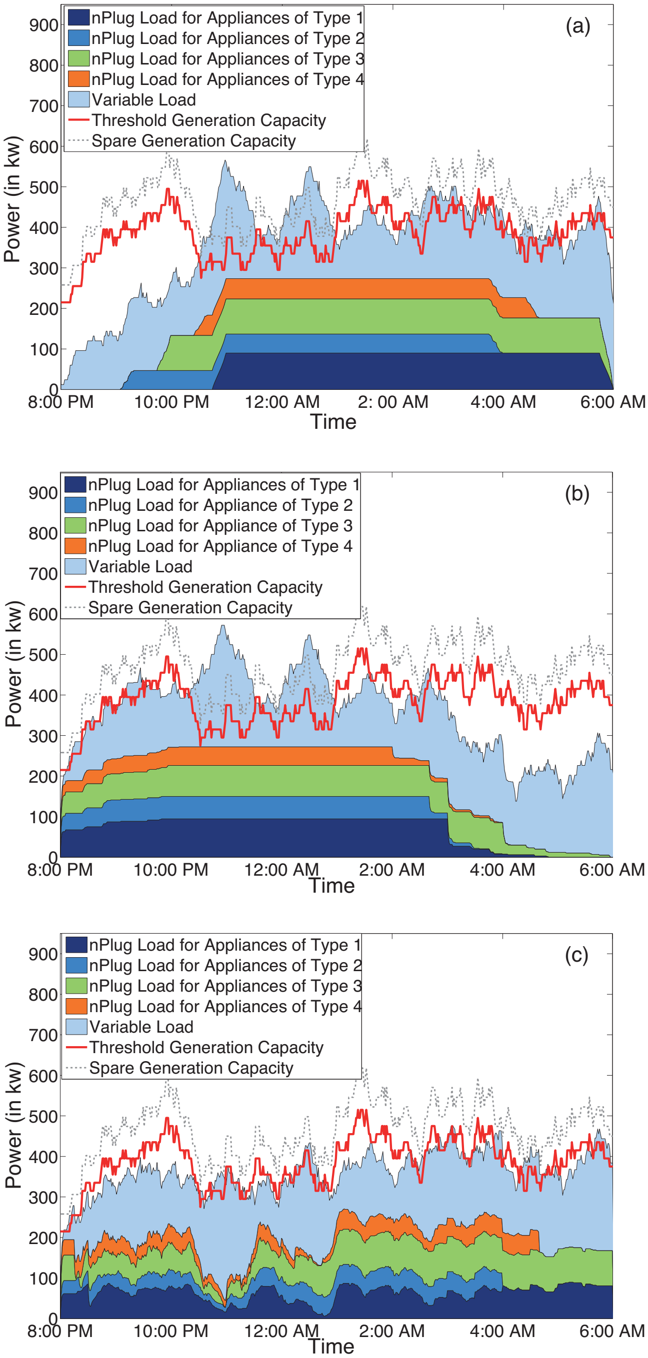


Fig. 10. Performance of PNLB with storage-based loads of long charging duration such as PHEVs/Inverters: (a) Base case without nPlug, (b) PNLB without preemption and (c) PNLB with preemption where loads can be interrupted by specifying hold times.

shows the average waiting times of appliances along their standard deviations as a function of hold time duration. We observe that as hold times reduce, the deviation from average waiting time reduces and the waiting times approach the hold times implying fairness.

VI. Related Work

Demand Side Management or Demand Response(DR) [14] is essentially a mechanism for inducing consumers to alter their consumption patterns in response to changes in supply so that available capacity may be shared efficiently. These demand changes are usually induced through variable pricing, financial (dis)incentives, and explicit or direct load control. Although these are more popular in the power sector, they are applied in various sectors including transportation (e.g. congestion pricing) as well. Several DSM systems and programs have been proposed for reducing the peak power loads and some of these are even operational today. In this section, we review devices used for both the centralized and decentralized demand management schemes.

TABLE IV
Average Waiting Time for different hold times

Hold Time	Average Waiting Time
20	18.72 ± 0.79
10	9.76 ± 0.18
5	4.96 ± 0.03
1	$0.9986 \pm \sim 0$

One of the earliest proposed grid-friendly appliances is Frequency Adaptive, Power-energy Re-scheduler (FAPER) invented by Schweppe [15]. FAPER senses grid frequency and reschedules the power flow to a load on the basis of deviations in frequency. As explained previously, frequency alone may not be sufficient to sense peak loads. Moreover, FAPER does not consider consumer's preferences while scheduling loads. For example, on a particular day, if the load on the grid is high during a time period, consumers may not be able to run their appliances during this period if only the grid conditions are considered. Responsive Load Controller from RLtec [16] uses an approach identical to that of FAPER and has similar shortcomings. Another example is the Grid-Friendly controller from PNNL [17], that can be installed in refrigerators, air conditioners, or other household appliances. It monitors the power grid and turns appliances off for a few seconds to minutes in response to grid overload. RLtec and Grid-Friendly devices are not standalone devices and must be incorporated into the appliances. Although new appliances could be fitted with such controllers, it may not be possible to retrofit millions of appliances already in use. Moreover, these controllers react only to grid conditions and do not support a mechanism to proactively schedule appliances to reduce load or as per consumer convenience. Nest [18] is a thermostat management system that learns the preferred temperature settings of the consumer and maintains the room temperature accordingly. But, Nest can manage only heating and cooling loads. It is not a appliance-level schedule management device. Peaksaver [19] is a smart thermostat that allows utilities to cycle central air conditioners and reduce their run time - typically during hot weekdays of summer - when the load on the grid is usually high. Peaksaver requires centralized control and is designed to work only with air conditioners and not with other loads that can be time shifted. Bluepods from Voltalis[20] are devices that plug into home electrical panels and are controlled over the web. During peak demand, a signal is sent to Bluepods to turn off air conditioners. Williamson et al. have proposed Distributed Intelligent Load Controllers (DILC) [21] to mitigate the power imbalance due to intermittent renewable energy sources. Similarly, Barker et al have proposed SmartCap [22] for reducing residential electricity demand during peak hours. SmartCap is based on a home gateway that receives information from multiple potential sources, including real-time electricity prices and demand-response signals from the grid, generation data from on-site renewables, and consumption data from each household load. These information sources inform the gateway's load scheduling policy. Bluepod, DILC and SmartCap are solutions that depend on external data sources and require network infrastructure to communicate with those sources. [23] has proposed a distributed control mechanism that still requires network communication, even though minimal, for managing residential loads.

Unlike above systems, nPlug provides an inexpensive and autonomous load scheduling mechanism that can minimize peak loads while respecting consumers preferences.

VII. Conclusions and Future Work

There has been an increasing interest in DSM strategies to address the peak load problems faced by utilities all over the world. In this work, we present *nPlug*, an autonomous load controller that uses local sensing and control techniques to alleviate peak load on power grids. It uses voltage sensing to identify peak and off-peak periods of the grid. nPlugs time-shift the attached loads to off-peak periods while respecting the end user preferences and grid load conditions. They do not require any communication infrastructure nor any changes to the appliance or grid. They are simple, affordable, and scalable and could be used in developing as well as developed countries. We describe the high level architecture and the design details of nPlugs. Using preliminary voltage measurements collected at a household, we showed that line voltage is a good indicator of grid load and presented simple analytics techniques to infer peak and off-peak periods from voltage time series. We presented novel decentralized scheduling algorithms - randomized scheduling and GSMA-based PNLB+ that is inspired by CSMA protocols in networks. Our experimental results show that both these algorithms could be used by nPlugs to achieve significant peak load reduction and load-leveling in the presence of varying grid load. However, we need to study the impact of battery lifetimes with varying charging/discharging cycles.

nPlug has applications in developed nations as well. Energy demand is expected to increase in future, especially with the introduction of heavier loads such as electric vehicles (EVs). In such a scenario, construction of new power plants and resizing grid assets may be deferred with the help of nPlugs that facilitate demand response. For example, nPlugs may be used to alleviate load on the local transformer when there are a large number of EVs in a service area. The resulting demand reduction can help reduce the dependence on expensive power during peak periods. Furthermore in sparsely populated areas such as the country side where the cost of communication infrastructure per customer is high, nPlugs can help provide a less expensive alternative to achieve demand response.

We are considering several future extensions to our work. Firstly, we plan to study how sensing additional power-system parameters such as power factor could be useful in improving the observability of grid conditions. Second, we will study how nPlugs can be used to manage reducible loads such as Air Conditioners whose power consumption can be reduced during the peak hours. Third, we will analyze how nPlugs can be incorporated into appliances so that their power consumption can be finely modulated without simply turning them ON/OFF.

A few questions also remain unanswered. Even though nPlugs are inexpensive, the economic incentives for end users to use them is not clear. It might require legislative changes to encourage appliance manufacturers to embed nPlug-like functionality into deferrable loads. If incorporated, appliances can be both grid and user friendly with minimal user intervention. If a large number of nPlugs are deployed in the field, the load curves used by distribution companies may also need to be altered and that in turn could alter the generation portfolio.

References

[1] T. Ganu, D. P. Seetharam, V. Arya, R. Kunnath, J. Hazra, S. A. Husain, L. C. D. Silva, and S. Kalyanaraman, "nplug: A smart plug for alleviating peak loads," in *Third International Conference on Future Energy Systems, e-Energy*, May 2012.

[2] Central Electricity Authority, Ministry of Power, Government of India, "Load generation balance report 2011-12."

[3] KPMG, "India electricity market outlook 2008," 2008.

[4] R. Tongia, "A smart solution to power shortage," *Business Standard*, July 2011.

[5] Ministry of Power, "Energy conservation act," 2010.

[6] R. Tongia, "What the smart grid means-and does not mean-for india," *IEEE Smart Grid*, July 2011.

[7] Juxt Consulting, "Internet usage behavior & preferences of indians," http://www.juxtconsult.com/, 2011.

[8] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, ser. DMKD '03. New York, NY, USA: ACM, 2003, pp. 2–11.

[9] V. Barnett and T. Lewis, *Outliers in Statistical Data*. John Wiley and Sons, 1994.

[10] T. Bapat, N. Sengupta, S. K. Ghai, V. Arya, Y. B. Shrinivasan, and D. Seetharam, "User-sensitive scheduling of home appliances," in *Proceedings of the 2nd ACM SIGCOMM workshop on Green networking*, ser. GreenNets '11, 2011, pp. 43–48.

[11] G. Krishnan K, M. Duraisamy, and L. S. Ganesh, "Energy usage in indian urban households: The role of renewable energy technologies," 2003.

[12] I. R. Pillai and R. Banerjee, "Methodology for estimation of potential for solar water heating in a target area," *Solar Energy*, vol. 81, no. 2, pp. 162–72, 2006.

[13] C. More, S. J. Saikia, and R. Banerjee, "An analysis of maharashtra's power situation," 2008.

[14] U.S. Department of Energy, "Benefits of demand response in electricity markets and recommendations for achieving them, 2006." http://eetd.lbl.gov/ea/emp/reports/congress-1252d.pdf.

[15] F. Schweppe, "Frequency adaptive, power-energy re-scheduler," Patent no US4317049, Feb. 1982.

[16] "Rltec," http://www.rltec.com.

[17] Pacific Northwest National Laboratory, "Grid friendly appliance controller."

[18] "Nest, the learning thermostat," http://www.nest.com.

[19] "Peaksaver," https://www.hydroottawa.com/.

[20] "Blue pods," http://www.voltalis.com.

[21] W. Ian, K. Ruth, T. D. Philip, R. David, T. D. Stathis, and N. D. Aristomenis, "Intelligent load control strategies utilising communication capabilities to improve the power quality of inverter based renewable island power systems," in *International Conference RES for Island, Tourism & Water*, 2003.

[22] S. Barker, A. Mishra, D. Irwin, P. Shenoy, and J. Albrecht, "Smartcap: Flattening peak electricity demand in smart homes," in *IEEE PerCom 2012*, March 2012.

[23] P. Srikantha, S. Keshav, and C. Rosenberg, "Distributed control for reducing carbon footprint in the residential sector," in *IEEE SmartGridComm*, November 2012.

Tanuja Ganu is a Research Software Engineer at IBM Research - India. Prior to joining IBM, she completed her Masters in Computer Science at Indian Institute of Science (IISc), Bangalore. Her research interests are in machine learning, optimization and smart grids.

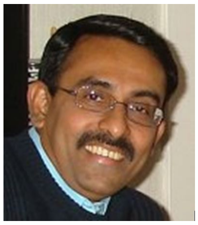
Deva P. Seetharam leads IBM India's Smarter Energy research initiatives. Previously, he led France Telecom's electronic paper display research group in Boston. He has also worked on various embedded wireless networking technologies including active RFID and wireless sensor networks. He holds a MS degree from the MIT Media Laboratory and a BE degree from the Bharathiar University.

Vijay Arya is a Research Staff Member at IBM Research - India. Prior to joining IBM, he worked at National ICT Australia (NICTA) and completed his doctoral studies in Computer Science from INRIA Sophia Antipolis, France. His research interests include measurements and modeling, computer networks, and smart grids.

Jagabondhu Hazra is a Research Staff Member at IBM Research - India. Prior to joining IBM, he received his Phd from IIT Kharagpur, India and worked as a post doctoral researcher at SUPELEC, France. His research interest includes smart grid, network congestion management, blackout prediction, HVDC, HVAC, renewable energy, etc.

Deeksha Sinha is pursuing undergraduate studies in Electrical Engineering at IIT Bombay, India with specialization in Communications and Signal Processing. She is a recipient of Honda Young Engineer and Scientist Award (2012) and Temasek Foundation LEaRN Award (2012).

Rajesh Kunnath is the founder of electronic hardware design and consulting firm Radio Studio, India. He has Masters degree from Indian Institute of Technology (IIT), Madras.



Liyanage Chandratilake De Silva received BSc Eng.(Hons) degree from the University of Moratuwa Sri Lanka in 1985, M.Phil. Degree from The Open University of Sri Lanka in 1989, MEng. and PhD degrees from the Univ. of Tokyo, Japan in 1992 and 1995 respectively. Currently he is an Associate Professor at University of Brunei, Brunei Darussalam.

Saiful A. Husain received BSc Ed degree from Universiti Brunei Darussalam in 1995, MSc Degree from University of East Anglia, Norwich, England (2000) and PhD degree in Mathematical Sciences from the Australian National University, Canberra, Australia in 2005. He is currently a senior lecturer, Department of Mathematics, Faculty of Science, Universiti Brunei Darussalam (UBD), Brunei and the leader of the Modelling & Simulation Research cluster, Universiti Brunei Darussalam, since 2009. Currently, he is appointed as the UBD—IBM Technical Coordinator as well as Deputy Director of the Centre.

Shivkumar Kalyanaraman received the B.Tech. degree in computer science from the Indian Institute of Technology, Madras, India, in 1993, the M.S. and Ph.D. degrees in computer and information sciences from the Ohio State University, Columbus, in 1994 and 1997, respectively. He also received the Executive M.B.A. degree from Rensselaer Polytechnic Institute, Troy, NY, in 2005. He is a Senior Manager with IBM Research India. Previously, he was a Professor with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute. His research in IBM is at the intersection of emerging wireless technologies, smarter energy systems, and IBM middleware and systems technologies. Dr. Kalyanaraman is an ACM Distinguished Scientist. He is a visiting Professor at Universiti Brunei Darussalam.