

Its Different: Hitchhikers Tryst with Energy Consumption Patterns in India

Alice Security

Department of Computer Science
University of Southern California

alice@example.edu

Bob Privacy

Networked Embedded Systems Group
Swedish Institute of Computer Science

bob@example.se

Abstract

This paper provides a sample of a L^AT_EX document for ACM Sensys. It complements the document *Author's (Alternate) Guide to Preparing ACM SIG Proceedings Using L^AT_EX2_E and BibT_EX*. This source file has been written with the intention of being compiled under L^AT_EX2_E and BibT_EX.

To make best use of this sample document, run it through pdflatex and bibtex to directly produce a pdf document.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; D.2.8 [Software Engineering]: Metrics—complexity measures, performance measures

General Terms

Delphi theory

Keywords

ACM proceedings, L^AT_EX, text tagging

1 Introduction

Buildings (both residential and commercial) account for 30% of all energy consumption globally. The contribution of buildings to overall energy usage for different countries is as follows- Australia:20%, China:34%, India:47%, Japan:33% and USA:29%. In many of these countries residential buildings contribute significantly to overall building energy usage (China:90%, India:93%, Australia:67%).

Residential deployments can provide detailed insights about occupant behavior (specifically activities of daily living (ADLs)) and energy consumption (gas, water and electricity). These deployments also provide data sets which can be leveraged to do realistic building simulations. In the recent past several datasets, such as REDD[9], BLUED[5], Smart*[3] , monitoring household electricity and ambient parameters. These datasets were specifically aimed at Non intrusive load monitoring[7] and have helped advance the

field. Most of the previous deployments have been done in the context of developed countries. However, it is equally important to understand building energy consumption in developing countries such as India. The peak demand deficit in India was reported to be 10.3% in the year 2010-2011. Also, the demand is increasing at a fast pace due to additional construction.

We have been involved in such deployments in Indian context for more than a year whereby we have until now instrumented - more than 20 homes with smart meters, smart campus with sensors for ambient monitoring in a research wing and energy meters in the dorms. In this paper we discuss an extensive ongoing deployment in one of the homes in Delhi, India. This home is instrumented with - smart electricity and water meters, plug level load monitoring for major appliances in the home, ambient environment across every room etc. To the best of our knowledge, this is first such extensive deployment outside any developed country. We discuss in detail specific characteristics of energy consumption observed in our deployment context that are different from those observed in other deployments in the past and their impact on energy consumption related research, specifically NILM. The deployment was maintained as an open source project clearly showing the issues faced and how these were addressed. Unlike many of the past deployments, detailed metadata logs are also provided.

2 Related Work

3 Deployment Overview

Over the past year, we have performed extensive sensor deployment for ambient monitoring in research environment and smart meters across more than 25 homes in India. Building upon these experiences, we undertook a detailed deployment, constituting 33 sensors measuring electricity, water and ambient parameters across different resolution, in one of the homes in Delhi, India during June-July 2013. Primary objective for this deployment was to bring forth the differences in the Indian context, as compared to the context of developed countries along the dimensions of - 1. The ecosystem of available sensing options that restrict the possible deployments; 2. Energy and water consumption patterns; and 3. Grid and network reliability. We believe that this discussion will enrich the existing research in building energy domain, which has only leveraged deployments and data collection in the context of developed countries, for wider ap-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

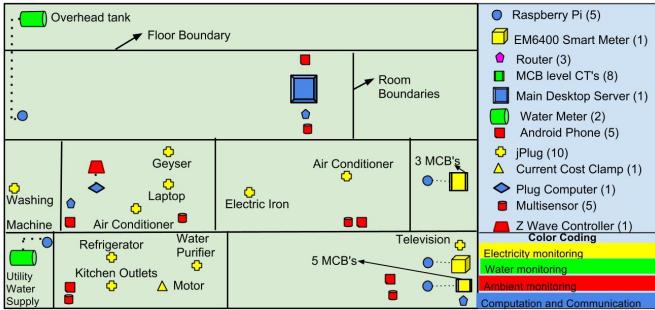


Figure 1: Schematic showing overall home deployment

plicability across different contexts. Figure 1 shows the deployment of these sensors in a 3 storey home, together with 5 single board computers and 3 routers for computing and communication. We now provide detailed description for our deployment.

3.1 Sensing Infrastructure

For sensing, we took a “leave no stone unturned” approach, where we chose to monitor as many physical parameters (ambient conditions, electricity usage and water usage) and non-physical parameters (such as network strength and network connectivity, among other). We, however, chose to deploy these sensors in a way that home users can continue their daily routines without getting affected. Many of these sensors were procured from outside India due to limited options available in the Indian context. Detailed description of the sensors used for monitoring various parameters are explained next.

Electricity monitoring: We chose to monitor electricity consumption across different resolutions - electricity meter monitoring the consumption at the home aggregate level, Current Transformers (CTs) monitoring current for Miniature Circuit Breakers (MCB’s) (each connected to a combination of appliances) and plug level monitors for consumption monitoring of plug load based appliance (see Figure 3a for illustration).

1. **Meter level:** Schneider Electric EM6400¹ smart meter is used to instrument the main power supply (see Figure 4a showing EM6400 smart meter deployed in the electricity panel). 30:5 ratio CT was used on the power mains coming from the grid to ensure that the load current is within the permissible limits (5 A) of the meter. All the necessary parameters including voltage, current, frequency, phase angle, and power (apparent, real, reactive) were read at 1 Hz using Modbus over RS485-serial link provided by EM6400. Based on our experience, we believe that the support required for large scale smart metering deployments, exists in the Indian context.

2. **Circuit level:** Split CTs, clamped to individual MCB’s, are used for monitoring circuit level current. Since no commercial solution is easily available in India for

panel level monitoring, a low cost microcontroller platform is used to collect Analog output (**sampling at**) from CTs, calculate the rms value and communicate it over serial UART interface (Figure 4b captures CTs monitoring 3 MCBs on the first floor MCB box).

3. **Appliance level:** Similar to circuit level monitors, there are no good commercial options for plug level monitors in Indian context as well. Correspondingly, we worked with our collaborators who in-house developed jPlug² to measure individual appliance power consumption. 9 of these jPlugs were used to monitor plug-load type appliances. jPlug sits in between an appliance and the socket and provides multiple parameters (including voltage, current, phase angle and frequency) at 1 Hz resolution. We also used Current Cost based CT to measure the power consumption for electric motor (used to pump water), which is not a plug-load, yet has significant power consumption (approx. 700 Watts). jPlug and Current Cost CT deployment are shown in Figure 4c and Figure 4d respectively. jPlug makes a HTTP POST connection to log measured parameters. Current Cost exposes apparent power data over the USB port.

Water monitoring: There are a few differences in home water distribution in India when compared to developed countries. Due to non availability of 24x7 water supply, overhead water tanks (typically 1000 liters capacity) are used to store water, when available, and supply it for home activities throughout the day. Due to low water pressure, even during the times when the supply is available, electric motors are used to pump the water for storage. Figure 3b illustrates the water flow distribution (**isn't there some water that is consumed directly from the utility - we have it in our home**), together with placement of water meters. One water meter is placed at the inlet (coming from the utility) and another one at the outlet from the water tank (flowing downwards) to capture the incoming (from utility) and outgoing (from tank) water consumption.

Due to prohibitive cost and difficulty in procurement of digital water meters in India, we chose to use Zenner Aquameter’s multijet³, a pulse output based water meters. Output pulse is generated through a 4-20 mA current loop, commonly used for analog signaling in industrial process control instruments. For the water meter connected to the utility, over a 0.5 inch diameter pipe, a pulse is generated for every 1 liter of water consumption. Water meter connected to the outlet of stoarge tank, with 1.25 inch diameter, generates a pulse every 10 liters of water consumption. Figure 4e shows the water meter deployed inline at the overhead tank.

Ambient conditions monitoring: We used ZWave based HomeSeer HSM100⁴ multisensors for monitoring motion, light and temperature across 5 rooms in the home. Till date, to the best of our knowledge, no commercial ZWave based sensor is available working on Indian frequency (865.2 MHz). We correspondingly imported EU frequency (868.4 MHz) devices and used it for ambient monitoring. For these

²A variant of nPlug[6]

³www.aquametwatermeters.com/multijet.html

⁴www.homeseer.com/pdfs/guides/HSM100_Release_Notes.pdf

HSM100, motion is reported in an event-driven manner (i.e. whenever there is change in motion status, a reading is reported) and temperature and light are polled at 1 Hz. We also placed 1 Android phone at a fixed location in each room and ran FunF journal application⁵ to log ambient parameters such as light and sound level every 30 seconds for 2 seconds (**Confirm the frequency - I am also not sure how you calculated the frequency mentioned in the table**).

Miscellaneous: Android phones, in addition to measuring ambient conditions, were also used to scan and log Bluetooth, WiFi and GSM networks. All home occupants were requested to keep the Bluetooth on during the duration of the experiment. The network scanning was done every 1 minute and is stored locally on the SD card. **Confirm the rate of data collection on the phone.** External weather conditions, such as temperature, humidity and wind speed, were also logged using publicly available API's from weather monitoring stations **Give citation to the API used and the rate of data collection.** Additionally, network traffic was sensed to monitor connectivity and signal strength. Such miscellaneous data collection, together with detailed electrical consumption, water consumption and ambient information, can be used for multiple applications including energy-apportionment (i.e. assigning energy usage to different occupants within the home), localization and correlating energy usage with outside temperature.

Complete sensing infrastructure is summarized in Table 1. **One of the parameters you can mention is power supply - battery/direct plug/not required.**

3.2 Communication and Computation Infrastructure

Multitude of sensing infrastructure, as described in Section ??, involves multitude of communication channels as well. Different computing platforms - microcontrollers, Single Board Computers (SBC's) (e.g. Raspberry Pi⁶ (RPi) and Ionics Stratus⁷) and desktops are used for data collection. SBC's served as a good low cost alternative (25 \$) constituting 700 MHz ARM processors, support for Linux based distributions and allow for coding in higher level languages such as Python.

We used 5 RPi and 1 Ionics Stratus plug computer in our deployment. We used a 2 GHz Desktop PC running Linux, as the main server where all the data was stored. One RPi was connected to EM6400 using RS485-USB converter. We developed a custom Python program based on pyModbus⁸ to collect the desired parameters. All the collected parameters, together with the UTC timestamp is stored in a local CSV file. A new CSV was created every 15 minutes. A background program would periodically try to upload CSV files older than 15 minutes, to the main server, where the data is pushed in MySQL database. This architecture for data collection - creating a new CSV periodically, uploading older CSV periodically to main server where the data is then pushed into MySQL, is used for all of our sensors connected

to SBCs. We call this model Sense-Local Store-Upload and describe it in more details in section Section 3.3. On similar lines, USB output from Current Cost (providing data in xml format) is received on another RPi and goes through the Sense-Local Store-Upload architecture for data collection. Although Current Cost has its own cloud based API's, we chose to collect data from it locally, in order to avoid data losses occurring due to network failures.

Two RPi were used to connect to the microcontroller based circuit monitor to read the data over the serial port and process it further using Sense-Local Store-Upload. One of the RPi connected to the meter was used for circuit monitoring as well for the CT's used in the same panel. GPIO headers on two more RPis were used to interface with each water meter respectively for collecting pulse output over the 4-20 mA current loop interface. We initially wrote an interrupt driven program in Python to detect 10 liter and 1 liter events for the tank and supply water meters respectively. We observed that noise introduced in the circuit due to long cable lengths led to a lot of false events. Correspondingly, we modified our program and polled at a frequency of 5 Hz to obtain GPIO status and further processed this data using Sense-Local Store-Upload.

Data from jPlugs (using the HTTP post request every second) is directly collected on the server machine where a web daemon listened to requests from jPlug and dumped them in MySQL. Ionics Plug Computer is used to collect data from all ZWave based sensors. We wrote custom wrappers around OpenZWave⁹ to collect temperature and light information on per second basis, and motion information based on events. A ZWave controller was connected over USB with Ionics and the data was processed further using Sense-Local Store-Upload. Figure 4g shows the plug computer collecting ambient sensor data from ZWave controller. A manual dump of locally collected data on each Android phone was performed every 15 days. Electricity failure and unreliable internet are well known problems in India and are highlighted in Section 4. Owing to these problems, we chose to collect weather data from 3 different weather stations, in one of the servers hosted by our collaborators in the USA.

A lot of issues, pertaining to SBC's, were observed in our deployment. As an example, the OpenZWave program that we used created log files for its own diagnostics, that eventually consumed the 512 MB flash drive space on the plug computer. This was fixed eventually by deleting older logs. Such problems encouraged us to develop soft-sensor streams, whereby we collected hard disk space, ping success, CPU utilized, remaining RAM space and temperature of the processor, for all the computing devices including the server at regular intervals. These soft-sensor streams can be used for offline analysis as well as for real time alerting and fault diagnosis. Similar soft-sensor streams have also been used in the previous work[8] for raising alerts.

Similar to prior research, reporting WiFi discontinuity in homes in the USA [8], we also observed that one WiFi router will not suffice for our 3 storey home. We thus used 3 Net-

⁵<http://www.funf.org/journal.html>

⁶www.raspberrypi.org

⁷www.ionics-ems.com/plugtop/stratus.html

⁸www.github.com/bashwork/pymodbus

⁹www.code.google.com/p/open-zwave

Table 1: Sensing Infrastructure

Sensor name	Sensor type	Sampling frequency (Hz)	Resolution	Quantity	Communication	Observed parameters
EM6400	Electric Meter	1	Home	1	RS 485 Serial	Voltage, Current, Frequency, Phase, Power(Active, Reactive and Apparent), Energy
Aquamet multijet	Water Meter	5	Main supply and tank	2	Direct wire connection to GPIO	10 liter events for tank and 1 liter events for main supply
Homeseer HSM100	Ambient multisensor	Light, temperature: 1 PIR: event based	Room	6	ZWave	Light, temperature and motion
Android phones	Ambient multisensor and network	Audio, light: 0.05 Nearby WiFi, cellular, bluetooth: 0.001	Room	5	Manually transfer files to PC	Audio features, light, nearby bluetooth, cell-tower, WiFi
Prototype CT	Electricity meter	20	MCB	8	Serial	Current
jPlug	Electricity meter	1	Appliance	10	WiFi	Voltage, Current, Frequency, Power (Active and Apparent), Energy, Phase
Current Cost	Electricity meter	0.1	Appliance	1	Serial	Apparent power

gear JNR1010¹⁰ routers, where the router on the first floor acted as the host and the routers on the ground and the second floor were bridged to it. More details regarding the need of multiple routers and network availability in homes is described in Section 4 and Section 5.

3.3 System Architecture

Various middleware systems such as sMAP[4], Building Depot[1] and SensorAct[2] have been proposed in the past for sensor data collection. However, we found that they are not fine tuned to our needs arising from faulty internet, power failures, etc. Moreover, these systems provide a lot more features than we intend to use. Thus, based on our experience and previous work[8], where importance of simplifying the architecture are proposed, we propose Sense-Local Store-Upload. Sense-Local Store-Upload involves two main ideas-local storage (on SBC’s) and periodic data upload (from SBC to server, and from server to cloud). As explained above, we used SBC’s to collect data from sensors. This data was **locally stored** in form of comma separated value files (CSV) and was **periodically uploaded** to the main desktop server. In case the upload failed, it was retried the next time again. Each SBC had sufficient flash based local storage to accommodate sensor data for a few days.

Web applications running on the server allowed a home user to locally visualize her data from multiple streams. Further, data from the server can be copied to cloud based servers such as Dropbox where applications such as analytics, alerts can be done, allowing outside users and specifically researchers to keep an eye on the deployment sitting in their labs. Figure 2 explains Sense-Local Store-Upload architecture. The salient features of our architecture are as follows:

- **Decoupled sensing and data uploading:** This ensures that an error in data uploading does not have an impact on sensing and vice versa. Thus, there is no data

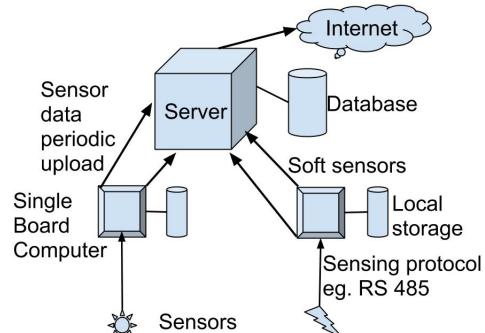


Figure 2: Sense-Local Store-Upload

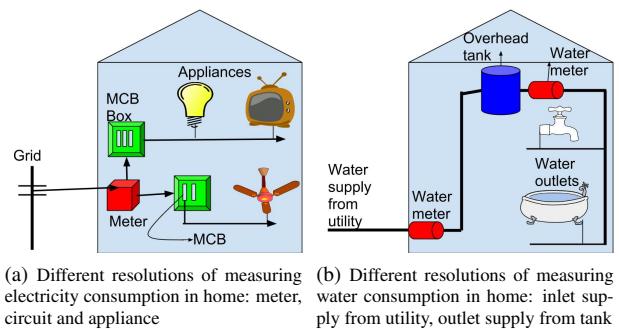


Figure 3: Electricity and water flow inside a home and different resolutions at which these parameters can be monitored

¹⁰www.support.netgear.com/product/JNR1010

loss owing to internet failure. The design choice of decoupling comes from the well established principles of software engineering.

- **Minimal internet requirement:** Internet is required **only** when outside researchers wish to view and analyze collected data in realtime. Internet failure does not have any impact on sensor data collection. Also the periodic nature of our uploads ensures that whenever internet is back, data would be uploaded.
- **Reduced load on server:** Since data is uploaded periodically in larger chunks rather than sending data for each sensor packet, computation and bandwidth requirements on SBC's and server are reduced. Though this comes at the cost of increased storage requirements on SBC's, we found that for all practical purposes, we did not need to provision for more than 1 GB.

4 How is this deployment different?

In this section we discuss some of the key unique aspects brought forward from our deployment, which are as follows:

Unreliable grid is commonplace: Load shedding or rolling blackout are common in developing countries, owing to several reasons such as improper infrastructure management, high demand and low electricity production. Power cuts are more common in summers when load is higher due to usage of air conditioners. As a result, voltage fluctuations or brownouts are also common.

We collected various statistics from our deployment which illustrate these aspects. Figure 5b shows power cut in number of hours per day during a two month period in summer of 2013. Electricity outage for upto 12 hours has been reported on some days. Figure 5b shows the duration in minutes of power outages. 107 power outages were found in 61 day period from May to July. While the average power outage lasted about 1 hours, there were outages lasting upto 9 hours. Figure 5c shows the power outage distribution by hour of the day. It can be seen that there are maximum outages around 10 AM in the morning and around midnight. These times correspond to office time and night time when air conditioners in home are on. We believe that this indicates heavy load on the grid. Figure 5d shows the voltage fluctuations in a typical day, where it can be seen that voltage is fluctuating from 180 to 250 Volts. We can observe that voltage drops during peak load hours. Figure 5f shows that voltage just before power outage in night hours (10 PM to 1 AM), when load is expected to be maximum. We can observe that voltage is well below rated voltage of 230 V. This is in coherence with previous work[6], which hypothesized that frequency and voltage indicate load on the grid. Figure 5e shows fluctuation in frequency in a typical day. Due to unreliable nature of grid, we ensured that all our systems were capable of automatically starting after a power outage. We put our scripts for data collection and upload in Linux startup script. This provided us another advantage. If we observed that a system was down, we could just ask the home occupant to repower the system again. This ensured that there was minimal data loss till the time our researchers could visit the site and repair the fault.

Unreliable internet connectivity: While India has one of

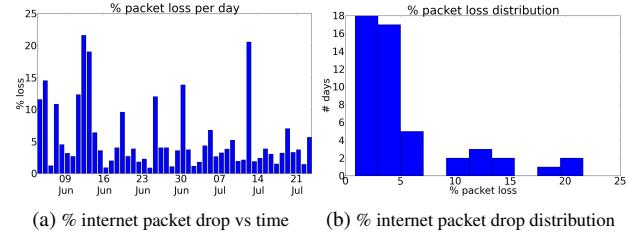


Figure 6: Unreliable internet

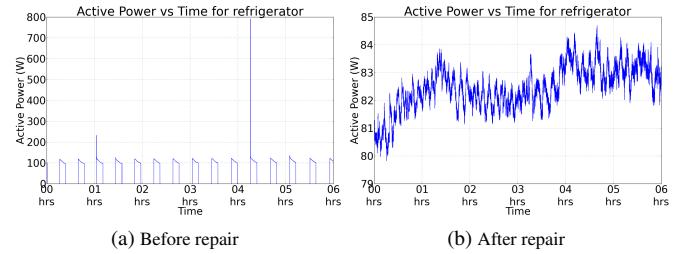


Figure 7: Refrigerator power consumption

the fastest growing internet population, only 11% of the total population is connected to internet, the corresponding figure in US is 78%[10]. Average connection speed in India is 1.3 Mbps, the corresponding numbers in developed countries are in excess of 10 Mbps[11]. Even though the condition of internet has vastly improved in the previous few years, from practical experience, we have found it to unreliable. We found internet would be unavailable several times during the day and sometimes it would be too slow. This was a prime motivation behind local storage and periodic data upload, which we have proposed in Sense-Local Store-Upload. We found that eventually all the data gets synced up to the main server and from there to the cloud over the internet. We thus believe that for deployments where internet is unreliable, architectures such as Sense-Local Store-Upload should be adopted.

To illustrate the need for Sense-Local Store-Upload , we collected network statistics from our deployments. We sent a packet consisting of 15 internet ping requests every 15 seconds and reported packet drop. Figure 6a shows that packet drop of upto 25% was observed on certain days. The average packet drop was around 6% per day. Figure 6b shows a histogram plot of % packet drop. It can be seen that on 1/6 fraction of total days, % packet loss is greater than 10%.

Importance of meta data collection: We collected metadata associated with electrical appliances throughout our deployment. We collected parameters such as appliance name, age, mode of usage (for instance air conditioner set temperature). We believe this detailed metadata can enhance NILM and can provide useful insights for conserving electricity. For instance, in one of the home refrigerator was sent for repairing on 2nd July. Figure 7a and Figure 7b show the active power consumption before and after repair. We found that after repair the refrigerator was configured in coolest mode

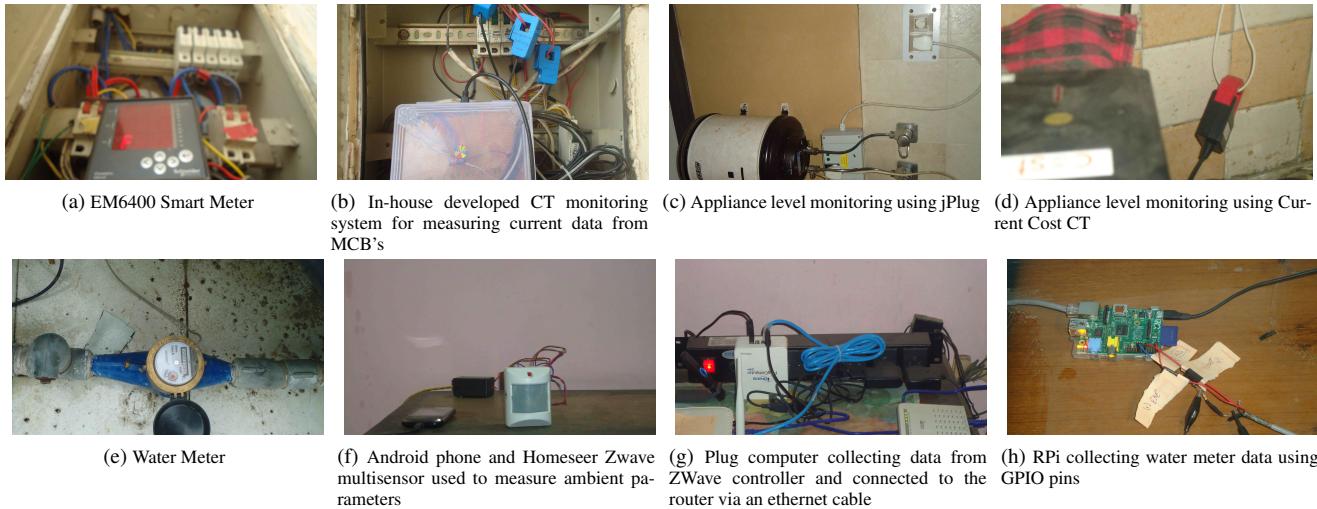


Figure 4: Sensing, computation and communication equipment used for deployment

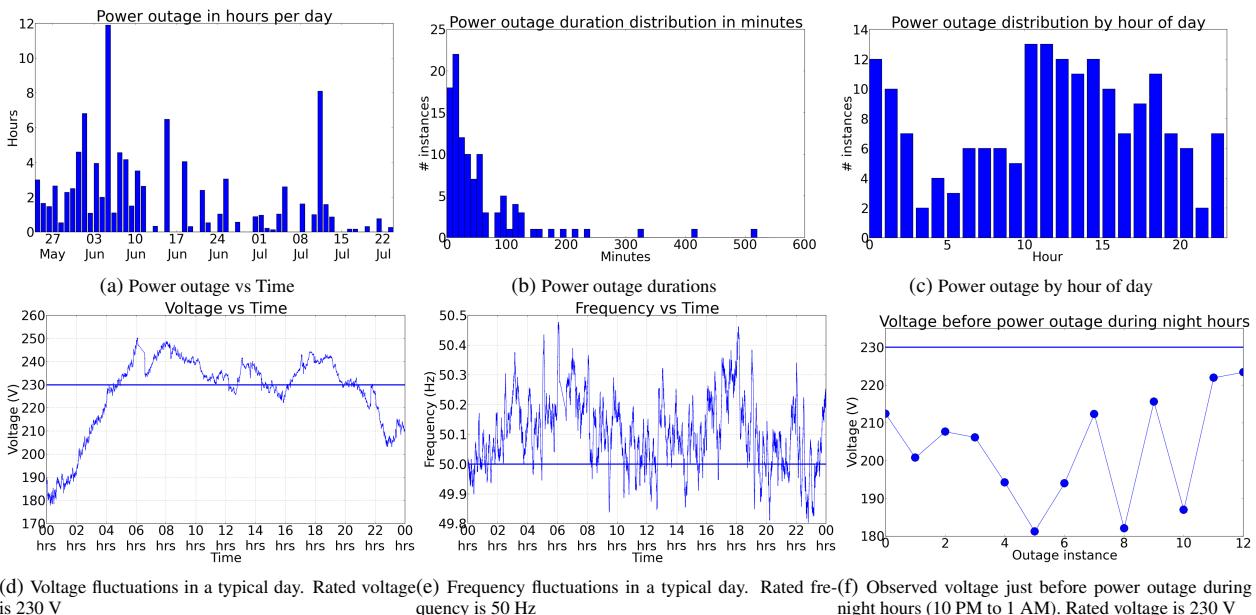


Figure 5: Unreliable grid

(lowest temperature), while before repairing, it was configured in least coolest mode (highest temperature). After repair the refrigerator was found to be consuming 1KWh more per day (which is 140% above normal). Moreover, its duty cycle pattern was modified significantly. This metadata can be useful to NILM approaches which model appliance usage duration as one of their parameters.

Load specifics: Electrical load setting in India are different from that in the US and Europe. There are no automated loads and HVAC systems are decentralized. Thus, air conditioning is at room level and water heating is done using geysers at bathroom level. From our deployments, we observed that air conditioners and geysers account for upto 70% and 50% of overall home electricity in summers and winters respectively. Thus, any small improvement in efficiency of these two appliance can significantly lower the home electricity consumption. [Add pie charts and show ref + geyser/AC is most significant](#)

Water pumping and filtering are two activities whose scope varies across both water and electricity dimensions. Water motor, described in section Section 3.1, is used to pump water upto water tank on roof. It was observed that without motor it took 8 seconds to fill 1 liter of water into the tank and after switching on the motor it took 4 seconds. The power consumption of the electric motor is around 700 W. If the water tank is empty, it would take 8000 seconds(2 hours 12 mins) to completely fill it up if the motor is not used. If however, motor is used it would take 4000 seconds (1 hours 6 mins) to completely fill the tank. An additional 770 Wh energy would be consumed by the motor. Thus, there is a fine balance between electricity and consumption and time taken to fill the tank. Since water flow from utilities is provided only for about 1 hour each in the morning and evening, motor is used for about 10 minutes a day.

Reverse Osmosis (RO) based water filters are commonly used to filter the drinking water. We found that to filter 1 liter of water, the water filter takes about 1 minute, during which it consumes 40 W. Thus, if 30 liters of water is filtered in a day, 20 Wh energy would be consumed by the water filter.

Another interesting distinction in the loads in India and that in US and Europe is that our loads are not always on and there is a switch with each plug socket. We found that the jPlugs attached to kitchen appliances such as microwave and mixer, which when used for less than 1 minute, did not report power readings. This was due to the fact that jPlug takes some time to reach steady state and before it would do so, the appliance would be turned off. If the appliances were to be always on, this data would not have been lost. From our survey with the home occupants, we found that their typical microwave and mixer usage was less than 2 minutes. We seek to resolve this issue in our future deployments.

Lack of quality measurement devices commonly found in US and Europe: ZWave sensors for Indian frequency (865.2 MHz) are not readily available in the market. Thus, we imported European frequency (868.4 MHz) based ZWave multisensors and plug load monitors. Import duty and shipment delays make them an expensive choice. As explained above, our loads are not always on and are controlled via

plug sockets. Due to this ZWave based plug load monitors could not be used in our deployments. A switch on the ZWave based plug monitor had to be pressed after each power outage and as explained before power outages are common, we chose not to use these plug monitors. Further other alternatives like Current Cost based CT's involve exposing live wire from appliances, an option which was rejected by home occupants. Also, Current Cost based Individual Appliance Monitors (IAM) provide only apparent power and energy at low frequency. Thus, we had to resort to jPlug, which is a prototype solution.

5 Hitchiker's guide revisited

In this section, we discuss similarities between our deployment and those explained in previous work, most specifically- "The Hitchhikers Guide to Successful Residential Sensing Deployments"[8]. Some prominent similarities are:

Homes are hazardous environments: We observed that one of our multisensors would fail after power outage. Initially, we felt that the sensor was faulty, but other multisensors also showed the same behavior in the same location. We figured that this behavior was due to the fact that this multisensor was put on inverter point (battery backup) and would not go down even on power outage. Thus, when power resumed, ZWave controller was not able to add this multisensor to its network. We resolved this by putting the multisensor at a non-inverter point. We also observed that one of the multisensor would always indicate motion. Initially we solved this problem by replacing its power supply. However, within a week, this problem resurfaced. We believe it is due to faulty power supply in that room. The home occupants did not allow a thorough investigation of that electrical socket and we had to do away with that data.

Although we used zip-ties extensively throughout the deployment to prevent hanging wires, we observed data loss in one of our multisensor and Android phone, which went out of power due to wire snag. This is shown in Figure 9b. Further, even after 1 month rigorous testing in lab, we found that 60 new issues (service complaints) were registered when we moved the deployment to the first home. This is only a testimony that homes present a unique scenario which can not be emulated completely in laboratory settings.

Aesthetics matter: As stated in previous work, sensor LED's can be bothersome to home occupants, particularly in the night. Our 33 sensor deployment had introduced 63 LED's in the home. Figure 9a shows our sensor LED's blinking in the night. For our current deployment, choosing appropriate sensor location sufficed. However, for future such deployments, we intend to build better casing to ensure that home occupants are not disturbed. Also, the home occupants complained of buzz like sound coming from the desktop server that we had put. We figured that this noise was due to the fact that its CPU was clogged due to dust. This is another unique aspect introduced in the Indian setting. Thus, we believe that any long lived deployment should have routine maintenance.

Homes are not designed for sensing: We observed that the data collected from our ground floor MCB's (5 in number)

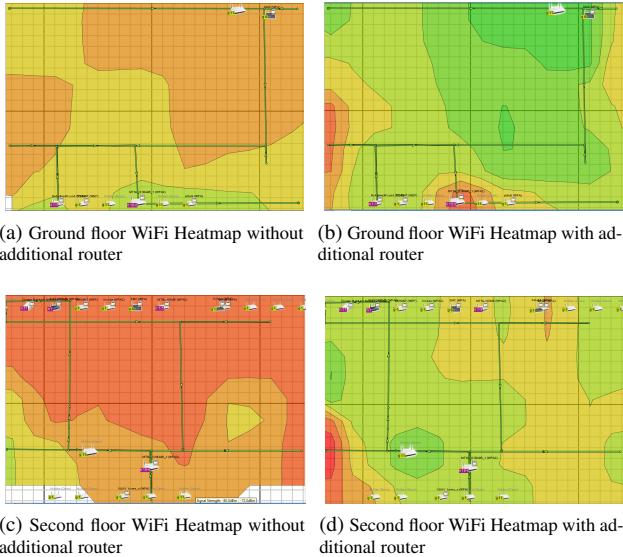


Figure 8: WiFi signal without placing additional routers on different floors is poor. The greener the region, the better the signal strength for the heatmap.



Figure 9: Common problems in residential deployments

was noisy. This was due to the fact that the MCB's are very close (shown in Figure 9c) to each other and this causes interference in our CT monitoring circuit. However, on the 3 MCB's on the first floor, there was adequate gap amongst MCB's and data was of good quality. We thus believe that homes are not designed for sensing and one should be prepared to lose some data quality due to that.

Redundancy-Accounting for sensor failure: During our deployment 3 jPlugs and 1 multisensor stopped functioning properly. Fortunately, we had accounted for such failure and had additional sensors ready. We believe that one should beforehand account for sensor failure in residential deployments.

Homes have poor connectivity: During the preliminary phase of our deployment, we tried to work with existing networking infrastructure in the homes. The WiFi router was placed on the first floor and signal strength on ground and second floor was poor. Figure 8a and Figure 8c show

the WiFi heatmap produced wrt the router placed on first floor. We observed that a lot of regions inside the home show poor signal strength. We bridged an additional router on the ground and the second floor with the existing first floor router. Figure 8b and Figure 8d show the corresponding WiFi heatmaps produced wrt newly introduced bridged routers. We can observe the improvement in signal strength with the introduction of additional hardware.

Another important facet of deployments is that additional sensors introduced inside a home may choke up the network bandwidth. We observed this when we tried using imported WiFi based microcontrollers for sensing ambient conditions. Occupants complained of low network bandwidth available for their personal usage and thus, we decided to use ZWave based multisensors for measuring ambient parameters. Since ZWave based sensors worked on a different frequency when compared to WiFi, they did not cause interference. Thus, in our experience, a careful mix of sensors working on different frequencies should be used to avoid interference issues.

6 Conclusions and Future Work

7 References

- [1] Y. Agarwal, R. Gupta, D. Komaki, and T. Weng. Buildingdepot: an extensible and distributed architecture for building data storage, access and sharing. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 64–71. ACM, 2012.
- [2] P. Arjunan, N. Batra, H. Choi, A. Singh, P. Singh, and M. B. Srivastava. SensorAct: A Privacy and Security Aware Federated Middleware for Building Management. In *Fourth ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, BuildSys 2012, 2012.
- [3] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht. Smart*: An open data set and tools for enabling research in sustainable homes. In *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011.
- [4] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. smap: a simple measurement and actuation profile for physical information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 197–210. ACM, 2010.
- [5] A. Filip. Blued: A fully labeled public dataset for event-based non-intrusive load monitoring research.
- [6] T. Ganu, D. P. Seetharam, V. Arya, R. Kunnath, J. Hazra, S. A. Hussain, L. C. De Silva, and S. Kalyanaraman. nplug: a smart plug for alleviating peak loads. In *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*, page 30. ACM, 2012.
- [7] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.
- [8] T. W. Hnat, V. Srinivasan, J. Lu, T. I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse. The hitchhiker’s guide to successful residential sensing deployments. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 232–245. ACM, 2011.
- [9] J. Z. Kolter and M. J. Johnson. Redd: A public data set for energy disaggregation research. In *proceedings of the SustKDD workshop on Data Mining Applications in Sustainability*, pages 1–6, 2011.
- [10] M. Meeker. Internet trends at stanford bases. KPCB, 2012.
- [11] A. Solutions. The state of the internet. Akamai Solutions, 2013.