nPlug: A Smart Plug for Alleviating Peak Loads

Tanuja Ganu
IBM Research, India

Deva P. Seetharam
IBM Research, India

Vijay Arya
IBM Research, India

Rajesh Kunnath
Radio Studio, India

Jagabondhu Hazra
IBM Research, India

Saiful A. Husain
Universiti Brunei Darussalam

Liyanage Chandratilake De Silva
Universiti Brunei Darussalam

Shivkumar Kalyanaraman
IBM Research, India

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 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. In this work, we present *nPlug*, a smart plug that sits between the wall socket and deferrable loads such as water heaters, washing machines, and electric vehicles. nPlugs combine real-time sensing and analytics to infer peak periods as well as supply-demand imbalance and reschedule attached appliances in a decentralized manner to alleviate peaks whenever possible. They do not require any manual intervention by the end consumer nor any enhancements to the appliances or existing infrastructure. 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.

Categories and Subject Descriptors

B.m [**Hardware**]: Miscellaneous; E.4 [**Data**]: Coding And Information Theory; F.m [**Theory of Computation**]: Miscellaneous; I.6 [**Computing Methodologies**]: Simulation And Modeling

General Terms

Algorithms, Design, Experimentation

Keywords

Smart Plug, Demand Response, Demand Side Management, Peak Loads, Scheduling, Multiple Access

e-Energy 2012, May 9-11 2012, Madrid, Spain.

1. 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 [7]. 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 [10].

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 such as diesel or open-cycle gas/hydro turbines. 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% [19].

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 [13]. 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 [20]. One of the disadvantages of existing DSM strategies such as direct load control is their centralized nature which requires communication between appliance-level monitors and a central controller at the utility. Since monitors require communication capabilities, it increases their cost. More importantly, this requires a communication infrastructure between the utility and appliances which is expensive to deploy. Although cellular communication is inexpensive in India, existing infrastructure will need a capacity upgrade to support household appliances as well. An Internet based solution may not be widely applicable as only 11.3% of Indian households have access to Internet [9].

In this paper, we present a decentralized DSM system based on smart plugs called nPlugs 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 simple data mining algorithms to identify the peak and off-peak periods of the grid. It runs the attached load(s) during off-peak periods as much as possible without violating user-specified constraints. To ensure grid and appliance safety, it avoids scheduling appliances during periods of supply-demand imbalance. Furthermore each nPlug runs a decentralized load rescheduling algorithm that contributes to peak load reduction by distributing the loads over time.

The key contributions of this paper are:

- Design of a low-cost standalone smart plug that can schedule appliances during off-peak periods. It neither requires any communication infrastructure nor any changes to the appliance or grid. It can work with common deferrable loads such as water heaters, washing machines, and electric vehicles.
- Design of simple and effective data mining algorithms to determine peak and off-peak periods as well as supply demand imbalance, that can run on low-cost microcontrollers.
- Design of novel decentralized load scheduling algorithms that contribute to peak load reduction and load-leveling by spreading the joint load efficiently under varying grid load conditions.
- Experimental evaluation of above-mentioned algorithms.

The rest of the paper is organized as follows. Section 2 presents the theoretical 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 3. The data mining algorithms used to identify peak and off-peak periods as well as load scheduling algorithms used by nPlugs are presented in section 4. Experimental evaluation of our algorithms is presented in Section 5. Section 6 presents related work and finally section 7 concludes with a discussion about future work.

2. POWER SYSTEMS BACKGROUND

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

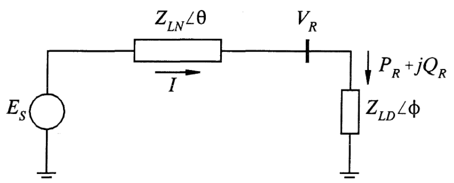

Figure 1: A two-bus power system

2.1 Inferring grid load from voltage

Figure 1 shows a simple "power system" wherein a load is connected to a generator using a transmission line. $\tilde{E_S}$ is the generator voltage, $\tilde{V_R}$ is the load voltage, $\tilde{Z}_{LN}$ is the transmission line 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

$$\tilde{I} = \frac{\tilde{E_s}}{\tilde{Z}_{LN} + \tilde{Z}_{LD}}$$

$$\text{where} \quad \tilde{Z}_{LN} = Z_{LN}\angle\theta = Z_{LN}cos\theta + jZ_{LN}sin\theta$$

$$\tilde{Z}_{LD} = Z_{LD}\angle\phi = Z_{LD}cos\phi + jZ_{LD}sin\phi$$

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

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

Therefore the magnitude of load voltage V_R is:

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

$$= \frac{Z_{LD} \times E_S}{\sqrt{(Z_{LN}cos\theta + Z_{LD}cos\phi)^2 + (Z_{LN}sin\theta + Z_{LD}sin\phi)^2}} \quad (1)$$

Since the source voltage E_S and transmission line 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. Therefore as the collective load on the grid increases, the corresponding voltage drop can be sensed at households. In section 5, we shall plot the variation in voltage V_R measured at a household and show how it drops during peak hours.

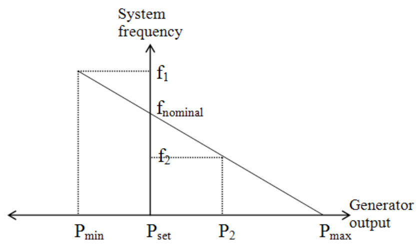

Figure 2: Load-frequency control characteristics

2.2 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 Figure 2. The plot shows that when the 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 shoots up.

Although frequency is a good indicator of imbalance, our measurements show that it varies continuously and may not be sufficient to identify grid load 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. Even though the power systems theory explained in this section is well-known, to the best of our knowledge, none of the existing systems learn the voltage and frequency patterns to derive load schedules that can help reduce peak loads.

3. NPLUG HARDWARE DESIGN

Figure 3 shows an initial prototype of nPlug. The hardware design is based on cost-performance trade-offs.

Figure 3: nPlug Prototype

As shown in Figure 4, the hardware modules (drawn in blue) of nPlug are user controls, grid sensors, 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**
  Voltage sensing is achieved with 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 Analog to Digital Converter (ADC) can cover the input voltage range (110 V - 350V) [1]. Since nPlug is only required to identify voltage changes due to peak demand, the measurement must be accurate between 185 V and 250 V ($\approx \pm 10\%$ of nominal voltage). Voltage calibration at the lower limit of decision making, at around 185V addresses ADC resolution issues, providing an overall accuracy better than 99% after calibration.
- **Frequency Sensing**
  Frequency is sensed using a current-limiting resistor directly connected from the phase to the microcontroller input. Frequency is determined by counting the number of zero-crossing positive transitions that occur in one second.
- **Real Time Clock (RTC)**
  Scheduling decisions are based on the accuracy of clock. Since there is no network interface for a nPlug to synchronize its internal clock with an accurate source of time, an accurate yet power-efficient RTC (DS1307) with battery backup (coin cell - CR 2032) is included onboard.

4. NPLUG SOFTWARE COMPONENTS

Figure 4 shows the high level architecture of an nPlug. An nPlug has four software components: (i) UI manager, (ii) Data manager, (iii) Analytics module, and (iv) Load

[1]In India, we have observed line voltages fluctuating between 150 V to 300 V.

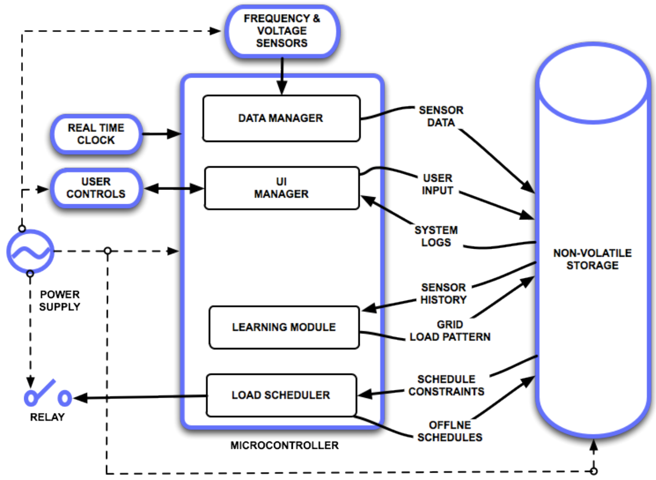

Figure 4: nPlug: System Overview

scheduler. The following sections explain the functionality of each component.

4.1 UI Manager

The UI Manager accepts three 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.

4.2 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 [12] 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 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 5 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).

4.3 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.

4.3.1 *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.

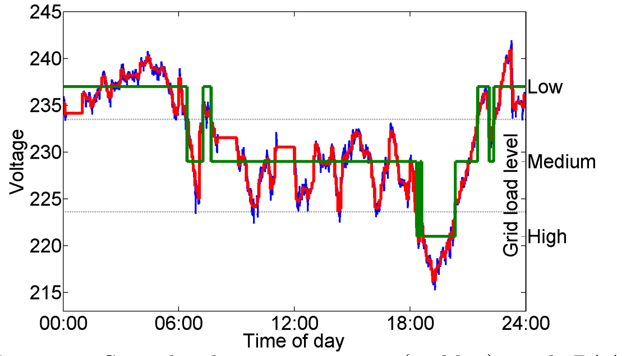

Figure 5: Sensed voltage time series (in blue), with PAA data compression (in red) and the grid load pattern (in green).

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) [12]. 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 $\beta_1, \beta_2, \ldots, \beta_{k-1}$ 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 does not follow a Gaussian distribution and yields a distribution that is skewed towards one side. Therefore we use domain knowledge and identify lower and upper break-points using the following heuristic:

$$V_\ell = \min(V') + 0.3 \times (\max(V') - \min(V'))$$
$$V_u = \min(V') + 0.7 \times (\max(V') - \min(V'))$$

Thus values less than or equal to ($\leq$) V_ℓ are classified as high load level, greater than or equal to ($\geq$) V_u as low load level, and the values in between as medium load level. The resulting three-alphabet time series is called as the grid load pattern $\bar{V}$. In figure 5, the grid load pattern is shown using the green color. The dotted lines in the figure indicate the two break points V_ℓ and V_u.

Let $\bar{V}^1, \ldots, \bar{V}^c$ denote the grid load pattern for previous c days. 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 $\hat{v}_t = median_{i=1}^{c}(\bar{v}_t^i)$. All times periods of high load in the median grid load pattern are regarded as peak periods and the balance as off-peak periods.

4.3.2 *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 2. 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 [5] to compute the thresholds of normal operating frequency. The lower and upper operating thresholds, f_ℓ and f_u, are computed as:

$$f_\ell = f_\mu - 2 \times f_\sigma$$
$$f_u = f_\mu + 2 \times f_\sigma$$

where f_μ and f_σ are the mean and standard deviation of sensed frequency data. Since mean and standard deviation 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 by using f_t. Figure 6 shows the frequency time series along with the thresholds f_ℓ and f_u.

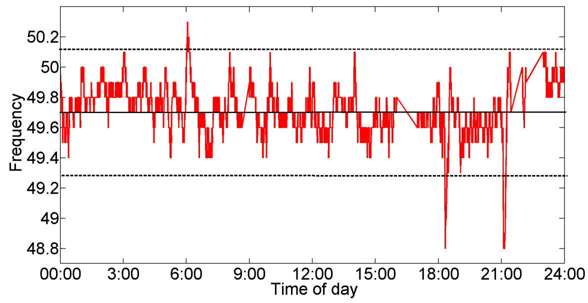

Figure 6: Sensed frequency time series for a day at a household. The mean, f_μ, and 2-SD operating frequency thresholds, f_ℓ and f_u, are indicated using the solid and dashed lines respectively.

4.4 Scheduling

This section discusses various strategies used by nPlugs to schedule deferrable loads by taking into account user specified constraints as well as grid load conditions. The scheduling algorithms used by nPlugs contribute to peak load reduction and load-leveling without any centralized control.

An nPlug receives the earliest start time S_t, latest end time E_t, and the operational duration d of the appliance from the end user. 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 time slots and $D = d/\tau$ be the number of contiguous slots needed by the appliance to finish work (loads are assumed to be non-splittable). We now discuss three scheduling schemes that may be used by nPlugs: (i) Off-peak scheduling, (ii) Randomized scheduling, and (iii) GSMA Scheduling.

4.4.1 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 4.3.1, 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 continuous slots. The 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.

4.4.2 Randomized Scheduling

Although Off-peak scheduling is useful and easy to implement, 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 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 average μ_t can be compared by using Chernoff bound. 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.

4.4.3 GSMA (Grid-sense multiple-access)

Both off-peak and randomized scheduling schemes above help reduce peak loads. However they cannot respond to fluctuations in demand or supply since they are agnostic to the running load in the grid. In GSMA-scheduling, which is inspired from multiple-access protocols in 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 the pseudocode of 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 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 4.3.1). The first step mimics the behavior of the optimal scheduling scheme (section 4.4.2) and the second step ensures that users react to varying grid load whenever possible.

Algorithm 1 Probabilistic Negative Linear Back-off (PNLB)

Input: S_t, E_t, τ, d, V_u, V_ℓ, f_ℓ
1: $n = (E_t - S_t)/\tau$, $D = d/\tau$, $t \leftarrow 0$
2: **if** $t \geq n - D$ **goto** step 15
3: $w_c \leftarrow \frac{n-t}{D}$ % 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: $\quad p \leftarrow 0$
9: **else if** $(v_c > V_u)$ **then**
10: $\quad p \leftarrow 1$
11: **else**
12: $\quad p \leftarrow (v_c - V_\ell)/(V_u - V_\ell)$
13: **end if**
14: **if** **rand**$(0, 1) < p$ **and** $f_c > f_\ell$ **then**
15: $\quad$ **switch**(ON) % acquire service with probability p
16: $\quad t \leftarrow$ **safewait**(D, t) % switch ON for D time slots
17: $\quad$ **switch**(OFF); **exit**
18: **else**
19: $\quad t \leftarrow$ **wait**$(1, t)$; **goto** step 2
20: **end if**

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

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 $k = \frac{m}{(n/D)}$. 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 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 $w'_c = n'/D$, so that m'/w'_c will attempt to acquire service in a future slot. As time progresses w_c gradually decreases so that the remaining users sense the grid at the right rate to finish on time. If an appliance has failed to acquire service in all slots, it is switched ON before its finish time. After an appliance is switched ON, nPlugs use "safewait" where they sense the grid voltage and frequency regularly and if necessary switch OFF the appliance to ensure grid reliability.

4.4.4 Performance of GSMA scheduling

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 is hard 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 they continue to receive service until finished. If not, each 'temp' nPlug leaves the system with probability p_ℓ. Note that both protocols are fair and do not prioritize nPlugs.

It is not hard to see that at any time t, the optimal value of $p_j(t) = \{k - c(t)\}/m(t)$. Similarly, the optimal leaving probability for 'temp' nPlugs is $p_\ell(t) = \{c(t) - k\}/c_\ell(t)$, where $c_\ell(t)$ denotes the number of 'temp' nPlugs in the system. Figure 7 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 convert the persistent GSMA protocols above into those that use a varying contention window so that optimal values of p_j and p_ℓ can be chosen in an automated manner.

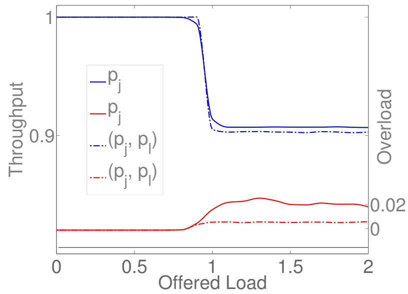

Figure 7: Asymptotic performance of p_j-persistent and (p_j, p_ℓ)-persistent schemes for demand dispatch

Differences with networking protocols The above protocols differ from CSMA protocols used by the MAC-layer to share a communication channel in the following way. 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.

5. EVALUATION

In this section, we present the experimental evaluation of the nPlug algorithms. For this evaluation, we use data from an ongoing project [4], 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 also report the line voltage and frequency every second. This time series is used by the analytics module to infer peak periods and detect supply-demand imbalance. We use three months (October 2011 to January 2012) of voltage and frequency time series data in our experiments.

5.1 Inferring peak and off-peak periods

Figure 8(top) plots seven different voltage time series corresponding to seven days, as sensed by a smart plug at a household for a week. The plot shows that (i) the line voltage varies over a wide range from 218 to 250V 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 typically switched on. The voltage decreases during the day mostly remaining within a range and decreases further in the evenings after about 6PM when people generally return from work and electricity is used for lighting and other appliances. This shows that voltage could be a good indicator of the load in the grid.

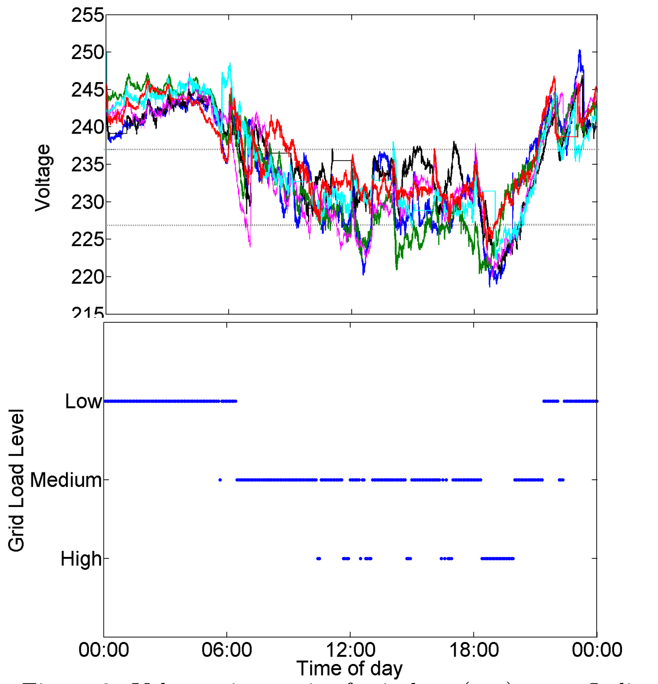

Figure 8: Voltage time series for 7 days (top) at an Indian households and the inferred grid load pattern (bottom)

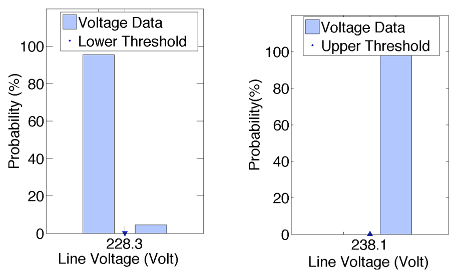

Figure 9: Histograms of voltages at 7:00 PM (left, peak) and 3:00 AM (right, off-peak)

Figure 8(bottom) plots the corresponding median grid load pattern that is inferred by the analytics module after the voltage time series is compressed using PAA (sections 4.2 and 4.3.1). 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 determine if the inferred peak period is indeed a period when the load on the grid is high, we compare the voltage-based inference with the grid load reports published by the Southern Regional Load Despatch Center (SRLDC) in India [18]. According to these reports, the evening peaks in south India occur at about 7:00PM and 7:30PM during winter and summer months respectively while 3:00AM is off-peak. Figure 9 (left) shows the histogram of voltages from 7:00-7:20PM for three months of data. We see that 95% of the time the voltage remains below $V_\ell = 228$ implying a peak period. Similarly figure 9 (right) shows the histogram for 3:00-3:20AM in the morning when the voltage always remains above $V_u = 238$ implying an off-peak period. Thus the voltage-based inference concurs with the published grid load measurements.

5.2 Inferring supply-demand imbalance

Figure 10 shows the distribution of frequency time series along with the 2-SD thresholds $f_\ell = 49.4$ and $f_u = 50$ (section 4.3.2). We see that the frequency measurements vary in a very narrow range and exhibit a Gaussian-like distribution. About 95% of the 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 frequency measurements may be used to detect supply-demand imbalance.

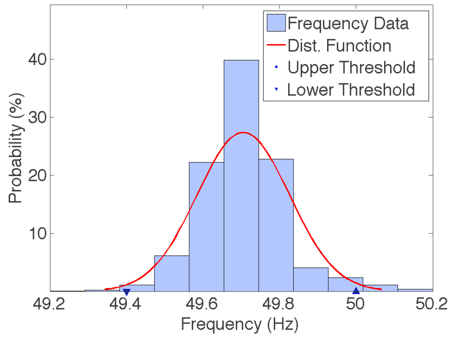

Figure 10: Distribution of frequency time series along with 2-SD thresholds

5.3 Scheduling

In this section, we present the results of Monte-Carlo simulations conducted to evaluate the performance of decentralized scheduling schemes discussed in section 4.4: (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: *maximum peak demand*, *throughput*, and *overload* (defined in section 4.4.4).

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). 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 1: (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. 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

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

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]

Table 1: Summary of experimental scenarios (the user preferences are specified as [earliest start time, latest end time, operational duration(min)])

Scenario	Maximum Peak Demand (kW)			Throughput (%)			Overload (kWh)		
No.	w/o nPlug	Random	PNLB	w/o nPlug	Random	PNLB	w/o nPlug	Random	PNLB
1	250±0	80±3.8	**72.5±1.5**	23.89±1.67	82.29±6.73	**93.24±2.07**	94.54±1.34	20.08±6.57	**6.12±1.63**
2	370±6	208±8	**190±3**	78.24±1.87	87.34±3.45	**94.14±2.39**	89.29±1.63	23.54±6.21	**9.61±3.23**
3	325±5.20	211.9±10.23	**185±6.65**	80.11±1.78	88.23±3.87	**94.28±2.28**	91.42±2.58	34.84±9.22	**8.56±3.88**

Table 2: Summary of Monte-Carlo simulations for different decentralized scheduling algorithms based on three metrics: *Maximum Peak Demand*, *Throughput*, and *Overload (kWh load above threshold generation capacity)*.

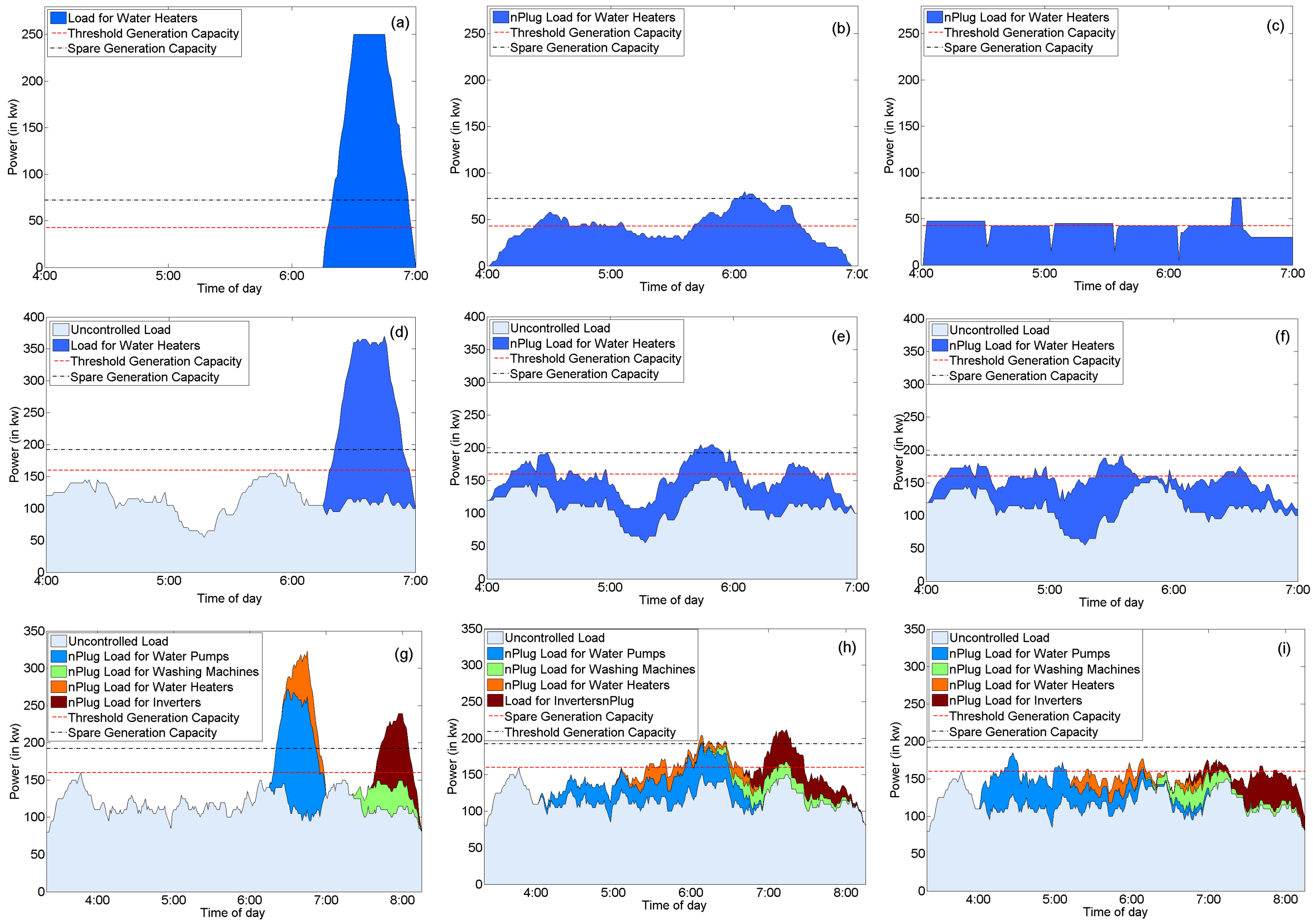

Figure 11: Performance of different scheduling schemes: rows top to bottom: Scenarios 1 to 3. Columns left to right: Base case without nPlug, Randomized scheduling, and PNLB. (a)-(c): Scheduling of of 100 water heaters using nPlug. (e)-(f): Scheduling 100 water heaters in the presence of varying background 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.

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, 16, 14]. Figure 11(a) shows such a peak when 100 water heaters are switched on between 6:15 and 7AM. In order 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 11(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 2.

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 11(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 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 11(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 1. Figure 11(h)-(i) plot the results for one sample run of randomized scheduling and PNLB. Table 2 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.

6. RELATED WORK

Demand Side Management or Demand Response(DR) [21] is essentially a mechanism for inducing the consumers to alter their consumption patterns in response to changes in supply so that available capacity may be shared efficiently. Such demand change is 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 both the centralized and decentralized demand management schemes.

One of the earliest proposed grid-friendly appliances is Frequency Adaptive, Power-energy Re-scheduler (FAPER) invented by Schweppe [17]. 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 [3] uses an approach identical to that of FAPER and has similar shortcomings. Another example is the Grid-Friendly controller from PNNL [15], 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 [1] 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 [2] 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[6] are devices that plug into home electrical pannels 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) [8] to mitigate the power imbalance due to intermittent renewable energy sources. Both DILC and Bluepod require a communication infrastructure to receive signals from control centers.

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

7. 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*, a smart plug that 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 give the high level architecture and the hardware 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.

We are considering several future extensions to our work. Firstly, we plan to collect voltage measurements from neighboring homes which are under the same phase or transformer to study how load of one household affects the voltage of other households and how soon other households can sense this. Second, we are analyzing the performance of GSMA-based schemes and plan to study congestion control techniques in the context of decentralized scheduling. Third, we are attempting to relax the assumption that the deferrable loads must be run continuously. we are investigating the mechanisms to incorporate the ability to handle loads that can be interrupted such as water heaters and air conditioners. It requires a thorough understanding of the impact on the appliance performance and consequent energy consumption. For instance, if a water heater is powered up and down frequently, the total energy consumption might increase. Moreover, if there is a large gap between two consecutive running time slots, the heating elements could cool down and might require additional energy to heat up to reach prior levels. Lastly, we plan to extend the scheduling schemes to consider appliances like electric vehicles and inverters which have varying electricity demands depending upon their existing charge and storage capacity. However, such scheduling schemes must include additional heuristics as they have to factor in the impact of battery lifetimes with varying charging/discharging cycles.

A few questions also remain unanswered. Even though the 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. Another issue is that of security. Smart plugs such as nPlugs can potentially be tampered by users and used to automate launching coordinated attacks that intentionally overload the grid.

8. REFERENCES

[1] Nest, the learning thermostat. `http://www.nest.com`.

[2] Peaksaver. `https://www.hydroottawa.com/residential/saveonenergy/programs-and-incentives/peaksaver/`.

[3] Rltec. `http://www.rltec.com`.

[4] 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*, GreenNets '11, pages 43–48, 2011.

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

[6] `http://www.voltalis.com`.

[7] Central Electricity Authority, Ministry of Power, Government of India. Load generation balance report 2011-12. `http://www.cea.nic.in/reports/yearly/lgbr_report.pdf`, 2011.

[8] 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.

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

[10] KPMG. India electricity market outlook 2008, 2008.

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

[12] 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*, DMKD '03, pages 2–11, New York, NY, USA, 2003. ACM.

[13] Ministry of Power. Energy conservation act, 2010.

[14] C. More, S. J. Saikia, and R. Banerjee. An analysis of maharashtra's power situation. `http://www.me.iitb.ac.in/~rangan/publications/recentpapers`, 2008.

[15] Pacific Northwest National Laboratory. Grid friendly appliance controller. `http://www.pnl.gov/news/release.aspx?id=856`.

[16] I. R. Pillai and R. Banerjee. Methodology for estimation of potential for solar water heating in a target area. *Solar Energy*, 81(2):162–72, 2006.

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

[18] SRLDC. Daily operation report. `http://www.srldc.org/var/ftp/reports/psp/2012/Jan12/19-01-2012-psp.pdf`.

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

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

[21] 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`.