Smart Energy Case Study

Christof Baumann
ETH Zurich, Switzerland
christof@boumann.ch

Stephan Holzer
ETH Zurich, Switzerland
stholzer@ethz.ch

Miguel Rodriguez
AIZO, Zurich, Switzerland
miguel.rodriguez@aizo.com

Roger Wattenhofer
ETH Zurich, Switzerland
wattenhofer@ethz.ch

Abstract

As the world moves towards renewable energy production, the importance of involving energy-consumers into the process of scheduling energy increases. In this paper we investigate which devices should be used to balance energy consumption to match its availability. We classify electrical devices in categories that can either start consuming energy when available or can be stopped when there are low levels of energy available. We built a prototype system that enables any household to use its electrical appliances to load balance its consumption using information on how a user has configured its appliance available to a central energy distribution system. We simulated two algorithms that could be used at an energy provider to make use of the information on how users have configured their appliances for load balancing. We present our results and a recommendation on the algorithm that can be used for load balancing on the consumer side.

Categories and Subject Descriptors

H.4 [**Information Systems Applications:**]: Process control systems

General Terms

Smart energy-aware devices and appliances, Demand-response technologies for smart-meter/smart-grid integration

Keywords

Smart Grids, Load Balancing, Heuristics, Consumption Control

1 Introduction

Balancing energy consumption to match what is generated from renewable energies like wind or solar needs a way to regulate the consumer side of the network. In some european countries the ripple control system is used [6]. A ripple control sender sends signals over the power line which are received by the households and may trigger the start or stop of devices. This communication is only unidirectional therefore the energy provider has no feedback on the number of devices that actually turn on or off. Energy providers have to predict future power consumption [8, 9] and must keep the power grid running at all times. Failing to cover consumption needs might result in power outages. Unfortunatelly currently the only options are to start up or shut down power plants, use ripple control and regulate the consumer side by adjusting energy-prices during peak-times.

Another approach documented in studies such as [4] show that in demand-response systems [1, 2, 7] the total load and peak-usage can be reduced significantly by performing local load balancing. In this paper we go one step further and implement a system in which energy provider and devices can communicate bidirectionally such that individual devices can be scheduled. The system can directly communicate with the customer's devices and switch them on/off at the required times. This regulation must not reduce the comfort of consumers.

2 Implementation

2.1 Home Automation System

The main advantage of the automation system we used is that it communicates over the existing power lines. Some of the details about the power line communication and the devices we used in our setup are documented in [3]. The system is suited to be deployed in existing buildings: no additional wiring is needed. This also means that the system can be deployed in existing buildings without major investments in hardware. The user only needs to install a module on each end point/device that should be controlled by the system. A typical installation consists of the following components:

- Several **meters** are deployed in the building's electrical cabinet or fuse box. One meter is needed for every circuit in the building. It communicates with the modules in the building via power line and with other meters over an RS485 bus
- The **filter** is used to condition the current and thus reduce interference with other devices. It conditions the signals on the power line and does corrections on the 50Hz sine wave.
- The **server** is connected to the same RS485 bus as all the meters. It is an embedded linux server with net-

Buildsys'12, November 6, 2012, Toronto, ON, Canada.

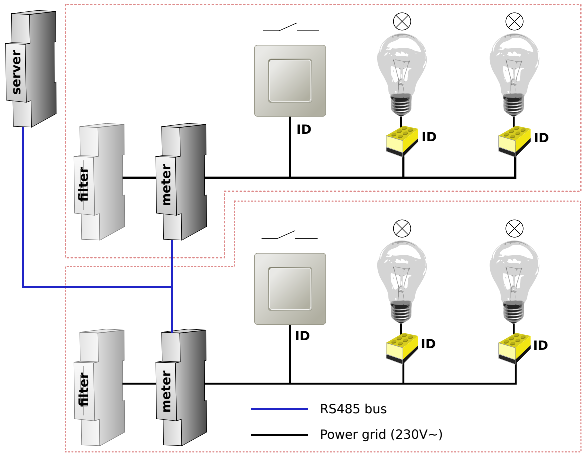


Figure 1: Overview of a sample installation

working interfaces to connect to a local network and access the internet. The server is used to enrich the functionality of the system.

- **Modules** are deployed wherever used in the house. Only the electrical devices that are used with the system have to be equipped with a module. Every module in the system has its unique ID, which is used to identify and address it. There are module adaptors that can be used to control large devices such as freezers, heaters or electrical vehicles.

An overview of a sample installation is sketched in Figure 1.

During our work we implement an extension application for the server of the home automation system. The server provides a JSON API that is accessible over a secure HTTP connection. In this extension the user specifies devices to which the energy provider has access. The energy providers can then remotely start up or shut down the configured devices according to the rules that the user defined. The compliance with the rules is enforced by the local installation. Each device in the installation can be configured to be in one of the following three states according to the smart-grid algorithm:

- Excluded from the algorithm.
- Delayed ON: a device in this state is usually switched OFF. If it is switched ON by the user, its start may be delayed by the energy provider. Examples of these devices are: Boilers, washing machines and dryers. The user sets for these devices a start time and a slot length.
- Short period OFF: a device in this state is usually switched ON all the time. It can be powered OFF by the energy provider for a short time to reduce consumption during peak times. Examples of these devices are heat pumps, freezers and air conditioners. The user defines the slot length and a maximal off time in this slot.

2.2 Demonstration: Energy Provider

We created an energy provider server simulation implemented in Java. This server can send either an energy overrun event or an underrun event to the installation. The protocol works like this:

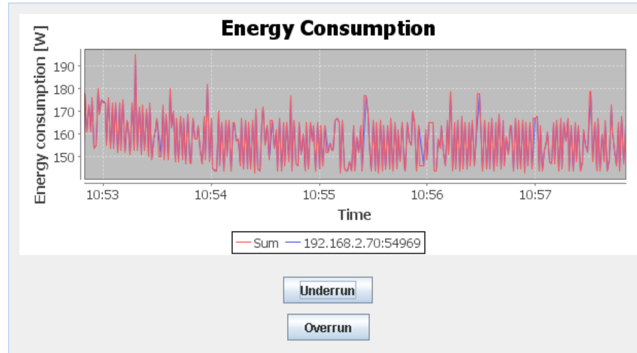


Figure 2: Screen shot of the energy provider demo application.

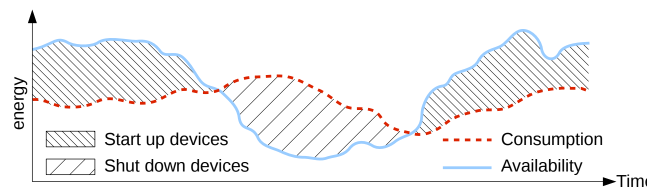


Figure 3: Illustration of the idea of the simple algorithm. Just the availabilities and the consumptions are compared.

1. The building sends its configuration values for all devices, that are configured short OFF or delay ON, to the energy provider. If information about the consumption pattern of the devices is available it is also sent.
2. The energy provider computes commands that will be sent to the building to trigger its configured devices.

We deployed our algorithm in a demonstration apartment using a Segway and a Freezer as test devices. The Segway had to be charged 20 hours after being plugged in while the freezer could be turned off 10 minutes within each hour.

3 Algorithms

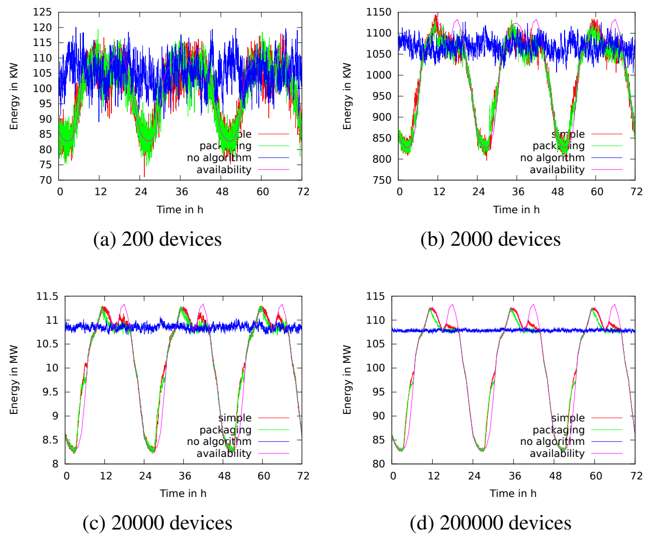
We tested 2 algorithms with our system:

1. The simple algorithm just reacts on under / over runs as seen in figure 3
2. A 2D bin packing algorithm as described in [10] and [11]. On this algorithm the "delay ON" devices are scheduled at the first possible time in its starting interval where enough energy is available as seen in Figure 4.

4 Evaluation, Simulations and Results

4.1 Simulating the Algorithms

To simulate the 2D packing algorithm predictions of the future availability and consumption are needed. The availability function was taken from the average energy production of Germany and Austria published by www.eex.com for the time between November 21, 2011 and December 12, 2011. The prediction of the consumption was computed by determining the average consumption if no algorithm is influencing the system. We then added the consumption of



(a) 200 devices (b) 2000 devices

(c) 20000 devices (d) 200000 devices

Figure 5: Results of simulation for four days

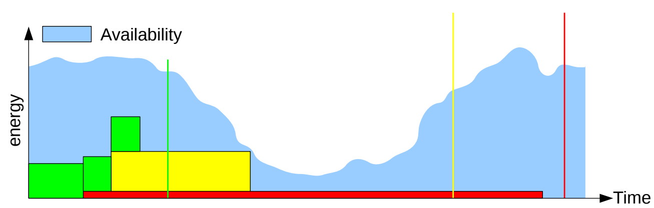


Figure 4: The devices have to be scheduled to be finished before their deadlines (vertical lines). The rectangles represent the energy consumption period of a device.

already scheduled delay ON devices.

We summarize the results of a simulattion of the algorithms for 200, 2,000, 20,000 and 200,000 devices. These numbers represent villages/towns of different (small) sizes. About $^1/_3$ of the devices were excluded from the algorithm to take e.g. plug-loads into account that cannot be controlled by our algorithm. E.g. [12] studies the effect of plug-loads to demand response systems and suggests ways how to deal with these. In some sense we capture the setting of controllable plug-loads indirectly by choosing $^1/_3$ of the devices to be in the delay ON category and the last $^1/_3$ to be short OFF devices. Of course not all these devices are plug-loads. Note that there is no statistical reason why we should choose $^1/_3$ of the devices. In a real environment the percentage of configured devices may be smaller. We chose these rather high values to have clearly visible effects on the consumption characteristics in the simulations. Both algorithms can not fulfill all the peak situations. But this is what can be expected: there is a time when all devices are turned ON or OFF and there are no more possibilities to do further corrections on the consumptions. The first peak can be served but then for the second peak no more devices are left that could be powered on in both algorithms. As one can see the simple algorithm performed better with many devices than the more sophisticated 2D-packing algorithm. This is mainly because of the fact that with many devices the expectations match reality closer than with few devices. On the contrary, the packaging algorithm performs slightly better with fewer devices. This is because with few devices the scheduling of the delay ON devices is much more important.

4.2 Future work

We did not have the possibility to deploy the system in a real village or town, however we believe it is possible to compare different algorithms in a quantitative way with this simulator. There is a follow up project that will use several existing installations to prove some of the theories that we present in this paper in the real world. This will include not only the algorithms used to control energy consumption but also the human acceptance factors of this remote controlling of electrical appliances. This project is being done together with a large electricity provider.

A further possible extension to this work would be to use the bidirectional communication channel to provide usage-patterns [5] of individual consumers to the energy-provider and consider this in the scheduling process.

5 References

[1] M. Albadi and E. El-Saadany. Demand response in electricity markets: An overview. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1 –5, june 2007.

[2] S. Borenstein, M. Jaske, and A. Rosenfeld. Dynamic pricing, advanced metering, and demand response in electricity markets. 2002.

[3] G. Dickmann. Digitalstrom®: A centralized plc topology for home automation and energy management. In *Power Line Communications and Its Applications (ISPLC), 2011 IEEE International Symposium on*, pages 352–357. IEEE, 2011.

[4] S. Kiliccote, M. Piette, J. Mathieu, and K. Parrish. Findings from seven years of field performance data for automated demand response in commercial buildings. Technical report, Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), 2010.

[5] Y. Kim, T. Schmid, M. Srivastava, and Y. Wang. Challenges in resource monitoring for residential spaces. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 1–6. ACM, 2009.

[6] E. Paessler. *Rundsteuertechnik*. Publicis MCD, 1994.

[7] F. Rahimi and A. Ipakchi. Demand response as a market resource under the smart grid paradigm. *Smart Grid, IEEE Transactions on*, 1(1):82 –88, june 2010.

[8] S. Rahman and O. Hazim. A generalized knowledge-based short-term load-forecasting technique. *Power Systems, IEEE Transactions on*, 8(2):508 –514, may 1993.

[9] N. Tang and D. Zhang. Application of a load forecasting model based on improved grey neural network in the smart grid. *Energy Procedia*, 12:180–184, 2011.

[10] L. Wei, A. Lim, and W. Zhu. A skyline-based heuristic for the 2d rectangular strip packing problem. In *IEA/AIE (2)*, volume 6704 of *Lecture Notes in Computer Science*, pages 286–295. Springer, 2011.

[11] L. Wei, D. Zhang, and Q. Chen. A least wasted first heuristic algorithm for the rectangular packing problem. *Comput. Oper. Res.*, 36:1608–1614, May 2009.

[12] T. Weng, B. Balaji, S. Dutta, R. Gupta, and Y. Agarwal. Managing plug-loads for demand response within buildings. In *BuildSys'11*, 2011.