Development of a Software Platform with Distributed Learning Algorithms for Building Energy Efficiency and Demand Response Applications

Avijit Saha

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Saifur Rahman

Manisa Pipattanasomporn

Jaime De La Ree Lopez

Alireza Haghighat

Guoqiang Yu

December 9, 2016
Arlington, Virginia

Keywords: Building Energy Management System, Energy Efficiency, Demand Response, Internet of Things, Reinforcement Learning

Development of a Software Platform with Distributed Learning Algorithms for Building Energy Efficiency and Demand Response Applications

Avijit Saha

Abstract

In the United States, over 40% of the country's total energy consumption is in buildings, most of which are either small-sized (<5,000 sqft) or medium-sized (5,000-50,000 sqft). These buildings offer excellent opportunities for energy saving and demand response (DR), but these opportunities are rarely utilized due to lack of effective building energy management systems and automated algorithms that can assist a building to participate in a DR program. Considering the low load factor in US and many other countries, DR can serve as an effective tool to reduce peak demand through demand-side load curtailment. A convenient option for the customer to benefit from a DR program is to use automated DR algorithms within a software that can learn user comfort preferences for the building loads and make automated load curtailment decisions without affecting customer comfort. The objective of this dissertation is to provide such a solution.

First, this dissertation contributes to the development of key features of a building energy management open source software platform that enable ease-of-use through plug & play and interoperability of devices in a building, cost-effectiveness through deployment in a low-cost computer, and DR through communication infrastructure between building and utility and among multiple buildings, while ensuring security of the platform.

Second, a set of reinforcement learning (RL) based algorithms is proposed for the three main types of loads in a building: heating, ventilation and air conditioning (HVAC) loads, lighting loads and plug loads. In absence of a DR program, these distributed agent-based learning algorithms are designed to learn the user comfort ranges through explorative interaction with the environment and accumulating user feedback, and then operate through policies that favor maximum user benefit in terms of saving energy while ensuring comfort.

Third, two sets of DR algorithms are proposed for an incentive-based DR program in a building. A user-defined priority based DR algorithm with smart thermostat control and utilization of distributed energy resources (DER) is proposed for residential buildings. For commercial buildings, a learning-based algorithm is proposed that utilizes the learning from the RL algorithms to use a pre-cooling/pre-heating based load reduction method for HVAC loads and a mixed integer linear programming (MILP) based optimization method for other loads to dynamically maintain total building demand below a demand limit set by the utility during a DR event, while minimizing total user discomfort. A user defined priority based DR algorithm is also proposed for multiple buildings in a community so that they can participate in realizing combined DR objectives.

The software solution proposed in this dissertation is expected to encourage increased participation of smaller and medium-sized buildings in demand response and energy saving activities. This will help in alleviating power system stress conditions by employing the untapped DR potential in such buildings.

Development of a Software Platform with Distributed Learning Algorithms for Building Energy Efficiency and Demand Response Applications

Avijit Saha

General Audience Abstract

In the US and many other countries around the world, the daily peak load experienced is frequently much higher than the daily average load. This low load factor causes inefficient use of generation and transmission resources. Besides inefficient use, the peak load also increases system stress conditions resulting from inadequate generation, transmission line outages or transformer failures. This can create supply-limit conditions which may induce cascaded failures and large area blackouts. To avoid system stress conditions due to increasing demand and to use power system resources more efficiently, demand response (DR) serves as an effective tool to reduce peak demand through demand-side load curtailment.

This dissertation focuses on DR applications in buildings. In the United States, buildings consume over 40% of the country's total energy use. These includes both commercial and residential buildings. Most of the commercial buildings are either small-sized (<5,000 sqft) or medium-sized (5,000-50,000 sqft). These buildings offer excellent opportunities for demand response, which can be implemented through use of building energy management/building automation software. But, building automation software is not yet very popular in small and medium-sized buildings due to lack of low-cost and easy-to-use software solutions.

A DR program offered by a utility can be price-based or incentive-based. Price-based DR programs employ dynamic pricing structure to encourage customers to reduce consumption to save bills, whereas incentive-based programs focus on customer commitment to the utility for providing requested load curtailment during peak load situations, in return for monthly or yearly monetary incentives. As most of the peak load reduction potential comes from incentive-based DR programs, this dissertation focuses on an incentive-based DR program. A customer can conveniently participate in such a program by using automated DR algorithms within an energy management software that can control building loads without customer intervention. Providing load curtailment may interfere with customer comfort, and therefore these algorithms must learn customer comfort preferences and consider them while making load shedding decisions.

In this dissertation, a software solution is developed for demand response implementation in buildings, which includes contribution to a secure software platform that enables monitoring and control of loads, and automated learning-based algorithms that can learn customer comfort ranges for building loads and use this learning to make load curtailment decisions in an incentive-based DR program, while ensuring customer comfort.

To humanity, which I have always had and will continue to have faith in…

Acknowledgements

If I want to properly thank everyone who have helped me come this far, this acknowledgement section will most definitely cross the limit of a standard IEEE paper. So, I will only mention the persons I am indebted to throughout the course of my PhD program.

First and foremost, I would like to thank my PhD advisor Prof. Saifur Rahman. He held my hands in a very crucial time for me, and believed in me all throughout, even when I was at my lowest. As a mentor, he has instilled in me the value of independent growth, while providing the supportive environment to thrive in. His knowledge and experience in the field has always helped me to see the big picture and keep the direction of my research aligned with practicality. His discipline and hard work sets a standard for me to reach for my own career. I am ever thankful for the opportunities he provided during my PhD and continues to provide for my career.

I would like to thank Dr. Manisa Pipattanasomporn for being so awesome. She has been a guide, a friend, and a constant source of positive energy. Her smile, support and encouragement helped me get through most of the hard times I faced. She has been a reviewer for my papers, a consultant for deciding my next steps in research, a guide to remind me of my timeline, a supporter for boosting my confidence, and a motherly figure who cooked and brought food for us in the weekends of hard work. I cannot thank her enough for the support she has provided in my PhD.

I would like to thank Dr. Murat Kuzlu for his guidance and support in my research projects. He has always been welcoming for any discussion and help, and his advices helped me fine-tune my work. It has always been a great pleasure working with him.

I would like to thank Dr. Jaime De La Ree, Dr. Alireza Haghighat, and Dr. Guoqiang Yu for providing their valuable time and expertise as my PhD committee members. Their insightful reviews have helped me to improve my work.

I would like to thank US Department of Energy and National Science Foundation for supporting my work through funding the two projects I worked on during my PhD. The feedback I received during project evaluation and Building Technology Office (BTO) meetings have been immensely helpful for shaping my work. I would also like to thank the personnel at Pacific Northwest National Laboratory (PNNL) related to the VOLTTRON™ project. They have always been great supporters and reviewers of our work.

I would like to thank the faculty and students at Yildiz Technical University, Istanbul, Turkey, who have hosted us to conduct research at their smart house facility. I would also like to thank the personnel at AMTI-Advanced Manufacturing Services for providing their expertise during one of our projects.

I would like to thank all my team members in the BEMOSS™ project. It has been a fulfilling experience working with Kruthika and Warodom in the initial phases of the project. Then came Rajendra, Xiangyu, Abdullah, Aditya, Meng Meng, and Imran. It has been a great pleasure working with all of them. Besides BEMOSS™ project, all other students and staff at Advanced Research Institute (ARI) have been great friends and colleagues throughout my PhD. Yonael, Moshiur bhai, Desong, Hamideh, Ade, Fakeha, Sneha – they all have been good friends, and made my life at ARI enjoyable. Also my other friends at VT-NVC have all been sources of energy when I needed them. I would also like to thank all the visiting scholars and researchers at ARI who have enriched my knowledge. Finally, I would like to thank my friends in the Bangladeshi Community both at Blacksburg campus and around the DMV area, especially Omee and Nuva, who have made me feel at home in my PhD life far away from home.

This acknowledgement section will remain incomplete without mentioning my family and friends, who together make up the backbone of my life, and I will not thank them.

My parents have been nothing but the best parents I could ever hope for. They believed in me to always let me choose my own goals in life and provided encouragements along the paths. They got my back when I fell down. They are my pillars of strength. Add to that: my sister, brother-in-law and my beloved niece Aura, whose aura shines up my day.

My wife Shibani has been my partner in all the senses of the word. She is my best friend, my partner in crime, my partner in dreams and aspirations, my partner in pursuing PhD, my significant other. Her family has also been a constant source of support and encouragement.

And lastly my close friends: Hridoy, Arun, and Himu – who I have been friends with for more than 17 years. I am writing this as I look forward to our next get-together.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Background

With the advent of newer, better and cheaper communication technologies, the traditional electric power grid has undergone significant changes in the past few decades. The communication technologies being overlaid on the grid network, a smarter and more reliable grid is emerging, which is commonly known as the smart grid. A smart grid, as opposed to the traditional power grid, has both electric and communication infrastructures that make the grid more observable and controllable, both locally and remotely. Added with automation technologies, the present day smart grid is becoming more and more resilient to undesirable disturbances in the system. The goal of these smart grid technologies including automated generation control, field devices for state observation, and distribution automation and automatic metering, is to make the grid self-healing, reliable and secure.

Within the smart grid, demand response (DR) plays a key role in alleviating power system stress conditions. It can be used to maintain a supply-demand balance even in peak demand situations. Peak load curtailment through demand response can also reduce the burden on transmission and distribution networks and therefore possibly delay the necessity of newer infrastructures to support the ever-increasing demand. Due to these reasons, demand response has been getting a lot of attention for the last two decades. Starting its journey as direct load control within demand side management in 1980s, it has evolved into a more customer-oriented system for efficient peak load reduction opportunities. This has also created the path for energy management systems in buildings, which can automatically implement demand response programs and achieve energy efficiency goals within customer premises.

A thorough literature survey in this field of research points to the need of secure energy management systems and demand response algorithms that are customer comfort oriented and at the same time effective in peak load curtailment. There is an excellent demand response opportunity both in commercial and residential buildings. But to implement demand response in buildings, there is a great need for a cost-effective easy-to-deploy reliable and secure software platform. Demand response algorithms are plentiful in literature, but maintaining customer comfort through learning customer usage behavior is still less explored. The usage of learning customer comfort preferences in making load curtailment decisions in an incentive-based DR program is also not explored. There is also a high demand response potential when multiple buildings are considered, if they can work together as a community to share demand response responsibilities. In case of residential buildings, an additional challenge is to bring the non-communicating hard-wired power-intensive loads of most households into the demand response picture. Among all these needs, there is always a constant requirement of addressing security and privacy concerns in any automated hardware or software platform within the smart grid.

The aim of this dissertation to address most of these knowledge gaps by proposing development of a robust and secure software platform with energy efficiency and demand response algorithms for implementation in commercial and residential buildings. It emphasizes addressing customer comfort concerns, while ensuring the required load curtailment.

1.2. Objective and Scope of the Dissertation

Based on the identification of the knowledge gap from a thorough review of the existing literature, the aim of this dissertation is to provide a complete solution for energy efficiency and demand response in buildings – both commercial and residential. This dissertation contributes to the development of a robust and secure software platform with learning algorithms for use in building energy efficiency and demand response scenarios. The first part builds on an existing secure software platform that allows building loads to be controlled. The second part focuses on devising learning-based algorithms for controlling loads in a building for energy saving. The last part is on developing algorithms that react to a demand response scenario where the learned/fixed preferences are used for realizing the demand response goals without sacrificing customer comfort.

For clarity, the dissertation objective is segmented into multiple objectives with associated goals and sub-goals, and the rest of the chapters in this dissertation are organized to reflect this breakdown. The following are the description of goals to meet the objectives of this dissertation:

Objective 1:

Address the issue of lack of cost-effective open-source software solution for energy management in buildings. This is accomplished by:

Goal 1: Design and implementation of a Building Energy Management Open Source Software (BEMOSS™) platform for small and medium-sized buildings, which can be deployed on low-cost hardware platforms. This dissertation - within its scope - specifically focuses on designing the architecture of BEMOSS™ to address the plug & play, interoperability, scalability and security issues and readiness for demand response signals and multi-building communication and coordination. The sub-goals of this goal include:

a) Develop a discovery agent with auto-discovery algorithms and discovery application programming interfaces (APIs) for seamless plug & play of devices in buildings;
b) Develop API translators within BEMOSS™ to allow interoperability among different communication technologies, protocols and custom-designed APIs of load controllers/devices, so that system can be integrated with different devices to be controlled by a communication-agnostic algorithm;
c) Develop an automated API translator that can streamline integration of devices with similar APIs or communication protocols by generalizing and combining the key points of the APIs;
d) Develop the method of receiving demand response signals in BEMOSS™ through OpenADR;
e) Develop the multi-building communication architecture and distributed data storage and flow for implementation of coordinated demand response algorithms; and
f) Assess and address security vulnerabilities and privacy issues in BEMOSS™.

Objective 2:

Address the issue of lack of learning algorithms in buildings that automatically learn the comfort preferences of users for building loads and utilize them for maximum energy savings while ensuring comfort. This is accomplished by:

Goal 2: Design of reinforcement learning (RL) based algorithms, which automatically learn the comfort preferences of users for building loads using distributed learning agents in a multi-agent system. The sub-goals include:

a) Develop RL based algorithms for three major types of building loads: HVAC, lighting and plug-loads, which can learn user comfort preferences through interaction with loads and accumulation of feedback from users, and then utilize the learning for energy efficiency;
b) Develop RL agents for a building energy management system that use the developed algorithms for automated learning and control and are sufficiently light-weight for deployment in low-resource machines;
c) Validate the efficacy of the algorithms for building loads using simulated load models and user feedbacks.

Objective 3:

Address the gap of incentive-based DR algorithms in buildings that can make load curtailment decisions in a demand restricted scenario based on fixed user defined priorities or learned comfort preferences using learning agents. This is accomplished by:

Goal 3: Design of incentive-based DR algorithms, which utilize pre-defined or learned user preferences for making load curtailment decisions. The sub-goals include:

a) Develop an incentive-based DR algorithm for residential loads that is robust against communication issues and uses user-defined fixed priority and comfort preferences to maintain demand constraints, while utilizing set-point control in smart thermostats and distributed energy resources (DER) - specifically Photovoltaics (PV) and storage if available;
b) Develop an incentive-based DR algorithm for commercial building loads that utilizes the learning from the RL agents for user comfort preferences and makes optimal load curtailment decisions to maintain demand limit, while minimizing user discomfort.
c) Develop an incentive-based cooperative DR algorithm for multiple buildings in a community/campus, that uses user defined priority for building zones and learned comfort

preferences for loads to control loads in multiple buildings to meet a combined demand curtailment requirement;

d) Validate the ability of developed DR algorithms in real/simulated buildings.

1.3. Contributions

1.3.1. Development of a Building Energy Management Open Source Software (BEMOSS™)

From the literature search, it is evident that a lot of demand response potential in small to medium-sized buildings is untapped due to lack of inexpensive solutions for building energy management systems. This dissertation contributes to an open-source energy management software solution named BEMOSS™ under development at the Virginia Tech - Advanced Research Institute. BEMOSS™ capabilities have been enhanced by this work; specifically, the features of auto-discovery and plug & play, interoperability with different communication technologies and protocols and Application Programming Interfaces (APIs), capability of receiving and interpreting demand response signals and multi-building communication and coordination. The security vulnerabilities and privacy issues are analyzed and countermeasures and security features are proposed to make the software platform more secure and reliable.

1.3.2. Development of Learning Algorithms for Building Loads

A multi-agent system with RL algorithms is proposed to determine the comfort preferences of user for different types of loads. The learning algorithms are developed based on the three major types of loads in commercial buildings: HVAC, lighting and plug loads. The algorithms explore the load state space in search of control options with more energy savings. They accumulate negative feedbacks from users to re-adjust reward values for explored states. Finally, with sufficient training, the algorithms exploit the knowledge and operate loads for maximum possible energy savings while within customer comfort range.

1.3.3. Development of Incentive-based DR Algorithms for Buildings

Two sets of incentive-based DR algorithms are proposed for peak load reduction. A DR algorithm for residential buildings is proposed that uses user defined priority and preference settings for power intensive loads in houses. This algorithm considers communication delays and errors and is robust against them. The algorithm controls HVAC through set-point control of smart thermostats and controls other power-intensive loads through ON/OFF control. If available on building premises, distributed energy resources (DER), specifically PV and storage management issues can be addressed by this algorithm. Another DR algorithm is proposed for commercial buildings, which utilizes the learning algorithm to make peak load curtailment decisions while ensuring customer comfort. Finally, a coordinated DR algorithm is proposed for multi-building scenarios that can utilize learning-based algorithms in each building and user defined priority of building zones to make optimal decisions of load curtailment in a cooperative environment.

2. Literature Review

This chapter delves into the background of this dissertation by discussing relevant work in its field of research. The chapter starts from top level concepts of the overall picture where this research fits into, and then gets deeper into most relevant work aligning with this dissertation. The objective of this chapter is to establish the relevance of this dissertation in its field of study. The first part of this chapter discusses the relevant literature and the second part identifies the knowledge gaps.

2.1. Literature Search

2.1.1. Basics of Demand Response

2.1.1.1. Background

In the US and many other countries around the world, the daily peak load experienced is frequently much higher than the daily average load. This low load factor causes inefficient use of generation and transmission resources. For example, in the area serviced by Dominion Virginia Power, roughly 20% of the generation assets are used only 5% of the time [1]. Besides inefficient use, the peak load may also cause system stress conditions. Over the past few decades, the electric power system has encountered frequent stress conditions due to continuously increasing demand [2]. This stress conditions can be caused by transmission line outages or transformer failures due to uncontrolled peak loads. This can create supply-limit conditions, which may induce cascade failures and large area blackouts. To avoid system stress conditions due to increasing demand and to use power system resources more efficiently, demand response (DR) can serve as an effective tool to reduce peak demand through demand-side load curtailment [3].

Demand Response was preceded by Demand Side Management (DSM) in the 1980s [4, 5, 6, 7] to tackle this same problem. The idea of demand side management was geared towards utility's control of demand side resources. This means that utility would control the customer side load as needed to match the supply resources. A very straightforward implementation of DSM is direct load control (DLC) [8], where customer gives consent to utility to directly cut its load if required to maintain supply-demand balance. DSM is effective in its purpose, but the downside is the possibility of major customer inconvenience during the actual load cut, because it is a one-directional command from the utility to the customer load and doesn't take into account customer's situation at the time. With advent of time and improved technologies in smart grid, especially after advanced metering system came into picture, the concept of DSM converted into demand response (DR), which offered more options for customer participation in demand curtailment, taking advantage of the two-way communication infrastructure.

The definition of demand response according to US Department of Energy (DOE) [9] and also used by Federal Energy Regulatory Commission (FERC) [10], states demand response as:

"*"Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized."*

FERC also categorizes the demand response activities into 14 classifications in two groups: incentive-based and time-based [11]. These classifications are:

Incentive-based DR programs:

This group of DR programs rely on some sort of incentive-based payments or contractual arrangements made in advance between utility and customers that requires the customer to reduce demand when a demand response signal is sent from utility. This category includes eight type of DR programs, which are:

1) *Direct load control*
2) *Interruptible load*
3) *Load as capacity resource*
4) *Spinning reserve*
5) *Non-spinning reserve*
6) *Emergency demand response*
7) *Regulation service*
8) *Demand bidding and buy back.*

The most commonly used of these are *direct load control* and *interruptible load,* which include direct intervention of utility in controlling demand.

The *load as capacity resource* is the program where customer commits to make pre-specified load reduction when system contingencies arise.

The *spinning reserve* is the commitment of a synchronized demand-side resource that can be shed within the first few minutes of a signal.

The *non-spinning reserve* on the other hand is the demand-side resource that can be shed after a delay of ten minutes or more.

The *emergency demand response* offers load reduction capabilities in response only to an emergency event.

The *regulation service* is the commitment to become a demand-side resource that can both increase or decrease the load in response to real-time signals.

The *demand bidding and buy back* is a program that allows a demand resource to bid in electricity retail and wholesale markets for load reduction at a price.

Time-based DR programs:

Also known as price-based DR programs, this group of DR programs rely on different types of time-varying price signals to encourage customers to reduce demand. These include:

1) *Critical peak pricing*
2) C*ritical peak pricing with direct load control*
3) *Time-of-use pricing*
4) *Real-time pricing*
5) *Peak-time rebate*
6) *System peak response transmission tariff*

The *critical peak pricing* program charges a pre-specified high electricity tariff during the peak load periods to discourage customers from increasing demand during that period.

The *critical peak pricing with direct load control* combines direct load control with a critical peak pricing approach. The *time-of-use pricing* program introduces varying electricity tariffs by time periods that are longer than one hour within a day.

The *real-time pricing* is also varying electricity tariff, but it fluctuates hourly or more often.

The *peak-time rebate* allows customers to earn a rebate by reducing demand during a specified period on critical peak days.

The *system peak response transmission tariff* program increases transmission charges during peak load periods to encourage load reduction.

These different DR programs have varied DR potentials according to FERC [11]. As can be seen in Figure 2.1 [Source: 11], almost 80% of the peak load reduction potential comes from incentive-based DR programs. This is due to the fact that time-based DR programs cannot guarantee load reduction, as customers may choose to consume high amount of energy during peak periods despite increased tariffs.

Figure 2.1 also shows that commercial and industrial customers offer higher peak load reduction potential than residential customers.

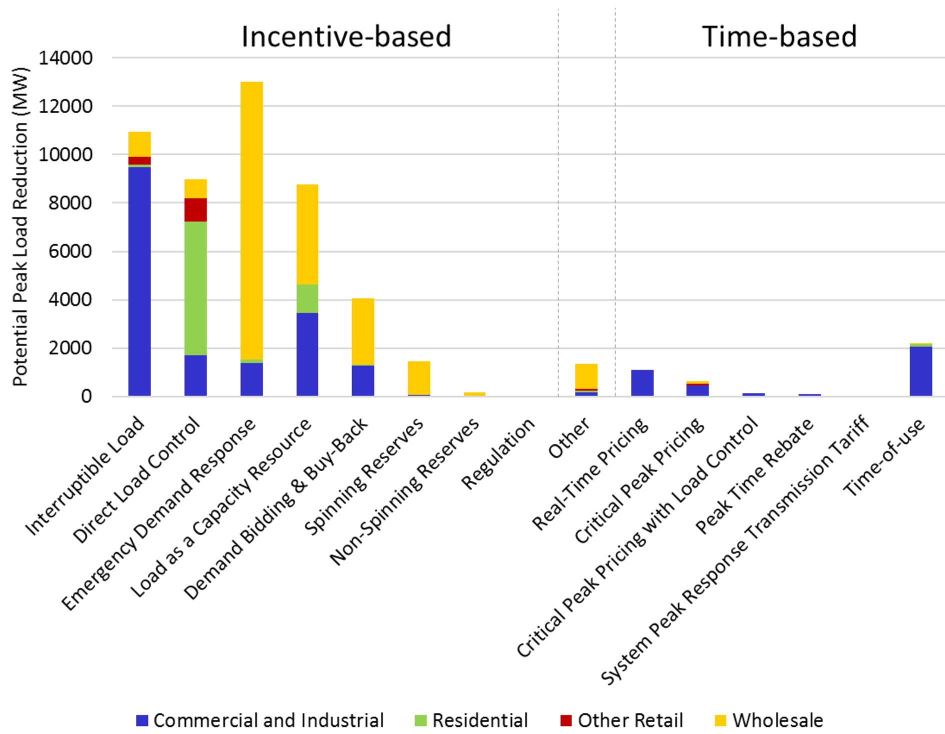


Figure 2.1. Reported potential peak load reduction by type of program and customer class [Source: 11]

In terms of automation, Demand Response can be three types:

1) Manual DR
2) Semi-automated DR
3) Fully Automated DR or auto-DR [12]

The fully automated DR is the most convenient option for customer as it can employ DR by automatically controlling loads without customer intervention. With the rise in Home and Building automation systems, implementation of auto-DR is also gaining widespread popularity.

2.1.1.2. Deployment of Demand Response and Smart Grid Infrastructure

There are several pilot projects and implementation of infrastructure for Demand Response and automation of grid in distribution side.

A "Perfect Power" system prototype following the High Reliability Distribution System design developed by S&C Electric was built on the campus of Illinois Institute of Technology [13], which focuses management of the campus electricity distribution and usage and includes coordination with ComEd and the PJM ISO for ancillary services and demand response.

In 2008, Xcel Energy launched SmartGridCity™, a fully integrated smart grid community to include emerging smart grid technologies [14].

PG&E's SmartAC program [15] focuses on controlling central AC units in households and small/medium businesses by installing programmable communicating thermostats (PCTs) and direct load control switches.

There are other smart thermostat programs like Energy Smart Thermostat Program by Southern California Edison [16] and the smart thermostat program at San Diego Gas & Electric Company [17], which evaluate the impact of installing and utilizing smart thermostats in customer premises.

In 2007, the Smart Meter Pilot program initiated the PowerCentsDC program [18], which evaluates the impacts of smart prices, smart meters and smart thermostats on consumer behavior.

A two-year collaborative research was conducted by LBNL and the Demand Response Research Center under the PowerChoice label which focused on studying behavioral response of consumers on voluntary time-of-use pilot rate offered by the Sacramento Municipal Utilities District [19].

From European perspective, there are numerous initiatives related to innovative infrastructures for smart grid automations. For example, projects like SmartHouse/SmartGrids [20], BeyWatch [21], BeAware [22] focus on energy efficiency. There are projects on Internet of Energy (e.g. Smart Watts [23]) and semantic technologies for energy efficiency (e.g. SESAME [24] and SESAME-S [25]). There are also demand response initiatives like DR 3 [26] and SmartCamp [27], and smart solutions for home automation (e.g. EcoGrid [28], DiYSE [29]).

There are many other running or under review projects for smart grid and demand response which can be found in the database of Virginia Tech's Smart Grid Clearing House [30].

2.1.1.3. Demand Response: PJM Perspective

PJM interconnection [31] is a regional transmission organization (RTO) that acts as a neutral independent party to operate a competitive wholesale electricity market and manage the high-voltage electricity grid in all or parts of 13 Mid-Atlantic States and the District of Columbia. PJM does not directly interact with the electricity consumers, instead it interacts with the entities that buy the electricity from its wholesale market or produces electricity and then sells electricity to the end customers. PJM encourages demand response (DR) activities to help reduce wholesale electricity prices and to reduce electricity usage during high demand periods to potentially avoid using less efficient generation resources to meet high demand [32]. For implementing demand response, PJM works with curtailment service providers (CSP) [33], which are the entities directly responsible for demand response activities of the consumers in the PJM wholesale market. The CSPs are responsible for identifying demand response opportunities with customers and then provide the operational infrastructure to enable demand response both at customer premises and the wholesale market.

PJM’s DR opportunities allow electricity consumers to earn a revenue stream for reducing consumption during periods of high wholesale prices or when the reliability of the system is threatened. There can be two broad types of DR participation: emergency or economic, and a customer may participate in any or both depending or circumstances.

Emergency demand response represents a mandatory commitment in which the participants commit to respond to PJM’s request to reduce load or only consume up to a certain level during an emergency operation condition or supply shortage. PJM considers these resources similar to a generator and expects them to comply during the time of need, and therefore non-compliance to the commitment is penalized. CSPs are responsible for managing the portfolio of their customers to meet these obligations in order to avoid penalties. The participation to this form of demand response is encouraged by revenue streams driven by “capacity” market as defined under Reliability Pricing Model (RPM) [34].

On the other hand, economic demand response represents a voluntary commitment to reduce load when the wholesale electricity price is higher than PJM’s published monthly net benefits price. An economic demand response is expected to displace a generation resource by realizing load reduction signal dispatched by PJM. The economic demand response resources may participate as ancillary services like: synchronized reserve (ability to reduce load within 10 minutes of PJM dispatch), day ahead scheduling reserve (ability to reduce load within 30 minutes of PJM dispatch) and regulation (ability to follow PJM's regulation and frequency response signal).

Through the evolution of economic demand response, PJM is now implementing price responsive demand (PRD) [35], which is considered as a predictable change in demand in response to changing wholesale electricity prices. PRD requires three aspects to be fulfilled: 1) it must be served under a dynamic retail rate structure linked to the wholesale electricity price to provide predictable response to changing prices, 2) must be subject to supervisory control to curtail any portion of the committed demand that has not responded to price in case an emergency situation occurs, and 3) subject to advanced metering capable of recording consumption at hourly or higher frequency.

2.1.1.4. PJM's Demand Response Signal and OpenADR

The Open Automated Demand Response Communications Specification, also known as OpenADR [36], began to be developed in 2002 following the California electricity crisis, to address the need for an open-standards based communication data model to promote communication exchanges between utility and customer to enable automated demand response. The specification was primarily adopted in California and is now referred to as OpenADR 1.0. It was donated by Lawrence Berkeley Laboratories to the Organization for the Advancement of Structured Information Standards (OASIS) in 2009 [37] for their development of the Energy Interoperation Standard – Energy Interop™ [38]. It followed the efforts of NIST in forming the Smart Grid Interoperability Panel (SGIP) [39] to create standards for energy communications. OpenADR 2.0 is a profile within the Energy Interop™ standard and offers data models to facilitate DR programs, wholesale market transactions, ancillary services etc. The OpenADR alliance was formed in 2010 to encourage adoption of OpenADR 2.0 profile specifications. OpenADR 2.0 uses web services to communicate and covers activities like scheduling DR events, opting in and out of events, registrations for trusted communications and reporting of meter readings or other data.

PJM worked with IPKeys Technologies, LLC to design its Advanced Technology Pilot project [40, 41] to use OpenADR 2.0 and demonstrate demand response, price conveyance, price responsive demand (PRD) and real time telemetry for verification of load shed. This pilot project was demonstrated in spring of 2012 [42] and included other participants: Walmart in the role of a large scale chain central control system, American PowerNet (APN) acting as a demand aggregator, Redners as a grocery store chain with no building automation, and Perfectly Green Corporation's cogeneration systems to fill the role of distributed energy resource (DER) [43]. IPKeys developed a cloud based OpenADR 2.0 Energy Interop™ Server and System (EISS™), which was used by all participants to implement the pilot. IPKeys implemented a REST/ poll based client-server architecture, in which the clients polled the server in a request/response fashion to get pricing/demand response signals. Walmart used a OpenADR 2.0 client in its central control system, Perfectly Green Corporation used the automated relays in EISSBox which is a java-based client developed by IPKeys in a linux single board computer, and Redners used EISSBox to receive signal and then a "human in the loop" employee to manually turn off circuit breakers. The EISS™ server was responsible for collecting and interpreting PJM signals of LMPs (Locational Marginal Prices) and traditional DR signals. The pilot demonstrated three use case studies: a traditional demand response event conveyed to end customer, a price responsive demand scenario with conveyance of PJM's LMPs to end customer and logic to load shed according to price, and verification of load shed through real-time meter telemetry. The pilot showed how OpenADR 2.0 can be implemented in an end-to-end electricity market scenario to enable demand response.

The request/response or pull method was changed to a push method, where OpenADR 2.0 server actively sends the signal to end devices when data is available. This was included in OpenADR 2.0b profile which adopted XMPP. Multi-phase pilot program with participants: PJM, IPKeys, LBNL, Walmart and Schneider Electric demonstrated the feasibility of using OpenADR 2.0b profile for ancillary services and regulation signaling. A press release by IPKeys in May 2014 announced the successful completion of the Pilot project [44, 45]. IPKeys EISS™ translated the PJM synchronous reserve (SR) web service signal into an OpenADR 2.0b message and sent to the client in Walmart's Building Management system to successfully initiate control if a series of lighting and HVAC loads.

The pilot projects demonstrate that OpenADR 2.0 and OpenADR 2.0b can be used to enable demand response in a PJM wholesale market. Therefore, in this dissertation an OpenADR 2.0

client agent will be used to receive DR or pricing signals to implement demand response within the multi-agent framework for building energy management.

2.1.2. Demand Response in Commercial Buildings

In the United States, buildings consume over 40% of the country's total energy usage [46]. 90% of these buildings are either small-sized (<5,000 sqft) or medium-sized (5,000-50,000 sqft) (Figure 2.2) [Source: 47]. Hence these small or medium-sized commercial buildings offer excellent opportunities for demand response. The U.S. Department of Energy (DOE) targets to save $2.2 trillion in energy-related costs by reducing building energy use by 50% compared to the 2010 baseline [48]. The energy savings in buildings can be made using a Building Energy Management System (BEMS) or Building Automation Systems (BAS).

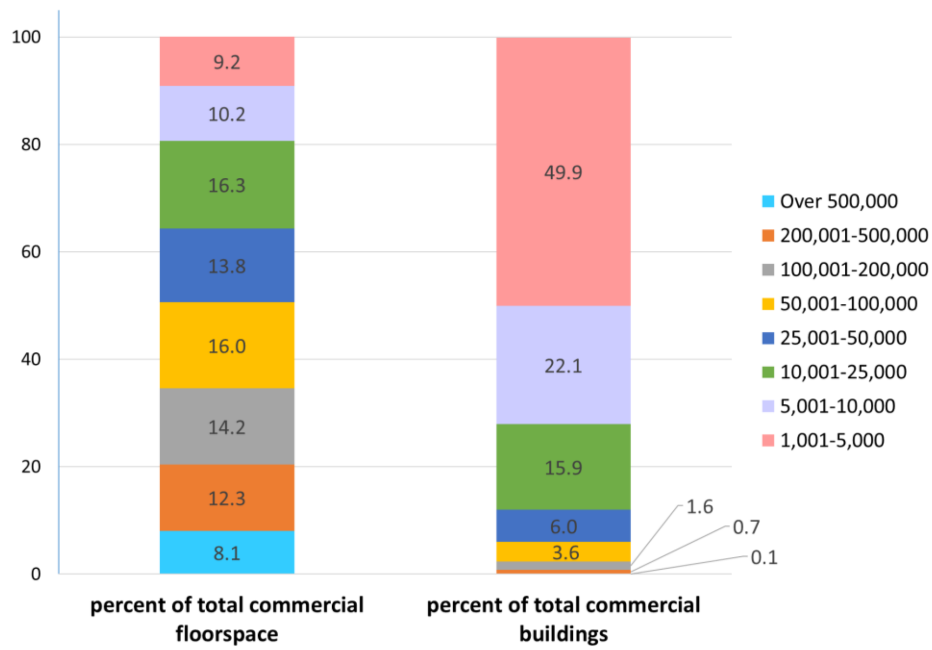


Figure 2.2. Commercial building sizes according to commercial building energy consumption survey [Source: 47].

2.1.2.1. Research on Building Energy Management/Building Automation Software

BEM/BAS is an automated system that monitors and controls building loads. It is essential for implementing energy efficiency and demand response in a commercial building. The term BEM is distinguished from the term HEM, where the latter is only used for residential building, and the former is mostly used in case of non-residential buildings, but may include residential buildings as well.

The study [49] shows that due to the lack of building monitoring and control, significant portion of the energy consumed in buildings is wasted. This is because existing Building Automation Systems (BAS) are still cost-prohibitive and being used mostly in large buildings. BAS are not popular in most small- and medium-sized buildings due to:

a) lack of awareness of benefits
b) lack of inexpensive packaged solutions, and
c) sometimes due to the owner not being the tenant and so finding no incentive to invest in these systems.

Recently, more and more commercial products for home/building automation systems have become available to tackle these problems. Some examples of these products are SmartThings [50], Staples Connect^{TM} [51], GE Brillion^{TM} [52], Lowe's Iris [53], Revolv [54] etc. These solutions allow homeowners or small building owners to monitor or control specific compatible devices. Most of these solutions are limited in terms of open-source development and interoperability and are not very cost effective.

Some open source building automation solutions are available such as Freedomotic [55], OpenRemote [56], and openHAB [57] etc. These platforms provide open, flexible architectures, and hardware/protocol agnostic tools for developing residential or commercial automation. The intelligence of buildings can be enhanced to implement automated rules, scripts, or event triggering for a specific device(s). But, autonomous communication and coordination among devices has not been addressed properly.

2.1.2.2. Loads in Commercial Buildings

For small- and medium-sized buildings, heating consumption is the dominant end use, followed by lighting, plug loads and cooling [58]. Specifically, Heating, Ventilation, and Air-Conditioning

(HVAC), lighting and plug loads account for most of the consumption in buildings (Figure 2.3) [47].

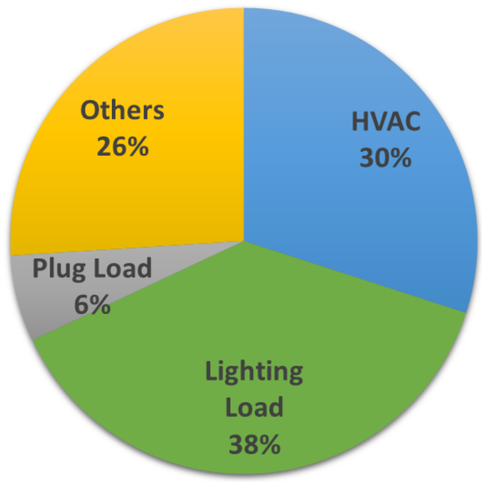


Figure 2.3. Electricity use in buildings by load types according to commercial building energy consumption survey [47].

Existing literature on BEM also consider HVAC and lighting loads for energy reduction [59]. Based on these, the loads in a commercial building that have the highest DR opportunities are:

a) Heating, Ventilation and Air Conditioning (HVAC) loads
b) Lighting loads and
c) Plug loads

2.1.2.3. Communication in Building Automation

BAS employ a wide range of communication technologies and protocols for communicating with devices and other system components. Authors in [60] discuss and assess different communication technologies and protocols for different smart grid applications. The most widely used communication technologies used in BAS can be found in Table 2-1.

Table 2-1. List of Communication Technologies used in Building Automation

Technology	Standard/ Protocol	Max. Theoretical Data Rate	Coverage Range
Wired Communication Technologies			
PLC	HomePlug	14-200 Mbps	up to 200 m
Ethernet	IEEE 802.3	10 Mbps- 1 Gbps	up to 100 m
Serial	RS-485	100 kbps – 35 Mbps	up to 1,200 m
Wireless Communication Technologies			
ZigBee	ZigBee	250 kbps	up to 100 m
	ZigBee Pro	250 kbps	up to 1,600 m
WiFi	802.11x	2-600 Mbps	up to 100 m
Z-Wave	Z-Wave	40 kbps	up to 30 m
EnOcean	ISO/IEC 14543-3-10	125 kbps	up to 30 m

The BAS and smart devices also use various communication protocols. The most commonly encountered communication protocols are listed in Table 2-2.

Table 2-2. List of Communication Protocols used in Building Automation

Communication protocol	Allow communication over:					
	Power line	Ethernet	Serial	WiFi	ZigBee	2-wire
1. BACnet (IP)		X		X		
BACnet (MS/TP)			X			
2. Modbus (RTU)			X			
Modbus (TCP)		X		X		
3. LonWorks	X	X		X		X
4. KNX	X	X	X	X		
5. DALI						X
6. M-bus						X
7. Web		X		X		
8. OpenADR		X		X		
9. Smart Energy Profile	X			X	X	

2.1.2.4. Cyber-security Concerns of Building Automation Systems

A smart grid, as opposed to the traditional power grid, is a very complex dynamic network of inter-connected devices for information exchange, decision-making, and actuation. Two of the key challenges for smart grid implementation are security and interoperability [61], and they should be addressed as a part of the design problem instead of after-thought.

For automated DR implementation within the smart grid, most building energy management systems typically rely on smart appliances/load control devices within a wireless/wired network, and make automated decisions of load reduction based on pricing/control signals provided by the utility. This exposes areas of security vulnerability that need to be addressed to protect the data and customer privacy. For example, wireless networks or communication channels between a utility and customer premises may become a possible target for security attacks. This opens up many concerns regarding the security of smart grid and privacy of customers participating in DR programs. This area is receiving increasing interests from the scientific community.

Work has been reported that identifies security vulnerabilities in wireless sensor networks for demand response applications [62, 63, 64, 65]. Authors in [66] propose role based access control and publish/subscribe based communication for security of DR applications. Authors in [67] propose an efficient privacy-preserving demand response (EPPDR) scheme with adaptive key evolution. Authors in [68] discuss security challenges in OpenADR2.0 based DR systems in cloud.

There is also work that addresses security concerns in advanced metering infrastructure (AMI). For example, author in [69] discuss the security concerns in AMI and possible countermeasures. Some of the previous work [70, 71, 72] propose key management schemes for secure communication in AMI.

Some of the previous work focuses specifically on smart homes and home energy management systems. Authors in [73] assess vulnerabilities and devise a possible defense mechanism against pricing-based cyber-attacks on smart homes. Authors in [74] discuss the security issues and countermeasures for smart homes in a smart grid environment. Authors in [75] propose an elliptic curve cryptography based key establishment protocol for home energy management systems.

Building automation systems are also being scrutinized for cyber-security vulnerabilities. Authors in [76] identify and address vulnerabilities in building automation systems and SCADA systems. Authors in [77] also discuss about threats and mitigation approaches for intelligent buildings. Authors in [78] analyze the security of building information systems from the intelligent city perspective.

2.1.3. Demand Response in Residential Buildings

As we have already seen in Figure 2.1, residential customers offer significant peak load reduction potential through demand response. Most of it comes from incentive-based programs, but price based programs also offer good opportunities. For an automated demand response program for a residential household to work, an entity has to be present to receive the demand response signal and control household appliances based on it. This entity can be a Home Energy Management (HEM) System. HEM is an essential part of the household for successful DR implementation [79]. It monitors and analyses consumption of various household appliances, considers customer comfort and preference, and controls to achieve energy efficiency and demand response goals [80]. With emerging trends in wireless sensors and actuators and increased penetration of plug-in electric vehicles in residential customer premises [81, 82], the energy management system is also undergoing transformation from traditional manual system to fully automated system for demand curtailment. It may also offer ancillary services [83].

2.1.3.1. Research on Home Energy Management (HEM) Systems

With continually increasing interest in residential demand response in the form of home energy management systems, there are various studies on HEMs and automated residential DR programs.

Many of the existing literature focus on designing effective DR algorithms for HEM. Most of them rely on pricing based algorithms. Authors in [84] focus on dynamic electricity price for home energy simulation. Authors in [85] also focus on pricing based algorithm and utilize price prediction for optimizing load control in a real time pricing environment. Authors in [86] also base their algorithm on real time pricing. Authors in [87] use a heuristic optimization model to solve the load control decision-making problem.

There are also incentive-based demand response algorithms. Authors in [88] propose an incentive-based optimization algorithm for demand response. Authors in [89] propose an incentive-based algorithm that takes into account customer defined priority and preference settings.

There has also been interest in designing complete HEM system with Hardware platform [90, 91]. To enable DR implementation, different communication technologies for load control have been implemented, i.e., ZigBee [92], power line carrier (PLC) [93], ZigBee and PLC [94], infrared-based remote controls with ZigBee communication [95], and BACnet [96], and cloud based technologies [97].

Different methodologies have been applied for DR algorithm. For example, authors in [97] implemented a dynamic priority based scheduling scheme; authors in [86] employ a convex programming based solution, whereas authors in [98] employed dynamic programming based approaches. In existing literature, most work provide simulation results for proposed algorithms [99, 100, 101, 102], with very few papers focusing on real experiments in smart house [103].

Renewable energy (RE) sources offer alternate sources of energy that are environment-friendly and sustainable. Considering continuous decline of fossil fuel resources and its impact on the climate, renewable energy resources are becoming more popular and the technology is getting better with time. With the movement of converting the traditional power grid to a smart grid, challenges of integrating RE sources with the smart grid are being studied [104, 105]. Distributed RE sources may potentially delay upgrade of transmission and distribution infrastructures with increasing demand. Therefore, RE resources at the residential level should be utilized to get the most benefits out of residential demand response (DR) programs.

Recent research topics are related to designing HEM systems with RE integration. Authors in [106, 107, 108, 109] consider this problem as a stochastic optimization problem to minimize energy costs based on dynamic pricing and stochastic nature of renewable energy output. In [97, 110, 111], DR programs are proposed that use a dynamic priority based appliance scheduling based on forecast of RE capability. Authors in [112] address the use of battery, discussing dependence of HEM algorithms on battery characteristics. Also, their optimization techniques rely on dynamic pricing. Authors in [113, 114] propose DR approaches based on dynamic pricing. In [115, 116], HEM systems integrated with PV have been proposed. Authors in [117] discuss a method of scheduling EV charging within a DR program considering PV availability.

2.1.3.2. Residential Loads for DR

Different loads in a residential building offer different load curtailment potentials. Existing literature in demand response have proposed different loads in the household for control. For example, authors in [118] employ stochastic approach for refrigerator control. PJM has a pilot project demonstrating water heater participation in their markets [119]. Authors in [120, 121, 122, 123, 124] all develop algorithms for scheduling selected loads in residential buildings.

Authors in [125] have provided load profiles of selected major household appliances and discussed their demand response opportunities. Based on this, the household loads that are most suitable for demand response application are:

Clothes dryer: An electric clothes dryer offers the highest DR opportunity. It is possible to only disconnect the heating coil while keeping the tumbler running.

Electric Water Heater: Electric water heaters can be good candidate for DR, provided that the hot water temperature can be monitored to decide customer comfort.

AC unit: AC unit offers some DR opportunity through controlling it via a smart thermostat.

Dishwasher: Dishwasher can also be part of a DR algorithm, but it should be taken into account that some dishwasher models cannot resume operation if interrupted in mid operation. Hence their start time can be deferred.

Clothes washer and Refrigerators: These offer very low DR opportunity which can only be availed if they are smart appliances.

Electric vehicle: An electric vehicle can be a very good candidate for DR. Not only rescheduling its charging can provide load curtailment, but also it can be used as energy resource if such hardware infrastructure exists.

Other loads in a residential household either have low electricity usage or are mostly of critical importance to customer. Hence they offer no DR opportunity.

2.1.3.3. Load Controllers for Residential Loads

For communication and control of household loads, most HEM solutions rely either on an existing network of wireless sensors [126] or new design of smart outlets/smart plugs/smart sensors in a system.

There exist commercial HEM solutions, e.g., from Schneider Electric [127], GE [128], Honeywell [129] or SmartThings [50]. Most of these products are only compatible with selected proprietary smart plug products and do not follow any open standards.

At the same time, there are a number of off-the-shelf smart plug products available in the market that can be used with some HEM systems. Examples are: Digi Xbee smart plug [130] from Digi International, remote appliance controllers [131] by EnTek International LLC, Ecobee smart plug [132] from Ecobee, WeMo switch [133] from Belkin, The Energy Detective [134], ThinkECO [135], Kill-A-Watt [136], Plug Smart [137], Alert Me [138], Enistic [139] etc.

Most of the existing smart plug solutions can only control 120V loads with current rating limited to a maximum of 15A.

2.1.4. Reinforcement Learning in Building Energy Management and DR

A reinforcement learning algorithm is ideal in situations where interaction with environment and corresponding adaptation is required. It has gained wide-spread interest in interdisciplinary fields to learn and adapt control strategies based on the feedback from the environment. It is already popular in robotics and artificial intelligence due to its model-free nature and optimal decisions based on current state of environment and learned knowledge from previous interactions with the environment. The idea of reinforcement learning is based on a Markov Decision Process (MDP) model, where a system is defined by its various states and how it goes from one state to another based on environment variables and user actions. The most popular reinforcement learning, Q-learning, associates rewards with each state-action pair and calculates these rewards based on interactions with the environment. The control decisions are made in order to maximize the rewards. The reward values are updated with each feedback from the environment, and therefore the algorithm can make optimal control decisions that adapt to the changing environment. Also, a prior knowledge of the model of the environment is not required, as it can learn the model through experimenting with the environment.

Recent research works have started investigating into the use of reinforcement learning in building load controls and demand response. Some of the existing literature only focus on utilizing reinforcement learning for energy conservation and comfort through control of HVAC system in a smart building [140, 141, 142, 143]. Authors in [144] only focus on ventilation systems. Authors in [145, 146] use simulated reinforcement learning control for building thermal storage inventory. Authors in [147, 148] apply RL for lighting control. In most of these works, Q-learning algorithm is employed where the interaction of the HVAC system with environmental variables and the energy pricing mostly govern the solution map. The dissatisfaction of customer is measured by a predefined satisfaction model, and a real-time interaction with the user is not accommodated. Authors in [142] take into account user interactions with thermostats, but do not consider other loads in the picture. Some works employ a device-based approach to incorporate all loads in the control algorithm. Authors in [149] apply Q-learning for loads in a smart home. [150, 151] utilize a batch reinforcement learning as opposed to Q-learning to save computation time. Although it reaches solution faster, it fails to incorporate real-time user feedbacks and control decision differences that do not fall into the pre-defined batches. Authors in [152, 153] utilize a device-based Q-learning algorithm. This method implements the control decisions after collecting requests from the user and scheduling loads based on these requests and previous learning, which means some of the user requests may be delayed or not completed. This approach is not a preferable solution due to customer dissatisfaction associated with inability to control loads on demand. Also, most of the existing literature fail to categorize the difference in control of HVAC, lighting and plug loads, and the corresponding differences in the reinforcement learning algorithm.

2.1.5. Coordinated Demand Response in Multiple Buildings

The concept of coordination of multiple buildings to reach optimal demand response solution has recently been gaining interest in the scientific community. This is further accelerated by the adoption of distributed energy resources at customer premises, which can be shared among multiple customers to reach optimal DR strategies.

Consumers who can generate and share energy with the grid have been identified as "prosumer". Rathnayaka et al. in [154] have proposed a concept of goal-oriented virtual prosumer communities for effective participation and management of prosumers by first identifying parameters that influence energy sharing behaviors of prosumers and using them in a multi-agent based model to form optimal prosumer communities. They have further developed the concept in [155] by formulating a method of ranking the prosumers based on their behavior and influence in sustaining the community group in long term.

Ciuciu et al. in [156] propose a framework that includes smart-metered homes and small medium enterprises (SMEs) in a demand response decision support system which allows prosumers and SMEs to participate in renewable energy exchange. Authors in [157] propose a coalition coordinator to maximize coalition utility among prosumers in a DR program.

Many authors in literature approach the problem with multi-agent systems to model key players in the system and solve distributed algorithms.

Authors in [158] propose a multi-agent system with home agents and grid to investigate maximization of user's utility, while using random models for diversity of renewables and storage and prediction of customer demand and prices.

Gamauf et al. in [159] propose a building agent as a load management gateway for communicating with other building agents to determine optimized schedule for providing load shifting potential to the grid.

Authors in [160] model the interaction between smart buildings and smart-grid by suggesting three interactive layers: building layer, neighborhood layer, and smart grid layer, and propose a multi-agent based system to employ stochastic optimization problems with different objectives in different layers and a general framework to combine all of them.

Authors in [161] also propose a multi-agent system using a bi-level programming problem where the upper level is the system-wide demand response management tool that tries to flatten total load profile, whereas the lower level are the HLM agents which try to minimize their energy expense.

Authors in [162] address the multi-agent optimization problem for optimal power flow and energy-sharing among smart buildings by proposing an advanced multiple traveling salesman problem optimization to identify optimal sub-routes for energy-sharing among buildings.

Authors in [163] present a model of Smart City using multi-agent system and Internet of Things, which is a basic infrastructure model for providing intelligence to the components of a smart city. The concept of smart city includes a network of smart buildings with seamless communication. Authors in [164] have identified eight critical factors of smart city initiatives: management and organization, technology, governance, policy context, people and communities, economy, built infrastructure, and natural environment.

Many existing works in literature utilize game theoretic approach to optimize the coordinated scheduling of multiple buildings using Nash equilibriums.

Authors in [88] propose an incentive-based autonomous energy consumption scheduling algorithm among multiple residential buildings sharing a common energy resource. They expand this in [85] which employs a Nash equilibrium based distributed algorithm to reach the optimal scheduling solution based on a proposed smart pricing tariff.

Authors in [165] take a similar approach by using mixed integer programming for scheduling consumption and a game-theoretic model to find Nash equilibrium among the objectives of different residential customers, but additionally they take into account local energy resources.

Authors in [166] also formulate a day-ahead optimization problem based on non-cooperative game theory to find Nash equilibrium for end users to reduce their monetary expense.

Authors in [167] approach the game theoretic problem in two ways: first, in the presence of a central unit which can provide the aggregated consumption profile to all consumers, and second, without a central unit, the consumers communicate their estimated information to neighbors for a distributed synchronous agreement-based or asynchronous gossip-based algorithm.

Authors in [168] consider the storage devices in the problem with the users being able to sell back stored energy. They treat the problem with a Stackelberg game to find the equilibrium with minimum cost and peak-to-average ratio.

[169] only focuses on consumer HVAC systems and proposes a distributed control algorithm to solve the non-cooperative game problem.

A few papers consider Markov decision process models for the optimization problem.

Authors in [170] take into account appliances that are flexible in deferring their operating times and propose a decentralized stochastic optimization for coordinated HEM based on a model predictive certainly equivalent control (CEC) method in Markov Decision Process model.

Authors in [171] propose to use a hidden semi-Markov model for power profiles in different households and a coordination based algorithm to find agreement point between users and aggregator.

There are also interest in other optimization algorithms for coordinated demand response, including multi-objective optimization, parallel programming, Analytic Hierarchy Process and Genetic algorithms among others.

Authors in [172] propose the management of energy consumption of a group of users by load scheduling based on a constrained multi-objective optimization problem using two evolutionary algorithms (EAs) to obtain the Pareto front solutions.

[173] incorporates EVs into the picture as energy storage units, and solves the scheduling problem through distributed optimization by parallel programming using alternating direction method of multipliers (ADMMs) based on electricity pricing.

[174] employs a parallel autonomous optimization scheme where each user requires the knowledge of the aggregate load of other users. A soft constraint on the user load schedule change between two consecutive iteration help the parallelization of the optimization problem.

Authors in [175] propose a demand-side bidding approach for rational but selfish consumers to determine day-ahead electricity demand curtailment in return of favorable prices.

Authors in [176] employ a distributed subgradient method to determine optimal scheduling, which can find near-optimal solutions even in case of lost AMI messages.

Authors in [177] propose a cooperative algorithm in absence of a central coordinator that can reach optimal solution for social welfare through coordination among the participants.

Rahimiyan et al. in [178] propose an energy management algorithm that coordinates the price responsive demands of a cluster with DERs while modeling the uncertainty in DER production and energy prices through robust optimization techniques.

Authors in [108] implement a Lyapunov-based cost minimization algorithm to determine jointly optimize demand management and energy management decisions. Although the optimization algorithm is distributed in each household, it still has to reiterate with the energy optimization algorithm to reach convergence.

Authors in [179] use NSGA II optimization technique on load prediction to obtain area-load based pricing that benefits both utility and consumers.

Authors in [180] use Analytic Hierarchy Process in an incentive-based demand response program.

[181] uses a multi-objective genetic algorithm to solve the demand side management problem.

[182] proposes a model that clusters buildings into local energy communities for sharing renewable energy resources to satisfy energy demand.

Some research papers propose the architecture of a system/network that facilitates communication between multiple smart houses for coordinated demand response [183, 184].

There also exists literature which considers the coordination problem in islanded microgrid perspective [185, 186, 187] and also consider the mode transfer problem from grid-connected to islanded [188].

2.2. Knowledge Gap

This chapter presents the knowledge gaps identified from the literature review. These knowledge gaps are categorized according to objectives and contributions of this dissertation.

2.2.1. Open Source Software for Building Energy Management

As already identified in the literature review section, there exists a need for a cost-effective building automation software solution for small and medium-sized buildings, which lose energy saving and DR opportunities due to lack of inexpensive hardware/software solutions.

Although some commercial software platforms exist, there are still certain limitations for users or developers of that particular platform including proprietary nature, incompatibility between vendors, limited number of supported devices, standards, communication technologies or data exchange protocols. Even though advanced users or developers can sign up to develop applications or add new devices to certain platforms, they are limited by the need to rely on specific tools provided by that platform. This makes the development ecosystem vendor specific for that particular platform.

An open source solution can provide the platform for developers to easily join in developing DR applications or adding support for more devices. There are some existing open-source solutions. However, the solution to make devices coordinate and communicate autonomously and seamlessly together in order to achieve the global objective of a building's owner (e.g. perform DR when the price of electricity is high) has not yet been addressed.

Besides being open-source and inexpensive, the more imminent issue that a BAS has to address is interoperability. To integrate a BAS with different devices, communication technologies and protocols to offer a vendor-independent, Application Programming Interface (API) agnostic solution, it has to be designed as a robust platform that allows different APIs to be integrated into a standardized common communication medium. A design that supports this form of standardization and interoperability is of maximum interest to the current building researcher community and industry.

Other expected features of a BAS are automated plug & play of devices for customer convenience, scalability for implementation in larger buildings or even multiple buildings, opportunity to use distributed algorithms for demand response, and more importantly reliability and security. With various hardware and software components in the loop, all tied by different types of communication, Building Energy Management Systems have gained increased exposure to potential cyber-physical threats. Although, some literature exists that aim to identify and address security concerns in these systems, very few literature entries analyze the security aspects in full.

2.2.2. Learning-based Algorithms for Building Energy Efficiency

Based on the literature review, the usage of reinforcement learning algorithms for building energy management has not been explored to its full potential. Most of the existing works focus on the control of a particular type of load, and fail to address the need for a complete solution that encompasses all types of loads in the building.

Most of the works also focus on offline learning algorithms, where the optimum solution can be reached by visiting all the possible states through numerous episodes of exploration. They fail to address the applicability of the algorithms in an online environment, where random explorations cannot be made due to user inconvenience related to some of the load states. Therefore, in an environment with users, the algorithm needs to have certain policies for exploration and exploitation, in order to avoid extreme discomfort and annoyance of user. This issue is often ignored.

Besides this, how explorations can be made with the goal of maximizing energy savings is not well addressed in the literature. Some of the works propose learning-based scheduling of loads, where user requests are kept pending until revised decisions have been made. This is very inconvenient for the user.

The convergence issues and complexity of the algorithms may also deter them from being used in hardware with low computational resources, which should be a concern while designing them.

Finally, the idea of learning-based algorithms to be used in an incentive-based DR algorithm has not been properly explored.

2.2.3. Demand Response through Building Energy Management

As discussed in the literature review, there has been a good amount of work in developing energy management algorithms for demand response applications. This includes both price-based and incentive-based algorithms, although more research work focuses on price-based algorithms. This leaves some opportunity to explore incentive-based algorithms, as they offer more DR potential than their price-based counterparts. Incentive-based algorithms focus on commitment to provide load curtailments. This may interfere with customer comfort, and therefore algorithms must consider customer comfort while making load shedding decisions. Some research work addresses this problem by assigning load priorities, but very few addresses the problem using learning algorithms for automatically learning user comfort preferences. This provides another knowledge gap to formulate algorithms to dynamically control loads based on learned usage pattern of loads in different periods of the day.

Another observation from the existing literature is that most of them are simulation-based studies. Very few papers actually present results from implementation of their algorithms in real smart house environments. This creates the opportunity for more case studies in smart house environments with DR algorithms. If DER resources are used, there are even more unanswered questions. For example, how DER can be integrated with an incentive-based DR program; how storage can be managed to provide energy during peak load hours; or how much benefit can DER provide to reduce both peak demand and demand restrikes – these are all unanswered questions.

The ability to make DR decisions in a coordinated fashion among a group/community of buildings has gained lot of interest in recent years. The literature review suggests that researchers have explored various optimization algorithms to address this issue.

Many address this problem with a centralized algorithm with inputs from multiple buildings. As opposed to distributed algorithm, this approach is less desirable due to increased complexity and dependence on a centralized entity. This prompted many researchers to try distributed algorithms. Many of the proposed distributed algorithms are in essence centralized algorithms broken into pieces for parallelization, but requires serial processing or iterative convergence with central algorithm at some point. Some can function without the central entity, but suffer from convergence issues or increased processing times.

Many of the existing approaches implement game theoretic methods for distributed and autonomous decision making of buildings by communicating and reaching a common optimal solution. These approaches are superior to others as they treat each building as a separate player with its own selfish goals, as is the case for real situations. They can act cooperatively or competitively for reaching personal DR objectives and global DR objectives. Very few of these game theoretic approaches have been implemented in multi-agent environment with autonomous decision-making agents at each building.

Most of the multi-agent based work only consider two layers of agents, one at customer level and one at community level. The third layer, which includes the devices is often ignored. Implementing multi-agent system at device level offers a deeper distributed approach by segmenting the complexity of decision-making process into device agents. This can expedite the decision making algorithm by utilizing more computing resources. Another benefit is the option of using machine learning algorithms at each device agent for usage pattern analysis. This approach has not been explored and constitutes parts of this dissertation.

Almost all of the research works in this area are simulation based. The only way to determine if a proposed coordinated algorithm will be feasible to implement in a real environment is to experiment with real devices and deployable embedded systems. Many of the works in literature show promising results in convergence issues and optimal DR solutions, but it has not been tested if their computational complexity can be managed by devices with limited resources.

Another knowledge gap resides in the treatment of residential and commercial buildings separately in literature. A set of algorithms is required that can take into account the differences in commercial and residential buildings while considering both in the DR scenario. This has not been addressed well in existing literature.

3. Development of a Building Energy Management Open Source Software (BEMOSS™)

3.1. Background of BEMOSS™

Developed at the Virginia Tech Advanced Research Institute, Building Energy Management Open Source Software (BEMOSS™) [189] is an operating system that is engineered to improve sensing and control of equipment in small and medium-sized commercial buildings. BEMOSS™ (www.bemoss.org) aims to optimize electricity usage to reduce energy consumption and help implement demand response (DR) programs. This opens up demand side ancillary services markets and creates opportunities for building owner/operators. This in turn can help accelerate development of market-ready products like embedded Building Energy Management (BEM) systems and device controllers for HVAC, lighting and plug loads. BEMOSS™ aims to offer: scalability, robustness, plug and play, open protocol, interoperability, cost-effectiveness, as well as local and remote monitoring, allowing it to work with load control devices form different manufacturers that operate on different communication technologies and protocols. Currently, BEMOSS™ supports the following prevalent communication technologies: Ethernet (IEEE 802.3), Serial (RS-485), ZigBee (IEEE 802.15.4) and Wi-Fi (IEEE 802.11); and protocols: BACnet, Modbus, Web, ZigBee API, OpenADR and Smart Energy Profile (SEP) protocols. The scope of this dissertation includes design of auto-discovery and plug & play, interoperability, DR signal reception, multi-building communication and security features of BEMOSS™. This chapter will initially go through the basic concepts of BEMOSS™ and the software architecture and then focus on parts that have been developed as part of this dissertation.

3.1.1. BEMOSS™ Key Features

BEMOSS™ offers the following key features:

Open source architecture – BEMOSS™ is an open source operating system that is built upon VOLTTRON™ – a distributed agent platform developed by Pacific Northwest National Laboratory (PNNL). BEMOSS™ is designed to have an open architecture to make it easy for hardware manufacturers to seamlessly interface their devices with BEMOSS™. Software developers can also contribute to adding additional BEMOSS™ functionalities and applications.

Plug & play – BEMOSS™ is designed to automatically discover supported load controllers in commercial buildings. Once the device is discovered, BEMOSS™ can then monitor and control the specific device for the desired function.

Interoperability – BEMOSS™ is designed to work with load control devices form different manufacturers that operate on different communication technologies and data exchange protocols. Currently, BEMOSS™ supports the following prevalent communication technologies: Ethernet (IEEE 802.3), Serial (RS-485), ZigBee (IEEE 802.15.4) and Wi-Fi (IEEE 802.11); and protocols: BACnet, Modbus, Web, ZigBee API, OpenADR and Smart Energy (SE) protocols.

Cost effectiveness – Implementation of BEMOSS™ is deemed to be cost-effective as BEMOSS™ is built upon a robust open source platform that can operate on a low-cost single-board computer. This feature can contribute to its rapid deployment in small or medium-sized commercial buildings.

Scalability and ease of deployment – The BEMOSS™ operating system is designed with a multi-layer architecture where multiple single-board computers hosting the BEMOSS™ operating system can communicate among each other and with a master controller to monitor and control a large number of load controllers in a multi-floor and high occupancy building.

Ability to provide local and remote monitoring – BEMOSS™ provides both local and remote monitoring ability with role-based access control.

3.1.2. Target Buildings and Target Load Types

BEMOSS™ mainly targets small- (<5,000 sqft) and medium-sized (between 5,000 and 50,000 sqft) commercial buildings. BEMOSS™ controls HVAC, lighting and selected plug loads. Load controllers with communication features for each of these target load categories are considered for BEMOSS™ platform integration. For HVAC systems, smart thermostats and VAV controllers are the most popular control devices. For lighting loads, dimmable ballasts and Wi-Fi light switches can be used. Smart plugs can be used as controllers for miscellaneous plug loads. In addition to these controllers, BEMOSS™ also supports power/energy meters and sensors (e.g., occupancy, light, temperature and humidity).

3.1.3. BEMOSS™ Architecture

3.1.3.1. System Architecture

The BEMOSS™ architecture is illustrated in Figure 3.1 for a small commercial building with a few load controllers of each type. In this architecture, only one single-board computer equipped with the BEMOSS™ is used to enable monitoring and control features of all load controllers in the building. This embedded system can communicate with different types of load controllers, i.e., thermostats, lighting load controllers and plug load controllers, and sensors/power meters via wireless signals.

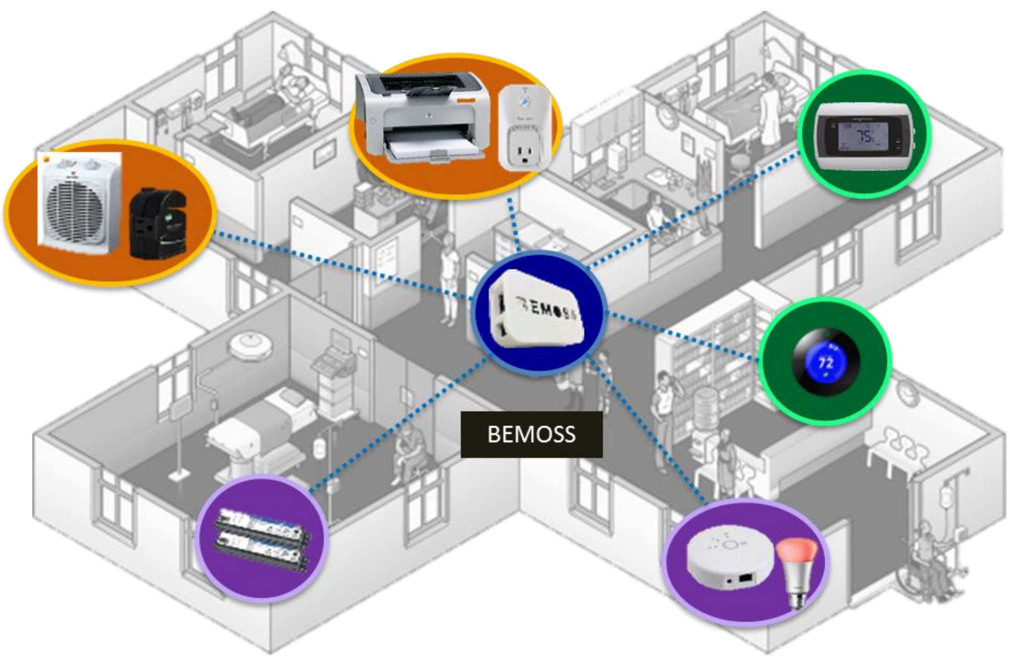


Figure 3.1. BEMOSS™ system architecture in small buildings with a few load controllers.

For larger buildings with larger number of devices, the BEMOSS™ operating system can be set up to deploy its multi-node architecture feature. In this architecture, as shown in Figure 3.2, several single-board computers (BEMOSS™ nodes) each equipped with BEMOSS™ communicate with each other and also communicate with a central single-board computer (BEMOSS™ core). Each BEMOSS™ node is responsible for monitoring and controlling a zone in which a selected set of load controllers reside, while the BEMOSS™ core is responsible for supervising the overall system operation and allow local and remote access for monitoring and control of all devices in buildings.

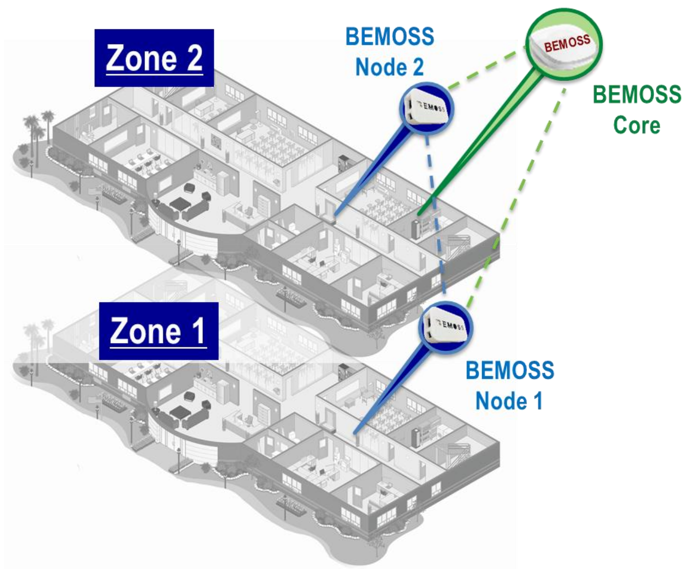


Figure 3.2. BEMOSS™ system architecture in large buildings with more load controllers.

For multiple buildings, the BEMOSS™ architecture can be expanded as shown in Figure 3.3. Here each building has one BEMOSS™ core and multiple BEMOSS™ nodes (as needed), which follow the architecture in Figure 3.2. BEMOSS™ cores in different buildings can communicate with each other to make coordinated decisions.

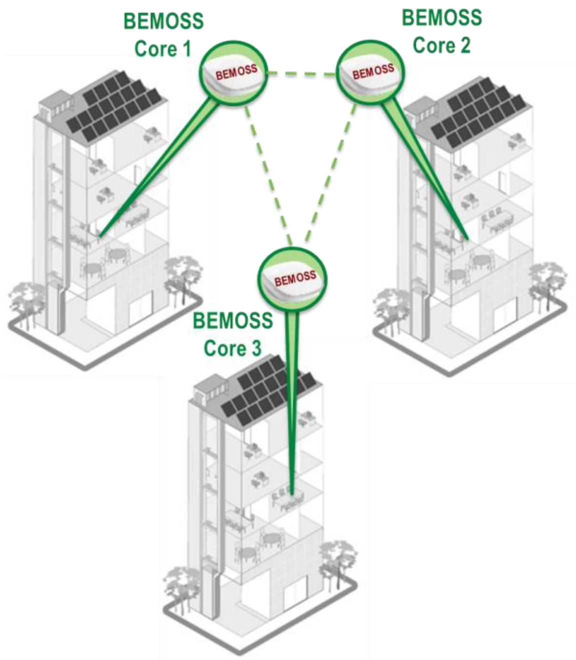


Figure 3.3. BEMOSS™ system architecture in multiple buildings.

3.1.3.2. Software Architecture

BEMOSS™ is a robust open source operating system for building energy management - built completely using open source software tools. The entire BEMOSS™ system comprises of four layers: User Interface (UI) layer, Application and Data Management Layer, Operating System and Framework layer, and the Connectivity Layer. In addition to these, there are also parallel BEMOSS™ databases that help with the storage of all information pertaining to BEMOSS™ and help smooth its functioning. Figure 3.4 depicts the overall software architecture, while highlighting the parts that are contributions from this dissertation in red.

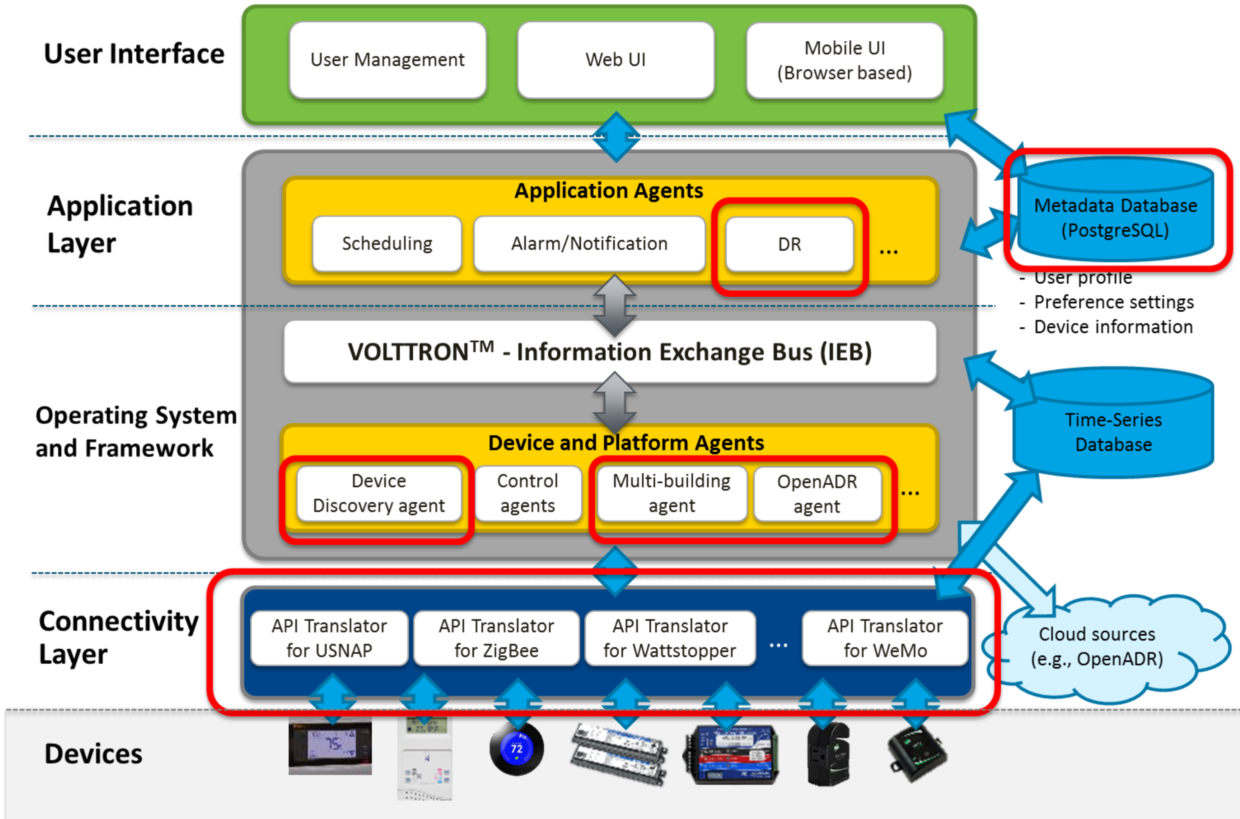


Figure 3.4. BEMOSS™ software architecture [190].

Layer 1: User Interface (UI) - The BEMOSS™ UI layer has two components: UI (i.e., web browser interface and mobile interface) and user management. BEMOSS™ web UI is a dashboard type interface with visuals and graphs to show current settings of devices in each zone. Authenticated users can also control these devices through an on-site interface. Regarding user management in BEMOSS™, role-based access control is implemented to allow different levels of access to different individuals. For example, building engineers will have full authority to adjust set points and schedules of loads in buildings, while tenants will have limited access to view current status and historical load data, or control selected loads in specific zones. In BEMOSS™, this role-based access control is achieved using access control lists.

Layer 2: Application Layer- This layer embeds algorithms to allow monitoring and control of hardware devices interfaced with BEMOSS™. Examples of possible applications include demand response, price-based management, planning and scheduling, behavior pattern analysis, load management, as well as alarm/notifications.

Layer 3: Operating System and Framework - In this layer, VOLTTRON™, a distributed agent platform developed by Pacific Northwest National Laboratory (PNNL), is chosen as the software platform for BEMOSS™. Several agents have been developed to support BEMOSS™, including device discovery agent, control agents (i.e., thermostat agent, lighting load agent and plug load agent), monitoring agent (i.e., sensor agent), multi-node agent, multi-building agent, OpenADR agent, Network and Platform Agents. There can also be DR agents for running centralized DR algorithms. Figure 3.5 illustrates this (Contributions from this dissertation are highlighted in red).

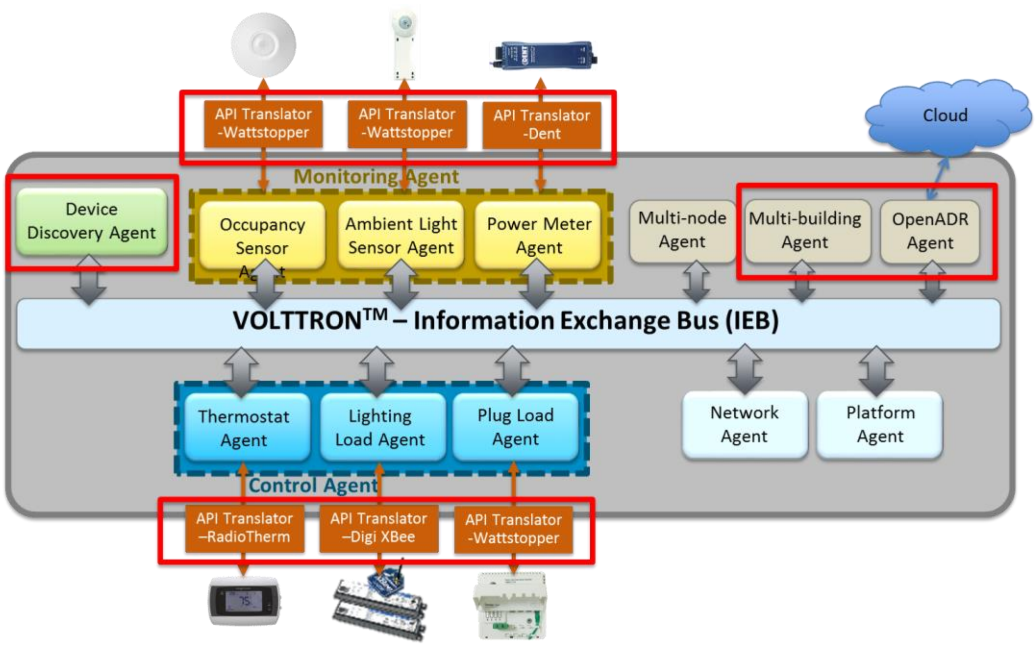


Figure 3.5. Detailed software architecture at BEMOSS™ OS and framework layer [190].

The different agents in BEMOSS™ are introduced below:

- **Device discovery agent** – is responsible for detecting the presence of devices in a building, querying their model numbers, identifying their APIs and launching a control agent to monitor/control the discovered device.
- **Control agent** – includes thermostat agents, lighting load agents and plug load agents. These agents are automatically generated to monitor, communicate and control hardware devices after they were discovered by the device discovery agent.

- **Monitoring agent** – is a sensor agent that communicates with sensors and/or power meters. Similar to control agents, monitoring agents are automatically generated to monitor and communicate with sensors and power meters after they are discovered by the device discovery agent.
- **Multi-node & Multi-building agent –** are responsible for handling the communication between multiple devices hosting BEMOSS™. Multi-node agent assists in communication between BEMOSS™ Core and BEMOSS™ nodes and among BEMOSS™ nodes. Multi-building agent assists in communication among BEMOSS™ cores in multiple buildings.
- **OpenADR agent –** is responsible for receiving OpenADR signal from a DR aggregator and interpret and send the DR information to other agents.
- **Network agent –** is used to discover other BEMOSS™ devices (other nodes or cores) and check the network status between multiple BEMOSS™ devices.
- **Platform agent –** is responsible for checking the status of the platform and other agents and facilitate platform/agent restarting and error logging in case an unexpected failure occurs.

All agents communicate over an Information Exchange Bus (IEB).

Layer 4: Connectivity Layer - This layer takes care of the communication between the Operating System and Framework layer and all physical hardware devices. To allow BEMOSS™ to communicate with hardware devices that use different communication technologies, data exchange protocols and have device functionalities (different device API's), the BEMOSS™ includes API Translators. Each API translator allows BEMOSS™ agents to communicate with a group of devices based on their unique API's. Basically, API translators provide a translation service for BEMOSS™ agents so that agents can get readings and send control commands to devices (without knowing their API's) using simple function calls: getDeviceStatus and setDeviceStatus.

Databases: BEMOSS™ has a time-series database for storing time-series data. A relational database management system (PostgreSQL) is used to satisfy the need to store the metadata for identifying users, devices, and process controls.

The following sub-chapters will discuss the parts of the BEMOSS™ system developed within the scope of this dissertation.

3.2. Development of Auto-Discovery & Plug & Play

Auto-discovery and plug & play features of BEMOSS™ are implemented through the device discovery agent. This dissertation contributes the design and implementation of the device discovery agent to BEMOSS™, including: the agent modeling; how the agent connects to the discovery APIs; the basic discovery APIs for WiFi, ZigBee, ZigBee SEP, BACnet and Modbus; and the behavior of the device discovery agent. Following the basic discovery agent process proposed in this work, the plug & play features for newer devices can be easily integrated by adding appropriate discovery APIs to the software. This is an on-going work at Advanced Research Institute, and the current software version includes auto-discovery of more devices than this dissertation covers.

3.2.1. Device Discovery Agent

Device Discovery agent is one of the first few agents that are launched when BEMOSS™ is started. As its name suggests, its primary purpose is to discover devices in the building that are compatible with BEMOSS™. The discovery process for any device can be divided into four steps:

1. Detect the presence of a device in the building
2. Query the device to find out its model information
3. Look up the device in the list of supported devices and check if API translator is available
4. Initiate a control agent with the relevant API translator for the device

Besides these four steps, discovery process also needs to supply relevant information about the device to the dashboard page at UI. It also needs to keep track of already discovered devices to avoid duplicate instance of agents for the same device.

To make this all possible, the discovery agent uses a database to keep record of the discovered devices. This database named bemossdb is a PostgreSQL database for storing BEMOSS™ metadata. The same database is checked by the UI to receive device discovery information for the dashboard page. Also, as the dashboard page needs to show primary device data (like temperature, setpoint and mode for thermostats) with each discovered device, the discovery agent also needs to collect these data at the time of first-time discovery of a device. Hence, a more complete discovery agent takes up the following responsibilities:

1. Detect the presence of a device in the building
2. Query the device to find out its model information and unique device ID (MAC address/UUID)
3. Check if the device unique ID already exists in bemossdb to avoid duplicate device entries
4. For a newly discovered device, look up the device in the list of supported devices and check if API translator is available
5. Create an unique device ID for the device to be recognized within BEMOSS™
6. Make an entry for the newly discovered device in bemossdb and fill its information
7. Initiate a control agent with the relevant API translator for the device
8. Keep track of already discovered device's status (approved or non-BEMOSS™ device)

3.2.2. Device Discovery Agent Knowledge Modeling

Device discovery agent uses two tables in the bemossdb database for the auto-discovery process. The first one is the supported_device table which facilitates the directory service by containing the list of supported devices, their discovery protocols and their corresponding API translators. The device discovery agent first looks through the table to find all the supported discovery protocols, which will be named as Discovery APIs. It then uses each discovery API to find devices that respond to that discovery API. The same API is used to find the device's MAC address. The other table that device discovery agent uses is the device_info table which contains the metadata for all discovered devices. Device discovery agent populates this table after each new device has been discovered. It also uses this table to check if a device has already been discovered.

Table 3-1 lists the metadata for each discovered device that discovery agent stores in the 'device_info' table in the PostgreSQL database.

Table 3-1. Metadata Stored in ‘device_info’ Table

Attribute	Data Type
Device ID	string
Device Type	string
Vendor Name	string
Device Model	string
Device Model ID	string
MAC Address	string
Min Range	string
Max Range	string
Identifiable	boolean
Date Added	date
Approval Status	string

3.2.3. Device Discovery APIs

The discovery process varies from one device type to another. The process depends on types of communication technologies/data exchange protocols being used (i.e., WiFi, ZigBee, ZigBee SEP, BACnet, Modbus etc.). Each of these processes are put within one Discovery API which takes care of the details of the discovery protocols and offers generic named methods like ‘discover’ or ‘getMACaddress’. This helps the device discovery agent to be protocol agnostic while looking for new devices. The same concept is applied in API translators. This is shown in Figure 3.6.

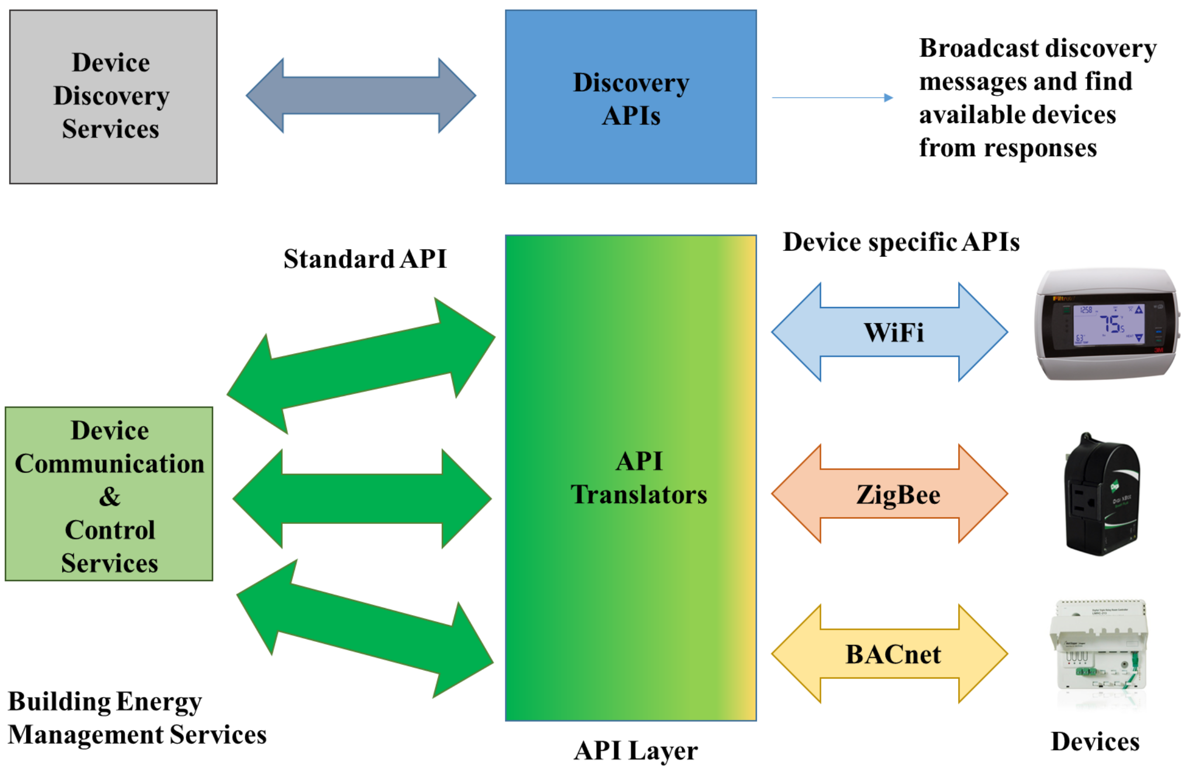


Figure 3.6. Concept of discovery APIs and API translators.

The following are the different discovery APIs in BEMOSS™, developed as part of this dissertation:

WiFi: As most Wi-Fi devices support Simple Service Discovery Protocol (SSDP) to allow Universal Plug n Play (UPnP), the discovery agent discovers these devices using SSDP. In discover method of the WiFi discovery API, a discovery message to the multicast address 239.255.255.250 at port 1900. The responses to this message are collected and their IP address are saved as the detected devices.

ZigBee: As ZigBee is asynchronous serial communications within a mesh network, all requests/commands are essentially frames of bytes, which are sent byte by byte serially through a ZigBee coordinator. These frames are received by all ZigBee devices within the mesh network, and the device that the frame is directed to, replies to the command. The communication can be point-to-point, or point-to-multipoint.

For ZigBee device discovery, the discovery message frame is broadcasted to all devices within the ZigBee network. The message frame can be custom designed by the ZigBee protocol used in the device.

ZigBee SEP: ZigBee SEP offers extended security for communications between the coordinator and devices by providing secured key exchange between them. This key exchange can be performed between approved devices only, and hence each ZigBee SEP device needs to be pre-configured to be added to the coordinator's list of approved devices. This can be done by adding the device manually to the coordinator's device table using the MAC ID and installation key of the device. All added devices remain in the device table of the coordinator, so it is actually easier to detect ZigBee SEP devices by looking at the coordinator's device table.

BACnet: BACnet devices can be discovered using the BACnet protocol. In this protocol, the BACnet server sends 'who-is' broadcast messages to the devices within the network. The BACnet devices then respond with 'I-am' messages with their device IDs.

In BACnet discovery API, a virtual BACnet server initiates the discovery broadcast. It then waits and collects the 'I-am' responses from all the devices that reply.

- For BACnet IP devices, the response also includes IP addresses of the devices.
- For BACnet MSTP devices, they are connected to a BACnet IP to an MSTP router. The router acts as a BACnet Broadcast Management Device (BBMD) and re-iterates the broadcast within MSTP network. It then collects the responses from MSTP devices and sends them over IP back to BACnet server. Therefore, for the MSTP devices, they are detected with the common IP of the router they are connected to, as well as the unique device IDs. These device IDs are used to direct any command/request to a specific device.

Modbus: Modbus devices do not offer any broadcast-response mechanism. Hence for Modbus devices, discovery works by polling each possible IP address in the network by sending Modbus queries and looking for valid Modbus responses. By sweeping through all the IPs in the network and collecting responses, the IPs of supported Modbus devices can be detected.

3.2.4. Device Discovery Agent Behavior

The discovery agent can run its discovery cycle either periodically or by manual request by user through UI. In both cases, the discovery agent goes to its whole process of device detection, identification, database checking, data collection, database update, and control agent initiation for each type of supported devices. One discovery cycle can be summarized in the flowchart in Figure 3.7.

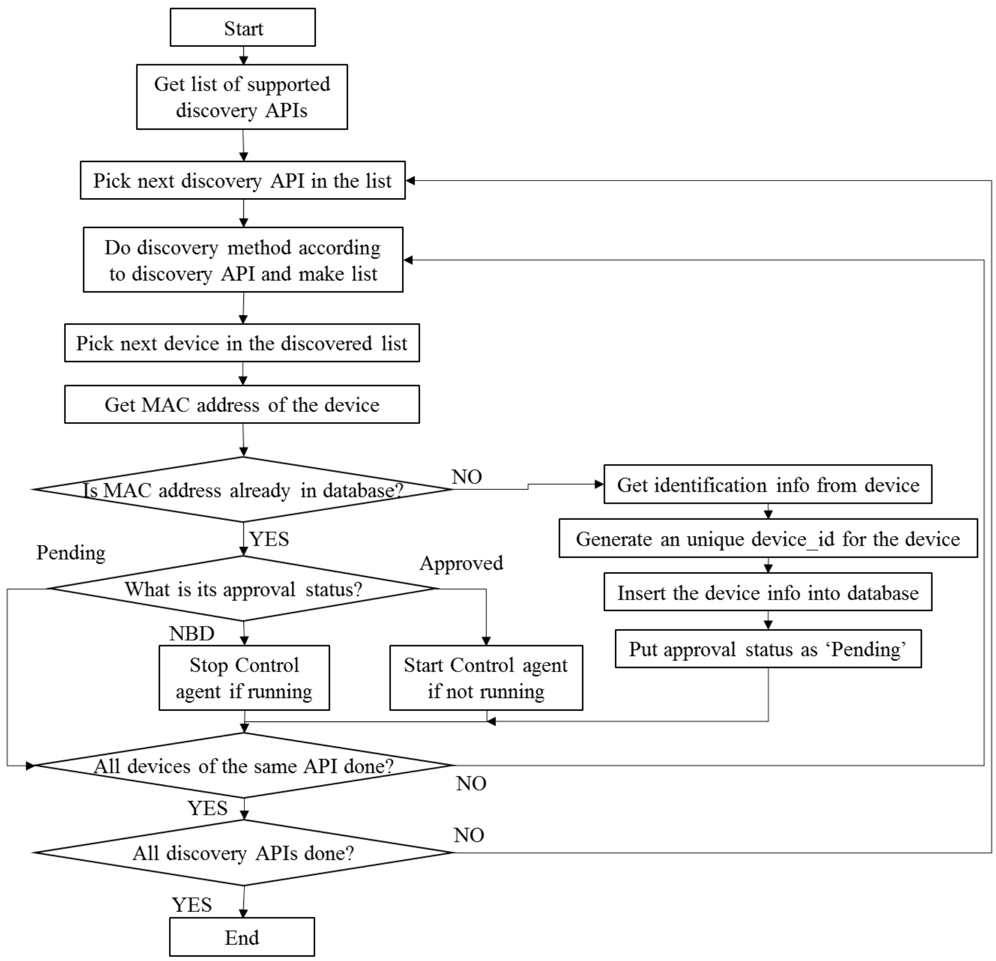


Figure 3.7. Flow chart for a discovery cycle.

As can be seen in Figure 3.7, the discovery cycle starts with the list of supported discovery APIs and follows discovery process for each API.

For each discovery API, the discovery agent detects the presence of devices of the same type by following its broadcast and detection procedure. Once the devices of the same type have been detected, the agent queries each of them for their MAC address or UUID, if it has not already been received with the detection method. Now, the agent can check if the detected MAC address or a UUID exists in the database or not. If a MAC address exists in the database, it means the device was previously discovered. It then checks for its approval status. If the approval status is ‘Pending’ then user has not made any decision on the device yet, so agent continues with next device. If the approval status is ‘Approved’, then the agent starts the control agent if it is not already running. On the other hand, if the approval status is ‘NBD’ (Non-BEMOSS™ Device), then it means the device has been manually marked by the user as a non-BEMOSS™ device, so the agent stops any control agent that is running for that device.

If the detected device’s MAC address/UUID does not exist in the database, then it means it is a newly discovered device. For such a device, the agent queries the device for device model information. It then looks up the list of supported devices to see if the device is supported and API translator is available. Then, it creates a unique ID for the device (containing device type and a number) and prepares configuration file for control agent. Finally, the agent creates a new entry in the database with this unique ID and fills up device_info table for the device with ‘Pending’ approval status. The process is then continued for all devices detected. Then discovery agent goes to next discovery API and follows the same process with all discovery APIs. An approval helper agent assists discovery agent by starting control agents with API translators found in the configuration file once it has been approved and an approval message is sent from UI to IEB.

3.3. Development of API Translators for Interoperability

The connectivity layer of BEMOSS™ is responsible for handling the communication between BEMOSS™ and devices. As different devices follow different communication protocols, and even within same protocol, they may follow different APIs, the interoperability is achieved by means of API translators, which work as mediators between devices with different APIs and BEMOSS™. This dissertation contributes the basic structure of API translators to BEMOSS™ as well as API translators for different devices. The structure can be followed to write new API translators to allow new devices to work with BEMOSS™. This is an on-going work at Advanced Research Institute, and the current software version includes more API translators than this dissertation covers. This section discusses the considerations in connectivity layer first and then delves into the design of API translators. The work also proposes a method for automated API translator generation for common standard protocols like BACnet and Modbus.

3.3.1. Connectivity Layer Considerations

The connectivity layer in BEMOSS™ software architecture takes care of the communication between the Operating System and Framework layer and all physical hardware devices. This encompasses different communication technologies, data exchange protocols and device functionalities. For this layer to work properly, physical communication mediums should be properly setup and configured.

3.3.1.1. Communication Setup

To allow BEMOSS™ to interface with Wi-Fi, ZigBee, Ethernet and Serial (RS-485) devices, certain physical communication mediums must be set up:

- To allow BEMOSS™ to communicate with *Wi-Fi* devices, a Wireless Local Area Network (WLAN) can be setup using existing wireless infrastructure in a building or by using wireless routers to setup a new wireless network. Both Wi-Fi hardware devices and BEMOSS™ must be configured to join the same network.

- To allow BEMOSS™ to communicate with *ZigBee* devices, BEMOSS™ must be connected with a ZigBee coordinator to allow a personal area network (PAN) to be formed. Then, all ZigBee devices must be configured to join associated PAN(s).

- To allow BEMOSS™ to communicate with *Ethernet* devices, all Ethernet devices are connected via Ethernet cables and switches. Then, Ethernet devices must be configured such that they are in the same subnet with BEMOSS™. Routers can be used to route messages between devices in different subnets.

- To allow BEMOSS™ to communicate with *Serial (RS-485)* devices, these devices are connected using physical wires. These devices may also be interfaced with BEMOSS™ via a router (for example, a BACnet IP to MS/TP router).

Once physical communication mediums are setup and properly configured, different layers of communication technologies can take care of message transmit/receive functions over physical mediums. Hence, the connectivity layer of BEMOSS™ only needs to take care of data exchange protocols and message interpretation functions.

3.3.1.2. Data Exchange and Message Interpretation

BEMOSS™ supports HTTP, HTTPS, BACnet, Modbus, SEP, ZigBee API and web service-based data exchange protocols (e.g., XML, JSON). The connectivity layer:

- Handles details of data exchange protocols based on the API of hardware devices,
- Provide a simplified and standardized API for the upper layer of BEMOSS™ based on required device functionalities, and
- Interpret messages between these two APIs.

API translator is developed in the Connectivity layer for each compatible device, which simplifies the task of agent developers by providing abstraction over details of communication technologies and protocols.

3.3.1.3. Hardware Device Functionality

BEMOSS™ primarily targets three types of hardware devices: HVAC, lighting and plug load controllers. These devices have different functional requirements. For example, a thermostat is required to provide room temperature data or temperature set point control, whereas a plug load controller is required to provide ON/OFF control. This is why the API translator also needs to address differences in required functionalities of devices in addition to their communication technologies and protocols.

3.3.2. API Translators

Each hardware device with an API that is supported in BEMOSS™ has an API translator in the connectivity layer of the BEMOSS™. API translator includes codes to utilize the API of the device to communicate with the device. It handles all details of the device specific communications, and offers simple methods for the control agent to get data or send commands to the device.

Each API translator maintains a list of attributes and capabilities.

- **Attributes** are the parameters that can be extracted from the device. These include data or settings of the device. For example, temperature, mode of operation, set point, etc. can be attributes for the API translator of a thermostat.
- **Capabilities** can be used by the agent using API translator. These capabilities are in the form of methods that can be called by an agent to perform operations on the device.

API translators of most BEMOSS™ devices offer the following methods:

- **getDeviceStatus** sends data request to a device following its API, receives and parses the response data (readings) and provides the data to the control agent. Device data vary from one device type to another. For example, a thermostat can give data, like current temperature, operating mode, set point, fan speed/mode, humidity, etc. A smart plug can give data, like current status (ON/OFF), power consumption, etc. A power meter can give detailed data, including voltage, current, power factor, real and apparent power, energy consumption, line and phase data for three phase systems, etc. The control agent may call getDeviceStatus method to receive these data in corresponding variables.
- **setDeviceStatus** is used to send control commands to the device. This control command may include any status change requests made to the device, i.e., changing temperature set point in a thermostat or ON/OFF status in a smart plug.
- **identifyDevice** helps facilitate identifying function in BEMOSS™ UI. If an identify command is issued for any device in the BEMOSS™ UI, this method is called by the device control agent. Depending on the device, this method sends some control signals to the device to make identifiable changes in the device (like backlight flashing, indicator light

flashing, status changes, etc.) This helps to relate devices appeared on BEMOSS™ UI with actual hardware devices through inspection of visual changes made.

Out of these methods, the getDeviceStatus method can be periodic. This means it can query and receive device data periodically. This ensures that the data are updated.

A control agent calls these methods as necessary to get data or send control commands to hardware device. An API developer can write codes within these methods pertaining to the hardware device API. These codes inside API translator methods will be different for different devices based on their APIs, but the API translator structure remains the same for all devices. This ensures that the code remains generic for generic control agents that use these API translators.

3.3.3. Automated API Translator Generation

This developed feature of BEMOSS™ aims to simplify API Translator codes and develop generic codes for devices that follow similar communication technology and protocol. The idea is to create one Auto API Translator generator code for one set of devices all of which follow same or similar communication protocol. This is especially useful for devices that follow standardized protocols like BACnet/Modbus, and the APIs of those devices only differ in names and register numbers of the data/control variable provided by the device. The Auto API Translator code takes care of the details of the protocol and provides a generic code to extract information from an excel file. The details of the read/write variables of a particular device can be provided in that excel file. Therefore the same code can be used for all the devices that follow the same protocol, using one excel file per device which contains the device specific information.

Example: Figure 3.8 shows a sample excel file for BACnet Auto-API Translator code. The use of different fields of the excel file are denoted in the figure.

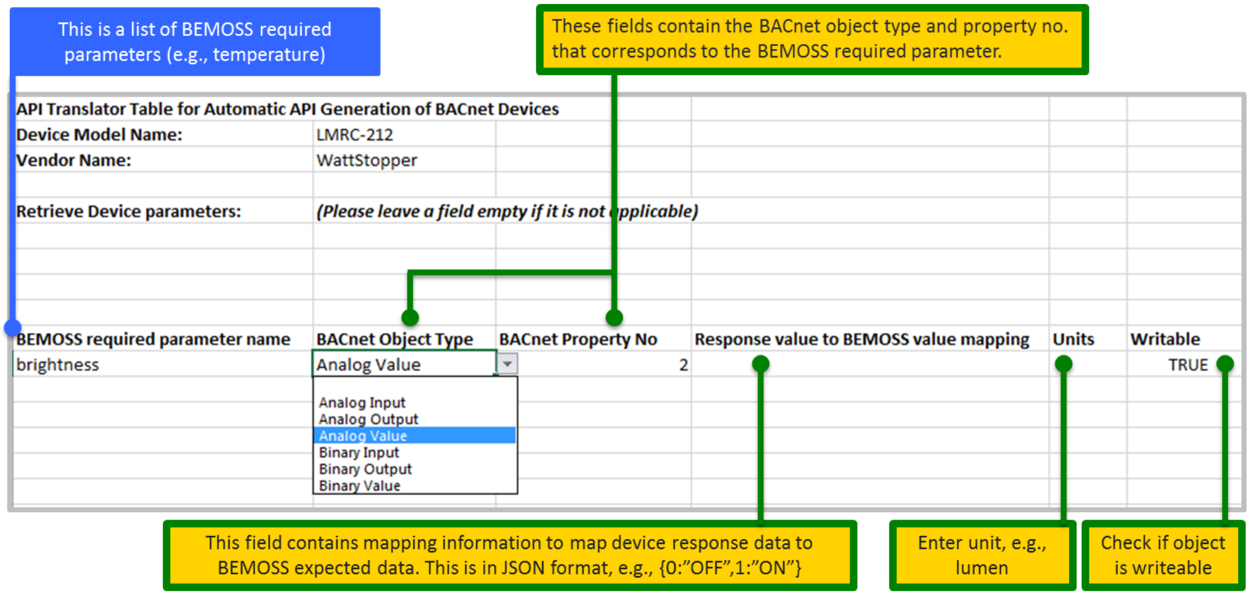


Figure 3.8. Sample excel file for BACnet auto-API translator code [191].

An Auto API Translator code for BACnet devices can look into the excel file for a BACnet thermostat EXL1610 and find out the exact object property numbers for the necessary variables for the thermostat (i.e., temperature, setpoints) from that excel file. It can then act as the API Translator for the BACnet thermostat. The same code can be used for a Wattstopper BACnet lighting device using another excel file for it.

Figure 3.9 shows the flow chart for Auto-API Translator code for getDeviceStatus method.

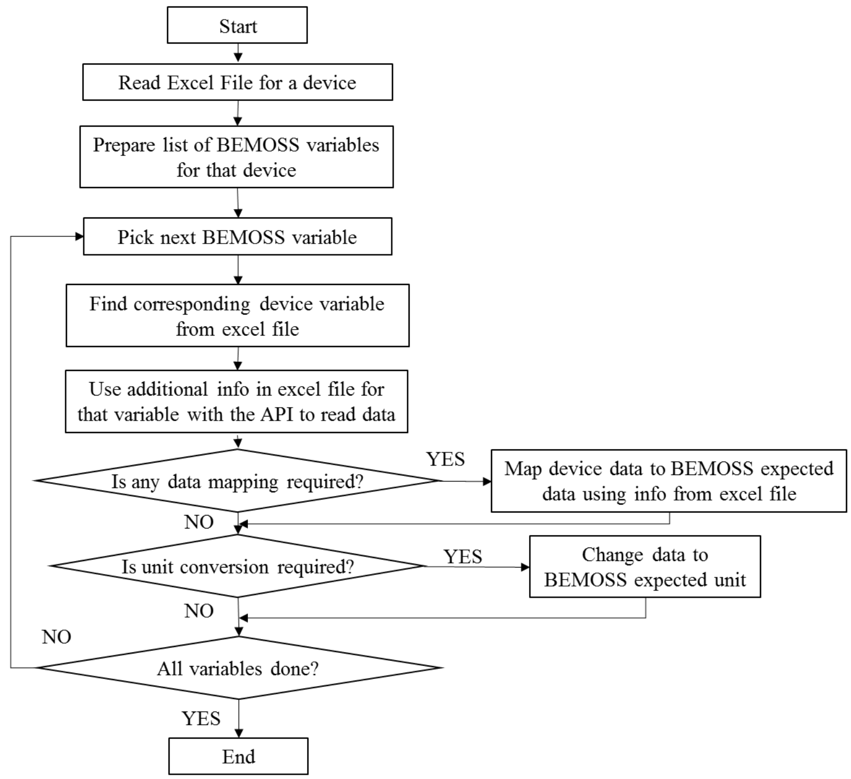


Figure 3.9. Flow chart of auto-API translator code for reading device data.

The benefits of using Auto API Translator codes are:

1) Only one code is needed for all the devices that use the same protocol. This reduces development complexity and time that is required to write individual API Translator code for each device.

2) A new device that follows a supported protocol can be easily added to BEMOSS™ by asking the device vendor to fill in the excel file formatted to work with the Auto API Translator for that protocol.

Auto API Translators have been developed in BEMOSS™ for BACnet devices, Modbus devices, and WiFi devices following JSON/XML data exchange formats.

3.4. BEMOSS™ Integration with OpenADR for DR Signal Reception

This work contributes to BEMOSS™ by integrating OpenADR. BEMOSS™ can receive OpenADR signals from a simulated OpenADR virtual top node (VTN) server and implement device control decisions based on the signal. An OpenADR agent in BEMOSS™ receives and interprets the signal, and sends the information to a DR agent in BEMOSS™. The DR agent then makes decisions based on a DR algorithm and sends the control commands to the corresponding device control agents. An alternative implementation is also possible where the OpenADR agent sends the information directly to all control agents and then they coordinate among themselves to make DR decisions. The setup is shown in Figure 3.10.

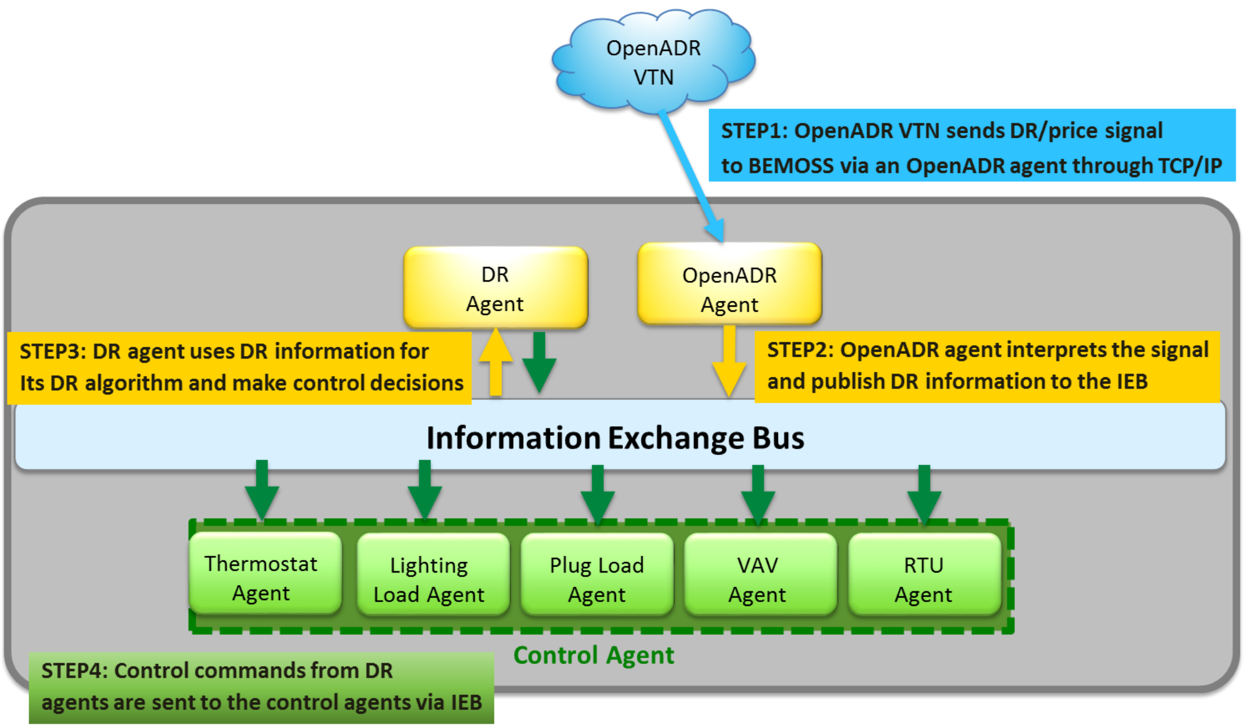


Figure 3.10. Method of enabling OpenADR signal and DR algorithms in BEMOSS™ [191].

3.5. BEMOSS™ Setup for Multi-building Communication

For multi-building coordinated DR algorithms to be implemented, the BEMOSS™ Cores in those buildings communicate among themselves through the multi-building agent. This work contributes to BEMOSS™ the communication methodology between multiple buildings using the existing multi-building agent, so that they can coordinate among themselves to reach a coordinated DR goal.

The multi-building agents can send TCP messages to each other through CurveZMQ. This can be used to share any form of data required for distributed DR algorithms. The BEMOSS™ core in each building has a OpenADR agent that can receive OpenADR signals from a DR aggregator. If DR in multiple buildings are to be coordinated, these buildings must be under one management control like on a university campus, or through an aggregator. The DR signal will be received at a central location managed by the campus operator, or the aggregator. They will then access each building's DR features and provide the most optimum control to meet the DR target.

Then the buildings communicate using the multi-building agent to share data required for running distributed algorithms. The complete data set from all buildings is then transferred by multi-building agent to the distributed DR agent, which runs the distributed DR algorithm. The algorithm decisions are then sent to control agents via IEB. It is also possible to incorporate learning algorithms in control agents so that they can also participate in the distributed DR algorithm for coordinated decision making. Figure 3.11 shows a multi-building setup between two buildings, which is similar for the case with more than two buildings.

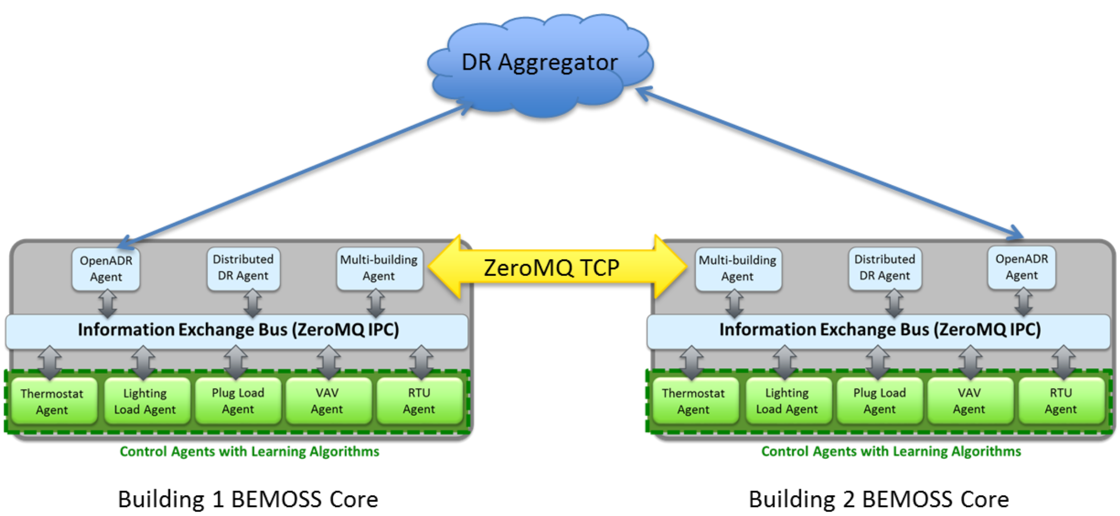


Figure 3.11. Setup for multi-building communication and coordinated DR between two buildings.

3.6. Security of BEMOSS™

3.6.1. Security of BEMOSS™ Operating System and Framework Layer

The operating system and framework is an essential layer of the BEMOSS™ as it links user interface layer, application and data management layer, and connectivity layer together. In this layer, VOLTTRON™ is used as an agent platform on which BEMOSS™ functionalities operate.

3.6.1.1. Security Concerns

Possible security risks in this layer include:

- An attacker may execute a malicious agent in order to sniff data from agents' communication messages over an Information Exchange Bus (IEB). The data can include important information such as user profile, user preference, or occupancy leading to the user privacy concerns.
- An attacker may execute an unverified agent to control or tamper with hardware devices without authorization by a user or a corresponding agent. This can result in shorter life of hardware devices, hardware devices malfunction or permanent damaged hardware devices.
- An attacker can bring in a tampered device that is supported by BEMOSS™, and once the device is discovered and added into BEMOSS™, the attacker can use that device to eavesdrop to the communication protocol between BEMOSS™ and devices, which may cause vulnerability towards communication security and data privacy.
- An attacker can bring in a malicious BEM that discovers and controls the devices in the building. An attacker can also bring in a malicious BEMOSS™ node and eavesdrop to the communication between BEMOSS™ core and nodes.

3.6.1.2. Security Measures Built in VOLTTRON™

BEMOSS™ relies partly on the following dedicated modules provided by VOLTTRON™ to address its confidentiality, integrity, and availability requirements:

(a) Authentication and Authorization (AA) module. The primary function of the AA module is to provide cryptographic integrity and authentication services to other modules in the VOLTTRON™ platform. The AA module performs this task by using public key cryptography (specifically X509v3 certificates). This module provides validation of agent payloads, authenticate peer platforms, and handle public and private credentials; address integrity while providing the infrastructure for confidentiality; and perform cryptographic authentication, authorization, and trust functions including asymmetric cryptography used for identification and integrity

(b) Directory Service (DS) module. The DS module provides name, resource, and public credential to location and network identity mappings.

(c) Resource Management (RM) module. The RM module is the gatekeeper for the platform. It decides if the platform has enough resources left to accept the execution of an agent. It manages access controls for AEE containers, and monitors use of resources and either warns or terminates misbehaving agents.

(d) Communication services (CS) module. The CS module is responsible for reliable and secure transfer of packaged agents and peer-to-peer communication between VOLTTRON™ platforms.

3.6.1.3. Security Features Developed within Scope of the Dissertation

3.6.1.3.1. BEMOSS™ Discovery Approval Process

BEMOSS™ discovery agent and UI have been designed to incorporate approval process for new device discovery to enhance security of the system. In this approval process, new devices are added to BEMOSS™ after approval from the user. It is to be noted that this process is applied only once during the lifetime of the device in the system, unless the device is lost from the database due to a software glitch or something similar.

The way this has been implemented is: instead of immediately launching an agent for a newly discovered device and adding it to BEMOSS™, in the modified system, the discovery agent keeps the status of a device pending after discovering it. It only adds an entry to the BEMOSS™ database with the discovery info of the newly discovered devices with its status as ‘pending’, without launching any agent for the device. The UI picks up entries in the database with ‘pending’ status and shows it in UI as newly discovered devices with approve/decline button beside it. This allows the user to check the device and then approve it based on a manual authentication protocol. Whether the user approves the device or marks it as a non-BEMOSS™ device (NBD), the UI makes the corresponding change to the status of the device in the database and also sends a message to the discovery agent notifying it about the status change. When discovery agent receives such notification from UI with the ‘approval’ status, it adds it to BEMOSS™ device data tables and launches agent for the device, thereby adding it to BEMOSS™. If the device is marked as NBD, discovery agent ignores the device and makes no further communication with the device.

This approval process provides the following security benefits:

1. It enhances BEMOSS™ security by adding an additional layer of authentication for devices before adding to BEMOSS™. This reduces the possibility of a security-compromised device to be added to BEMOSS™, as it has to go through the user’s approval first.

2. It prevents the BEMOSS™ from controlling other people’s devices, which are not part of a user’s building management. For example, BEMOSS™ may discover devices on a floor that is not part of the user’s system. This approval process ensures that BEMOSS™ does not control third party device on the same premises.

3.6.1.3.2. Encrypted Node-to-node Communication

This feature employs a public-private key combination for node-to-node communication. In this scheme, the BEMOSS™ core and each BEMOSS™ node have a unique public and private key pair which can be used to decrypt messages encrypted by the private key using the public key and vice versa. The public keys of all nodes are known to the BEMOSS™ core, and the public key of the BEMOSS™ core is known to all BEMOSS™ nodes. As messages between the core and nodes are encrypted and can only be decrypted by the correct key pair, it enhances the security of

BEMOSS™ by ensuring that a tampered and unauthorized BEMOSS™ node cannot take part in communication within BEMOSS™ network.

3.6.1.3.3. BEMOSS™ Malicious Discovery Detection

This feature is realized by a detection agent which monitors the network traffic and identifies the IPs of the machines that initiate discovery broadcast messages for BEMOSS™ devices. If the IP is not an approved IP, then the discovery messages are not authorized and therefore the detection agent notifies the BEMOSS™ operator about malicious device discovery attempts from the unauthorized IP. The operator can then block that IP from the BEMOSS™ network and prevent unauthorized discovery and control of devices in the building.

3.6.2. Security of BEMOSS™ Connectivity Layer

As agents communicate with different hardware devices using the connectivity layer and different communication mediums, security aspects of this layer are mainly dependent on the security of the message exchanges between the device and the agent.

3.6.2.1. Security Concerns

The most probable ways of attack to this layer are:

- An attacker may eavesdrop to the messages between the agent and the device. If they are not encrypted, the attacker may gain information about device status and control commands, which may give away private information such as: occupancy, user preferences etc., which may be used for malicious purposes. It also creates privacy concerns.
- An attacker can alter information in the data messages, thereby causing the agents to receive corrupted device data and make undesirable decisions.
- An attacker can send spurious control messages to devices, causing nuances. This can be done from external entities or from malicious code inside agent hosts.

3.6.2.2. Security Measures

Encryption and message validation should be implemented to ensure data privacy and prevent data corruption. Authentication can prevent devices from reacting to fake control messages, but not all hardware devices support encryption/authentication procedure. Following is a description of how BEMOSS™ implements specific security features in different protocols.

- **ZigBee:** ZigBee messages can be protected by means of symmetric key cryptography. Keys can be pre-installed or transported between devices. A separate key may be used for each pair of nodes. Distribution of keys can be managed from the Zigbee coordinator, which acts as the trust center. ZigBee suites typically use AES-CCM (a combined encryption and authentication block cipher mode), which use encryption to prevent eavesdropping and message integrity codes to ensure authentication and message integrity. ZigBee SEP 2.0 compliant devices use even a more rigorous approach in this case, where devices have to be authorized in the trust center with installation codes prior to communication, which then establishes key-based communications. Although this requires manual installation of devices, it is more secured as it prevents unauthorized devices from joining the network and send/receive messages.

- **BACnet:** Addendum G to the BACnet standard specifies requirements for security of BACnet IP or MSTP network. If this standard is followed, authentication and encryption are ensured to prevent eavesdropping or malicious control commands. The only issue is a lot of vendors do not include these security features in their products due to complexity and computational requirement at the device end. For BACnet MSTP devices, securing the MSTP messages with additional encryption/decryption may impact performance. However, it is also difficult to attack MSTP network, because the attacker has to gain access to actual physical RS-485 wires to launch any type of attack. Therefore, an alternate to securing MSTP network through encryption is to secure the MSTP hardware devices and wires physically through careful conduit design and making hardware devices accessible to authorized personnel only. For BACnet IP devices, key-based encryptions should be implemented. BACnet IP devices without any security feature should be connected through a separate LAN to isolate it from the main building LAN, which is more accessible to an attacker.

- **Modbus:** Modbus protocol by itself does not provide any security measures. Some vendors develop security features overlaid on top of the actual Modbus protocol to ensure security. But most devices do not offer any security by their own. In such cases, similar to BACnet, the Modbus RS-485 network and devices can be secured using physical security measures. For Modbus TCP, additional security can be overlaid on top of the actual message exchange protocol.

- **Wi-Fi:** A good number of Wi-Fi devices currently in the market use HTTP get and post messages for communication. This makes them very vulnerable, as an attacker can easily eavesdrop and generate spurious control messages. One solution to this is to upgrade device firmware by vendors to use HTTPS instead of HTTP. Another solution is to use separate wireless network for these devices. This minimizes possibility of threats, but doesn't eliminate it completely.

For devices without proper security in message exchanges, BEMOSS™ implements additional checking algorithms on the agent side to ensure that the device is not being controlled by unauthorized messages. These algorithms can be described as follows:

1. Periodic checking of status of a device and comparing it with expected status from the UI or algorithm can reveal if a device status has been undesirably changed. The agent can then send corrective control messages to the device and also issue warning messages to the upper layers of probable threats to the device.

2. Sniffing of outgoing control messages from the agent hosting system and comparing it with actual control commands in the message bus for control agents can reveal if any malicious code/agent is sending unauthorized control messages to the device. This can identify attacks to the agent host and issue warnings. In extreme cases, the communication interface of the agent host can be blocked to prevent any more sending of malicious control messages to devices.

To summarize, the security measures taken in connectivity layer and hardware devices are:

1. Use key-based encryption methods whenever possible, to ensure encryption, authentication and message integrity.

2. Secure physical communication medium, by securing network wires, using tamper resistant technologies for hardwire devices or in case of wireless network, by creating separate networks less accessible from outside.
3. Use checking algorithms in agents to ensure devices are being controlled by authorized control commands only.

3.6.3. Security of Connection to the Public Internet

BEMOSS™ connection to a local utility exists to allow BEMOSS™ to receive electricity price signals and demand response (DR) signals via a standard protocol, like OpenADR. OpenADR is a communication data model designed to interact with DR signals by automating DR actions performed by BEMOSS™.

Information flow in the OpenADR architecture can be discussed in the following steps:

1. The utility or ISO defines DR events and price signals sent to Demand Response Automation Server (DRAS).
2. DR event and price signals are published on a DRAS.
3. A DRAS client, in this case BEMOSS™, requests event information from the DRAS every specific time interval, e.g., one minute. The DRAS can also push information to the client.
4. BEMOSS™ determines load management actions based on events and price signals, and carries out load shedding/shifting strategies.

3.6.3.1. Security Concerns

Security concerns related to connection to a local utility, which is based on Internet communications, includes both the utility side (server side) and the BEMOSS™ side (client side). Main security concerns are:

- An attacker intercepts information sent between the server side and the BEMOSS™ client side to gain knowledge of DR events, pricing information and customer information. This can lead to loss of confidentiality, e.g., the exposure of customer data, unauthorized modification of information, manipulation of information and malicious attacks.

- An attacker may manipulate BEMOSS™ to communicate with a fake DRAS. This may make BEMOSS™ to receive false price/DR signals, which may result in turning on (or off) selected loads in the building. This can cause excessive loads to the grid, and grid instability, as well as financial impacts on customers. An attacker may also issue false time synchronization, causing events to occur sooner or later than the original schedule. In addition, it may make the utility server not be able to record the actual response of BEMOSS™ to the DR events received.
- An attacker may disable BEMOSS™ from receiving incoming DR signals. This can be done using denial of service attacks when an attacker floods the communication channels between the server and the client (BEMOSS™) with non-DR related Internet traffic.

3.6.3.2. BEMOSS™ Security Measures

Security features of OpenADR 2.0 conform to NIST Cyber Security requirements, and follow the guidelines provided by the "Security Profile for OpenADR" prepared by the UCAIug OpenADR Task Force and SG Security Joint Task Force. OpenADR 2.0 provides security services like authentication, confidentiality and integrity by implementing Public Key Infrastructure (PKI) certificates in both the utility server and the OpenADR client. Two primary public key cryptography algorithms supported by the utility server are:

- Elliptic Curve Cryptography (ECC) – 256 bits or longer keys
- Rivest, Shamir, and Adelman (RSA) – 2048 bits or longer keys

To establish a secure communication channel between the utility server and BEMOSS™, BEMOSS™ uses a certificate from the approved list of certificates. BEMOSS™ obtains a CA certificate(s) from a commercial vendor, i.e., an approved certificate authority (CA), and will use this certificate to issue OpenADR certificates to connect to the utility server.

4. Learning Algorithms for Building Loads

This dissertation proposes a set of learning algorithms for building loads based on the reinforcement learning (RL) framework. These algorithms run on the agents responsible for monitoring and control of loads, learn customer comfort preferences automatically, and operate the loads in an energy efficient manner which is also optimal for customer comfort. These algorithms also assist the incentive-based DR algorithm proposed later in Chapter 5. The current chapter first discusses the basics of RL and how it fits into the building energy management picture. Later, the details of the proposed algorithms for different types of loads are discussed.

4.1. Basics of Reinforcement Learning

In the reinforcement learning (RL) algorithm, the formulation starts from a Markov Decision Process (MDP) if the learning task satisfies Markov property [192]. In very general terms, a MDP is an extension of a Markov chain. In a Markov chain, a system is represented to have a number states that it can be in. There is a state transition probability for the system to be in a particular state or to move to one of the other possible states. A MDP is a system where at each state, one or more actions can be performed, which provides state transition probabilities to move the system to another state or keep it in the same state. A system is known to have Markov property and therefore be a MDP, if at each state, the current state and the number of actions that can be performed at the current state are known to the decision maker. A finite MDP is one in which the number of states are finite.

In the RL problem, an action in a particular state and the corresponding state transition has an associated reward value, which acts as a motivation for the decision maker to prefer the action. So, to summarize three very essential terms in the reinforcement learning problem:

State: A state is a signal from the environment that denotes the current situation a system is in. This can be a function of the environmental variables, i.e., one state can mean one particular set of values for the environment variables. In a finite MDP, the system has finite number of states. Let us consider the set of all possible states of the system as S, so any state in the system $s \in S$.

Action: An action is a decision that the decision maker agent can make at a certain state. Given the Markov property, at any state all possible actions for that state is known to the agent. Let us consider the set of actions as A. so, any possible action a ϵ A.

The state transition function δ denotes the function that governs which states can be reached from a certain state performing certain actions:

$$\delta : S \times A \rightarrow S$$

The system has a transition probability function, that denotes the probability of the system to move to a state s' given an action a is performed at the current state s. So, state transition probability,

$$p(s'|s,a) = \Pr\{s(t+1) = s'|s(t) = s, a(t) = a\}$$

For simpler MDPs, $p(s'|s,a) = 1$, if s' can be reached from s by performing a,

And, 0, otherwise.

Reward: A reward is an associated value for performing an action a at state s. So, the reward function,

$$r : S \times A \rightarrow \mathfrak{R}$$

r is the expected value of reward for moving to state s' from state s by performing a.

$$r(s',a,s) = \mathrm{E}[r(t+1)|s(t) = s, a(t) = a, s(t+1) = s']$$

In the RL problem, given the nature of the problem and the specific algorithm chosen, this reward values can be learned by the system over time.

The goal of a RL agent is to learn the policy π that maximizes the cumulative reward. So,

$$\pi : S \rightarrow A$$

And,

$$\pi (s(t)) = a$$

so that total cumulative reward is maximized.

The cumulative reward for policy π at state s(t) is given by the total reward that can be accumulated starting from initial state s(t),

$$V^{\pi}\big(s(t)\big) = \mathrm{r(t)} + \sum_{i=1}^{\infty} \gamma^{i} r(t+i)$$

Here, γ is called the discount factor which governs the effect of future rewards in the calculation of cumulative reward.

If $\gamma = 0$, only current reward is taken into consideration.

The closer γ is to 1, the more important future rewards are with respect to current reward.

This can help to adjust the balance between immediate short-term reward versus long-term reward in terms of choosing a policy. So, the goal of the learning agent is to determine the optimal policy π^* that maximizes the value of cumulative reward:

$$\pi^* = {}^{argmax}_{\pi} V^{\pi}(s)$$

The issue with learning this optimal policy is that there is no direct training data that provides the V values. So, an alternative approach is to use the Q-learning algorithm.

In a Q-learning algorithm, the Q values are calculated. A Q value for action a at state s is essentially the current reward for taking action a at state s plus the discounted value of all the future rewards for following the optimal policy thereafter. So,

$$Q(s,a) = r(s,a) + \gamma \, {}^{max}_{a\prime} \, Q(\delta(s,a), a')$$

Therefore, the optimal policy for an agent is to choose the action a from the available actions at state s, so that Q(s,a) is maximum.

But, the question that is more important is how this Q values can be estimated. Based on the recursive nature of the function Q, it can be estimated by iterative approximation. So, the agent has to choose actions at each state and see the following rewards and then update the Q function accordingly.

So, a learning agent starts with an initial value of Q function, and then visits state action pairs and updates Q values based on its experience with the rewards obtained. The algorithm is a simple value iteration update. It starts with the old value and makes a correction based on the new experience. A more detailed mathematical expression is:

$$Q'(\,s(t), a(t)) = Q(s(t), a(t)) + \ \alpha.\left(r(t+1) + \ \gamma\ .\max_{a} Q(s(t+1), a) - \ Q(s(t), a(t))\right)$$

Where,

$Q(s(t), a(t))$ = old Q value

$Q'(\,s(t), a(t))$ = new Q value

$r(t+1)$ = scalar reward for taking action a at state S

γ = discount factor

α = learning rate (0< α<1)

$\max\limits_{a} Q(s(t+1), a)$ = estimate of optimal future value

$a(t)$ = action that can be taken at time t and state S(t)

As the issue with any iterative algorithm, the question of convergence arises. For Q-learning, the convergence theorem states that if each state-action pair is visited infinitely often, then the estimated value of Q converges to the optimal value of Q. In practice, it is very difficult to reach without enormous amounts of iteration, and therefore a near optimal value is considered sufficient for operation.

There are two types of learning environments: offline learning and online learning.

In ***offline learning***, the reward values are deterministic and known, therefore, the iterative algorithm can be run to find out the optimal policies.

Online learning is when the agent learns through actually interacting with the environment, accumulating feedback from the environment and updating policies on the run.

Online learning is therefore more challenging, as an unfavorable action of the agent can have very unsatisfactory consequences.

This brings in the discussion of exploitation vs exploration in reinforcement learning.

Exploitation is the strategy of the agent to always choose the higher Q valued action to maximize rewards based on current Q function. In simple terms, this means doing what it knows to be good and not exploring alternatives. This strategy is only good if the Q value function is already optimal or near optimal. But, if it is way off, then the strategy misses out finding more rewarding action.

Exploration, on the other hand, is the strategy of visiting state-action pairs that have been visited lesser number of times before, in order to get a better estimate of Q values. This strategy helps the iterative algorithm to update Q values, and is the best way to find optimal values in offline learning. But, in case of online learning, choosing less rewarding actions may be very unfavorable depending on how good the current estimates of Q values are.

Therefore, a balance is required between exploitation and exploration for online learning agents. A more aggressive agent pursues exploration more, whereas a greedy agent exploits more to maximize rewards based on current knowledge.

4.2. Reinforcement Learning in Building Energy Management

The previous section discussed the basics of reinforcement learning and Q–learning to introduce the key concepts that are applicable for any application. This section focuses more on its application on building energy management and how it can be modified to apply for the different types of loads in buildings.

Based on the previous discussion, we can consider the building energy management system as an online learning environment. This certainly limits the capabilities of reinforcement learning, as any action taken by the agent impacts the physical environment of the building, and therefore directly affect the comfort of the user inside that environment. The exploration of the full state-action space through random actions is therefore not a viable option at all. Therefore, the efficacy of a model-free Q-learning algorithm that can learn policies by only interacting with the environment without any previous knowledge of the model of the environment, cannot be utilized.

The following are the concerns for applying Q-learning in a building energy management application:

1. The state-action space has to be finitely defined and still it may not be fully explored to avoid extreme dissatisfaction of the user.
2. The near optimal values have to be estimated with as less number of visits to state-action pairs as possible to avoid user inconvenience.
3. Both energy saving and user comfort has to be considered in the formulation of reward values.
4. Energy saving rewards are deterministic and can be pre-calculated based on a supervised learning of the model of a load.
5. User comfort rewards are to estimated and updated based on the user's control commands for the loads.
6. A balance between exploitation and exploration is a key element in this scenario.
7. The algorithms have to be sufficiently light-weight to be deployed in low cost computers for real-time decision making.

This dissertation proposes modified learning algorithms that consider all of these for different loads in a building. The learning algorithms will be categorized and the details will be described for each type of load. But, before that, a brief common discussion on some considerations are made here that apply to all of the loads.

4.2.1. Estimating User Comfort Rewards

Two strategies can be proposed to estimate the user comfort rewards for choosing actions at each state of the load:

Learning through positive feedback on user selected actions:

This strategy assumes that the user will change the load settings based on their comfort requirements, and the learning agent will be completely passive during its learning phase. In this case, the agent does not take actions by itself to find out rewards, but instead observes how user take actions at certain states to assign positive rewards to those actions.

So the algorithm is to assign:

$$r_c'(s,a) = r_c(s,a) + \beta$$

If user chooses action a at state s.

Where,

$r_c'(s,a)$ is the new comfort reward value for action a at state s

$r_c(s,a)$ is the old comfort reward value

β is an increment factor that depends on number of visits to this state-action pair

An example can be a user changing the temperature set-points at different environmental conditions, and the learning agent assigning positive user comfort rewards to those changes.

This is the most non-intrusive way of learning user comfort preferences, and completely favors exploitation over exploration. The end goal is to automate decision making after sufficient learning time, so that users do not have to make the changes by themselves in future.

Advantages of this learning strategy:

1. This is very convenient for the user with minimal dissatisfaction chances, as the learning agent does not make any change that has not previously been made by the user themselves.
2. This method can be useful to predict occupancy hours without occupancy sensors, only through the learning.

Disadvantages:

1. This learning method does not explore the major part of the state-action space that the user does not use in their control commands.
2. It is fully reliant on the comfort rewards and gives very little value to energy saving rewards.

Learning through negative feedback from user on uncomfortable actions:

This strategy considers exploration as the way to learn user comfort. If exploitation is not highly rewarding at a certain state, the agent may take less favorable actions to learn user reaction. If an agent takes an undesirable action at a certain state, the user can put negative feedback by changing the action manually, which feedbacks the algorithm to select lower comfort rewards for that state-action pair. Similarly, if a less-trained agent makes a correct random action not met by user's negative feedback, then this causes the algorithm to assign higher comfort rewards associated with that state-action pair so that in future the agent tends to choose similar actions.

So, agent assigns:

$$r_c'(s,a) = r_c(s,a) + \beta_{inc}$$

If action a was taken by agent at state s and the user did not intervene.

Where,

$r_c'(s,a)$ is the new comfort reward value for action a at state s

$r_c(s,a)$ is the old comfort reward value

β_{inc} is an increment factor that depends on number of visits to this state-action pair

On the other hand, if the user changes the command, it provides negative feedback to the agent in the form:

$$r_c'(s,a) = r_c(s,a) - \beta_{dec}$$

Where, β_{dec} is a decrement factor that depends on number of negative feedbacks to this state-action pair.

So, the more negative feedbacks are provided, the less likely the agent will be to visit that state-action pair again.

Advantages of this learning strategy:

1. This strategy is good for exploration, and can be used to better predict the near optimal Q values.

2. This strategy can be used to find actions that are more energy saving.

3. Can be used to predict occupancy, if properly designed.

Disadvantages:

1. This learning method may cause user inconvenience, based on how often it chooses less comfortable actions for user to intervene.

The exploration can be geared towards actions that increase energy savings. This way, the exploration is not random, and can faster find optimal values for balancing energy saving and comfort.

This dissertation prefers the negative feedback strategy over the positive feedback one, as finding opportunities of energy saving is one of the objectives of this work.

4.2.2. Calculating Total Rewards

The total reward for an action at a state has two different associated rewards. One is the reward from energy savings and the other one is the reward from user comfort.

Reward from energy saving, $r_e(s,a)$ is calculated based on the energy saving due to going from state s to state s' by performing action a. But, if the reward is based on just the value of energy saving, it cannot be used as a standard, because different loads may have different energy saving values for the same action at same state. For example, same setpoint change can cause different energy savings for two HVAC loads with different rating. Hence, the reward value should be normalized as well as scaled if required. So,

$$r_e(s,a) = \frac{E(s) - E(s')}{E_{max}} * \rho_e$$

Where,

$E(s)$ = Energy consumption at state s

$E(s')$ = Energy consumption at state s'

E_{max} = Maximum possible energy consumption for normalizing

ρ_e = Additional scaling factor

For loads whose power consumption is constant at a certain state and change proportionately with state change, this reward value can also be calculated based on power consumption values.

$$r_e(s,a) = \frac{P(s) - P(s')}{P_{rated}} * \rho_p$$

Where,

$P(s)$ = Power consumption at state s

$P(s')$ = Power consumption at state s'

P_{rated} = Rated power of the load

ρ_p = Additional scaling factor

Based on these equations, the energy saving reward can be positive for changing to a state with lower consumption, zero for staying at the same state, and negative for changing to a state with higher consumption.

If only energy saving reward was considered, the learning algorithm will always choose to move towards states with lower consumption. But, this is balanced by adding the comfort reward values to get the total reward values. The total reward values are given by,

$$r_t(s,a) = A_e * r_e(s,a) + A_c * r_c(s,a)$$

Where, A_e and A_c are scaling factors.

If the original reward values are properly scaled, the total reward value can be calculated as:

$$r_t(s,a) = A_{ec} * r_e(s,a) + (1 - A_{ec}) * r_c(s,a)$$

Where, A_{ec} is the energy comfort balance factor, and $0 < A_{ec} < 1$

If $A_{ec} = 0$, only comfort is considered.

If $A_{ec} = 1$, only energy saving is considered.

The energy saving rewards are pre-calculated for different loads based on their consumption profile. The comfort reward values are initialized based on initial settings on the building energy management system, and are updated based on exploration and user feedback.

4.2.3. Consideration of Occupancy

Occupancy of a room or a zone in a commercial space can be detected by utilizing an occupancy and/or motion sensor placed in a key location of the room or zone that can accurately detect the presence of occupants in the space. This variable can take data from the set {TRUE, FALSE} and is very significant in making control decisions.

$$Occ(t) \in \{TRUE, FALSE\}$$

If accurate detection of occupancy is possible, then the control algorithm can be divided into two parts: algorithms for learning when occupied, and switch to predefined default state when unoccupied.

If $Occ(t) = FALSE$:

If no occupancy is detected at a certain room or zone, it implies that the loads in that space can be controlled without violating user comfort. Therefore, it is an ideal opportunity to save energy by reducing load consumption. The unoccupied case can be further divided into two categories:

1. Unoccupied hours
2. Unoccupied during occupancy hours

Most commercial buildings have designated occupancy hours based on the working hours of the business. This is a known parameter and can be part of the initial agent knowledge.

The action of the agent during designated unoccupied hours is therefore straightforward. It will default to the state with maximum energy savings. This can be automated by setting A_{ec} = 1 for total reward calculation during unoccupied hours. Also, the state transition has to be designed so as to reach the minimum energy state in shortest possible time from the start of unoccupied hours in order to receive maximum energy saving benefits.

The case of no occupancy during occupied hours requires a different approach. This also depends on the type of load. Therefore, this case will be discussed in detail during the description of the algorithms.

If $\boldsymbol{Occ(t) = TRUE}$:

If occupancy is detected in either occupied or unoccupied hours, the learning algorithm has to take into account the comfort reward values. This is the period when the algorithm has to learn the user comfort preferences based on its exploration and feedback, and then later exploit it to gain maximum energy savings while maintaining comfort. The algorithms for different load types is developed which addresses the approach in this case.

If occupancy data is not available:

If an occupancy sensor or any method of accurate detection of occupancy is not available, the algorithm itself has to learn occupancy hours based on its exploration. While it is possible through any type of load, the easiest way to learn occupancy is through exploration of states of lighting loads due to two reasons:

A) A lighting load is always available for any zone or room in a building

B) Unfavorable changes in lighting load are more easily detected and is more likely to receive a negative feedback from a user.

Therefore, an additional algorithm will be discussed for lighting loads to learn occupancy hours in case of unavailability of occupancy sensors.

The complication with learning occupancy hours is that the agent has to store, process and update the reward values for state-action pairs for different hours of the different days in a week. This creates a large state space which requires significantly more storage and more time for training than the case where only reward values for state-action pairs in occupied and unoccupied cases have to be stored and processed. Hence, for the sake of simplicity and deployment in low cost computers, the case studies in this work assume that occupancy sensors are available.

4.3. Learning Algorithms for Loads in Buildings

Commercial buildings have different categories of loads that can provide energy saving and demand response opportunities. Figure 4.1 (Source: [49]) shows the disaggregation of energy end-use consumption of small to medium-sized commercial buildings (buildings that are 50,000 square feet or less):

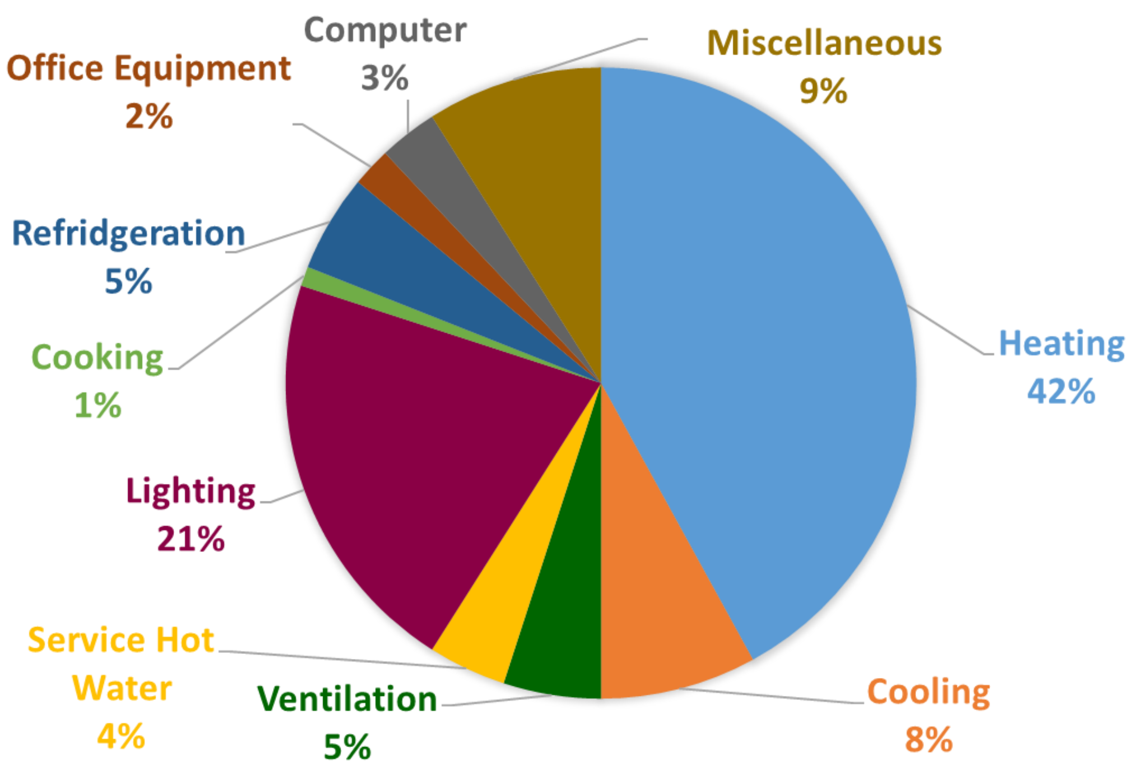


Figure 4.1. Disaggregation of energy end-use consumption of small to medium-sized commercial buildings (buildings that are 50,000 square feet or less) [49]

As can be seen from the figure, 55% of total consumption in small to medium-sized buildings can be attributed to HVAC loads. HVAC and lighting together consume almost 71% of the total energy, and remaining consumption is from plug-loads, water heating and refrigeration. Therefore, the loads in the small to medium-sized buildings can be categorized into three major categories based on consumption:

1) HVAC loads
2) Lighting loads
3) Plug loads

The learning algorithms proposed in this dissertation reside in the individual control agents within the BEMOSS™ platform. Each agent has its own learning algorithm, and the actions of the agent depends on the control opportunity of its associated load. Due to the difference in the nature of the loads in the three categories classified above and their usage pattern and control characteristics, the learning algorithms can also be classified into three categories based on the types of loads.

4.3.1. Learning Algorithm for HVAC Loads

HVAC loads consume the highest portion of total building consumption, and are directly related to customer comfort. Before developing the RL algorithms for HVAC loads, first, brief discussions on the HVAC comfort conditions and energy saving opportunities are presented.

4.3.1.1. HVAC Comfort Conditions

American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) defines standards for ventilation and indoor air quality requirements for comfort of building occupants [193]. According to the standards, the parameters that directly affect customer comfort are:

1) Temperature
2) Relative humidity
3) Air velocity
4) Air ventilation
5) CO_2 content of air

Based on ASHRAE standard 55-2013 [193], an index named predicted mean vote (PMV) can be used to measure the indoor thermal comfort condition. PMV is a value that denotes average thermal sensation response from a large number of people. It correlates with the percentage of people dissatisfied (PPD) index in that 80% or more of the occupants will be satisfied if the PMV value is found to be between -0.5 and +0.5. The ASHRAE thermal sensation scale given in Table 4-1 shows how PMV is related to thermal sensation.

Table 4-1. ASHRAE Thermal Sensation Scale

PMV value	Sensation
+3	Hot
+2	Warm
+1	Slightly warm
0	Neutral
-1	Slightly cool
-2	Cool
-3	Cold

According to Fanger's comfort analysis [194], PMV can be calculated from six psychrometric variables: air temperature, mean radiant temperature, air velocity, air humidity, clothing resistance, and metabolic rate due to activity level.

Based on ASHRAE standard, the air velocity should be kept below 30 feet/min, which can be governed by the HVAC unit automatically.

If a typical summer indoor clothing condition is considered (0.5 clo), with a metabolic rate of 1.1 met (typical value for typing, reading or writing while seated), and the air velocity is considered to be 20 feet/min, then the relative humidity vs dry bulb temperature for -0.5 to +0.5 PMV can be found by [195], which uses the formulation by Fanger's comfort analysis [194]. This is given in Fig 4.2.

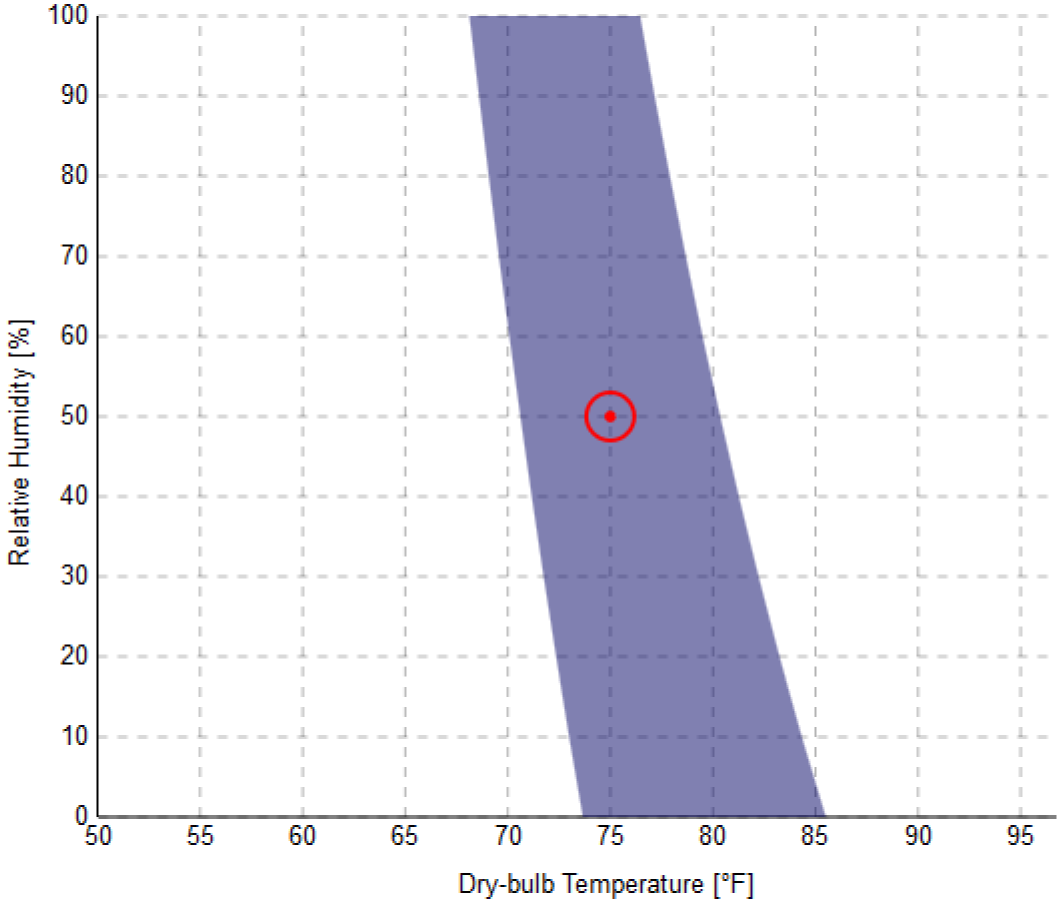


Figure 4.2. Relative humidity vs dry-bulb temperature for PMV=-0.5 to +0.5. [Source: 195]

Based on the ASHRAE standard, for a comfortable relative humidity level of 50 to 60%, the room temperature should be between 70 and 80^{o}F for a PMV value of -0.5 to +0.5.

Therefore, considering a typical summer office environment, the comfortable range of set-point for thermostats is 70 to 80^{o}F. For winter conditions, this range is from 68 to 78^{o}F approximately.

As the relative humidity is dependent on the humidity of outside air and the ventilation of the HVAC unit, we can consider that the comfortable PMV index can be maintained by directly controlling the thermostat set-point. This is the simplest form of HVAC control, and can be used for energy savings.

4.3.1.2. Savings from Set-point Control

Set-point control can offer large energy savings for HVAC loads. Figure 4.3 (Source: [196]) shows simulated energy savings incurred for changing set-points in heating and cooling modes in buildings. The results show as high as 65% savings for increasing cooling set-point by 10° F. Although the data is simulated, it gives a guideline on how much savings is possible through set-point control.

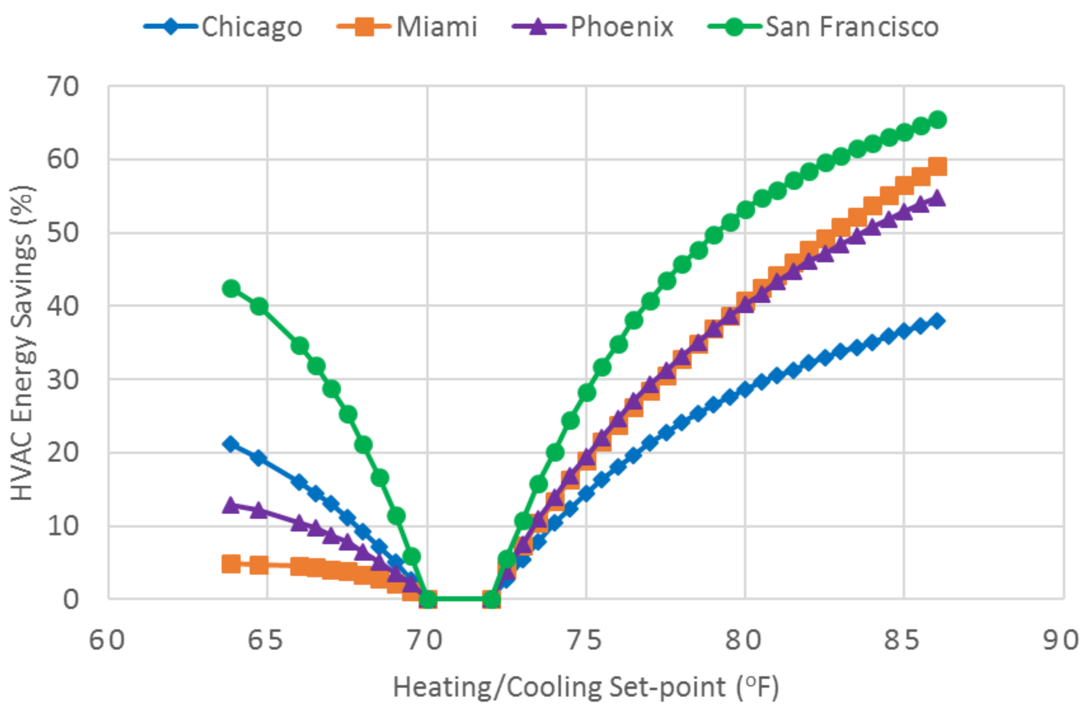


Figure 4.3. Summary of average HVAC energy savings in different cities, compared to baseline. [196].

Based on the analysis of energy saving potential of set-point control and the comfort conditions, this dissertation proposes RL algorithms for set-point control.

4.3.1.3. Algorithms for Set-point Control

4.3.1.3.1. Defining State-action Space

First step for developing a RL algorithm is to define the state space of the MDP of the system. This work considers the possible set-points on a thermostat as the possible states for the HVAC learning agent. Most commercial thermostats offer a maximum and minimum value of temperature set-point with an increment/decrement of 1^{O}F [Nest, RadioThermostat]. Instead of considering the actual possible range of set-point offered by the thermostat, a shortened range is considered for reducing storage requirements, based on comfort condition.

Let us consider the following:

T_{lowest} = lowest allowable set-point by learning agent

$T_{highest}$ = highest allowable set-point by learning agent

ΔT = increment/decrement in set-point possible on thermostat

In that case, we can consider the set of all the possible states of the MDP as:

S = $\{T = T_{lowest}\ , T = T_{lowest} + \Delta T, T = T_{lowest} + 2\Delta T, \ldots., T = T_{highest} - \Delta T, T = T_{highest}\}$

Hence, total number of states = $\frac{T_{highest} - T_{lowest}}{\Delta T} + 1$

At each state, there are three possible actions:

1. Keep the current set-point T
2. Raise set-point by ΔT to reach the state for $T + \Delta T$
3. Lower set-point by ΔT to reach the state for $T - \Delta T$

This is the simplified model, in which, from each state only the two adjacent states can be reached.

A more complete model is to consider that the temperature can be raised or lowered by a multiple of ΔT as an action to reach any state in the set S. The action space in that case is:

$$A : S \to S$$

Hence, a reward function takes the form of a $S \times S$ matrix, which looks like:

Table 4-2. Form of Reward Matrix

		Transition States					
		T_{lowest}	$T_{lowest}+\Delta T$	$T_{lowest}+2\Delta T$	$\cdots$	$T_{highest}-\Delta T$	$T_{highest}$
Current States	T_{lowest}	r_{11}	r_{12}	r_{13}	$\cdots$	$r_{1(n-1)}$	r_{1n}
	$T_{lowest}+\Delta T$	r_{21}	r_{22}	r_{23}	$\cdots$	$r_{2(n-1)}$	r_{2n}
	$T_{lowest}+2\Delta T$	r_{31}	r_{32}	r_{33}	$\cdots$	$r_{3(n-1)}$	r_{3n}
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	$T_{highest}-\Delta T$	$r_{(n-1)1}$	$r_{(n-1)2}$	$r_{(n-1)3}$	$\cdots$	$r_{(n-1)(n-1)}$	$r_{(n-1)n}$
	$T_{highest}$	r_{n1}	r_{n2}	r_{n3}	$\cdots$	$r_{n(n-1)}$	r_{nn}

Two such matrices have to be formed, one for the energy reward r_e and other one for the comfort reward r_c.

This is how the state-action-reward space of the MDP is formed.

4.3.1.3.2. Calculation of Energy Reward, r_e

Based on the validated HVAC model, the energy consumption for different states of the system can be calculated. To keep it simple and consistent, the same outdoor temperature and the same HVAC model is considered, and the energy consumption is calculated based on the cumulative power consumption for a period of time to maintain room temperature within the deadband of the set-point.

Therefore, energy consumption at state s=T is given by:

$$E(s=T)=\frac{\int_{t=t1}^{t=t2} P(t)}{t2-t1}$$

Where, t1 to t2 define a time period and P(t) is the consumption at time t to maintain temperature at set-point $T \pm deadband$.

These energy values can be calculated based on the HVAC model for all states.

4.3.1.3.3. RL Algorithm for Unoccupied Hours

No exploration is required for unoccupied hours, therefore, A_{ec} is set to be 1 to ignore comfort rewards and the action with maximum r_e is chosen immediately at the beginning of the unoccupied hours.

As the thermal inertia of the zone causes some time to change the room temperature, the ending of the unoccupied hour is considered an hour earlier than the actual start of occupied hour. This ensures that there is sufficient time for the HVAC to bring the room temperature to comfortable range before the occupants start entering the building.

4.3.1.3.4. RL Algorithm for Updating Comfort Reward, r_c for Occupied Hours

The learning agent is given initial schedules as the starting point for its learning. Based on the schedule, the initial value of r_c is assigned to the scheduled set-point. After this, the agent calculates the total reward r_t and makes decision on exploration vs exploitation.

To make the decision, at each state, the agent calculates the reward benefits for moving to the adjacent two states. For HVAC, these are the states $(T_{current} + \Delta T)$ and $(T_{current} - \Delta T)$. The agent calculates the relative benefit for transition to these two states and then makes decision to explore if the benefit is more than a pre-specified value of exploration reward tolerance, $r_{exp,t}$.

Hence, if $(r_{transition} - r_{current}) > r_{exp,t}$: the agent makes the state transition for exploration, otherwise, the agent stays at the current state.

If the agent makes a state transition, and is not met with negative user feedback, then it increases the comfort reward values accordingly, as mentioned in section 4.2. If the user makes a negative feedback, it changes the state to user selected state, and updates comfort rewards for all states in between. This is how the comfort reward values are updated at each episode.

To prevent the RL agent from causing inconvenience for user by exploring randomly, the algorithm proposed here provides a guided exploration path and period.

Exploration path: As the goal is to save energy, the RL agent should explore the states that have higher energy rewards r_e. But, initially, the opposite path is chosen in order to find the other end of the user comfort preference. This is to find out a set-point limit that can be used for pre-cooling/pre-heating purposes. After exploring for enough number of times in order to get reward differences that prevent further exploration, the RL agent explores the states for saving energy. The final comfortable set-point is decided based on the maximum total rewards.

Exploration period: The exploration is done at an hourly interval, and only 1°F change per 5 minutes for a total change of 2°F. This is to ensure that the change doesn't cause drastic changes to the environment, and the user is given opportunity to react to the change. If the change meets no resistance, the reward values are updated, and the next change is made on the next hour. If user provides a negative feedback, then the next exploration is postponed for a certain no-change period, Δt_{nc}. In practicality, this period can be anything from 1 hour to 1 day.

To summarize the algorithm:

Step 1: Calculate total reward values

Step 2: At exploration interval, if $(r_{transition} - r_{current}) > r_{exp,t}$, make transition to new state

Step 3: Wait for user feedback. If no feedback is provided, go to step 4, else go to step 5.

Step 4: Update r_c for new state and prepare for next exploration. Go to Step 1.

Step 5: Update r_c for all states from explored state to user selected state.

Step 6: Wait for Δt_{nc}, and then go to Step 1.

4.3.2. Learning Algorithm for Lighting Loads

Lighting loads in commercial buildings can offer two types of control:

1) Basic ON-OFF control and

2) Dimmer control.

Traditional ON/OFF control is simple and easy to implement, and the hardware required is less expensive than the one required for dimmer control. Dimmer control can be of two types:

a) Step-dim

b) Continuous or analog dim

Step-dimming usually offers two or three step dimming, while continuous dimming offers dimming of brightness level from 0 to 100% in an analog manner.

For designing lighting control strategies in commercial buildings, lighting comfort conditions have to be figured out first.

4.3.2.1. Lighting Comfort Conditions

The Illuminating Engineering Society of North America (IESNA) defines standard for illuminating building and open spaces. The standard specifies average illumination level, required lighting power densities (LPD) for different types of spaces. Based on the standard 90.1-2007 [197], the lighting level recommendation in lux is given by Table 4-3.

Table 4-3. Recommended Illumination for Building Spaces [197]

Building Type	Space Type	Maintained Average Illuminance at working level (lux)	Measurement (working) Height (1 meter = 3.3 feet)
Educational Buildings	Play room, nursery, classroom	400	at 0.0 m
	Lecture hall	400	at 0.8 m
	Computer practice rooms	30	at 0.8 m
	Classrooms	300	at 0.8 m
Office Buildings	Single offices	300	at 0.8 m
	Open plan offices	400	at 0.8 m
	Conference rooms	300	at 0.8 m
Hospitals	General ward lighting	300	at 0.8 m
	Simple examination	500	at 0.8 m
	Examination and treatment	1000	at 0.8 m
Hotel and restaurants	Self-service restaurant, dining room	100	at 0.8 m
	Kitchen	500	at 0.8 m
	Buffet	100	at 0.8 m
Sport facilities	Sport halls	300	at 0.0 m
Wholesale and retail sales	Sales area	500	at 0.8 m
	Till area	500	at 0.8 m
Circulation areas	Corridor	50	at 0.0 m
	Stairs	50	at 0.0 m
	Restrooms	300	at 0.0 m
	Cloakrooms. washrooms, bathrooms, toilets	300	at 0.8 m

The standard also specifies minimum lighting requirements, which can be used to determine lower bounds for brightness adjustments.

4.3.2.2. Basics of Brightness Control

If the lighting control device allows dimmer control, then the control agent make energy savings by reducing brightness level. The total illumination at any space is given by:

$$lux_{total} = lux_{ambient} + lux_{lighting}$$

Where,

lux_{total} = Total effective illumination level

$lux_{ambient}$ = Illumination from the ambient sources

$lux_{lighting}$ = Illumination from the lighting load

The standard comfort condition has to be maintained for lux_{total}. Now, $lux_{ambient}$ is dependent on the setting of the building space and daylight conditions. If the building space has windows through which daylight from outside can come into the space, then it contributes to the total illumination level. Also, if the space is adjacent to another space with window or transparent separation in between, then the lighting source on the other space contributes to the ambient lighting of the space. As these are not directly under control of the control agent, the adjustment can only be made by the agent on $lux_{lighting}$ through dimmer control. Therefore, a feedback control can be designed based on reading of lux_{total} by using an ambient light sensor.

$lux_{lighting}$ is directly correlated with brightness level on the dimmer. For LED lights, the relation is usually:

$$lux_{lighting} = \frac{brightness\_level(\%)}{100} \times lux_{100\%}$$

Hence, the brightness level can be adjusted to change the light output of the lighting loads. It is also directly correlated with power consumption of the lighting load, which is given by:

$$P_{lighting} = P_{0\%} + \frac{brightness\_level(\%)}{100} \times (P_{100\%} - P_{0\%})$$

Where,

$P_{0\%}$ = base power consumption of the lighting controller

4.3.2.3. Savings from Brightness Control

Figure 4.4 shows possible power savings from dimming LED lights by WattStopper lighting controller, which gives an idea of how much saving is possible by changing light intensity of lighting loads.

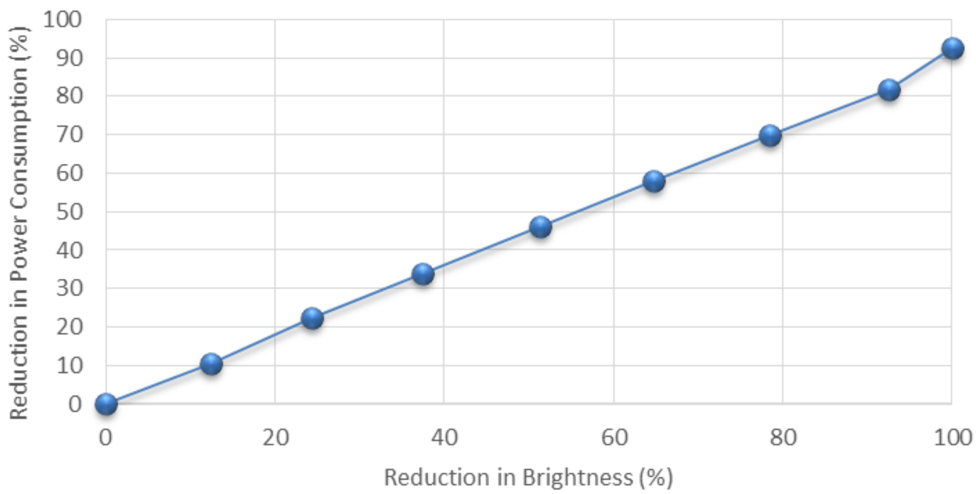


Figure 4.4. Power savings by reduction in brightness of LED lighting controlled by Wattstopper lighting controller. [Experimentation conducted at Advanced Research Institute, Virginia Tech]

4.3.2.4. Algorithms for Brightness Control

4.3.2.4.1. Defining State-action Space

This work considers the possible brightness levels on a lighting load controller as the possible states for the lighting learning agent. Considering a continuous dimmer controller, the possible brightness levels are 0 to 100 with an increment/decrement of 1 [WattStopper]. Therefore, the possible states for such a lighting control agent are:

0, 1, 2, 3, …. , 99, 100.

For a step dimmer control, this can be based on the number of steps. For example, a four-step controller, the possible states are: 0, 25, 50, 75, 100.

For a two-step dimmer, the possible states are: 0, 50, 100.

4.3.2.4.2. RL Algorithm for Unoccupied Hours

No exploration is required for unoccupied hours, but for lighting control, the RL agent cannot ignore comfort rewards in this case, because for certain spaces a minimum lighting level is required by standard, even during unoccupied hours. If RL agent only looks at energy reward, it may violate the standard by always choosing the state with maximum energy rewards, which is 0. So, in this case, the comfort reward values are designed in such a way so that it prevents the agent to go below the minimum brightness level for that space. Also, the reward difference is made high, so that the agent does not explore, and directly selects the minimum brightness level.

4.3.2.4.3. RL Algorithm for Updating Comfort Reward, r_c for Occupied Hours

The learning agent is given initial schedules as the starting point for its learning. Based on the schedule, the initial value of r_c is assigned to the scheduled brightness level. After this, the agent calculates the total reward r_t and makes decision on exploration vs exploitation. This algorithm is similar to the HVAC control algorithm, except, in this case, the agent only explores for energy benefits.

Exploration path: For lighting load control, the RL agent explores the states that have higher energy rewards r_e. The final comfortable brightness level is decided based on the maximum total rewards after certain number of exploration episodes that create reward difference which prevent further exploration.

Exploration period: The exploration is done at an hourly interval, with 1 brightness level change per minute for a total change of 10. This is to ensure that the change doesn't cause drastic changes to the environment, and the user is given opportunity to react to the change. If the change meets no resistance, the reward values are updated, and the next change is made on the next hour. If user provides a negative feedback, then the next exploration is postponed for a certain no-change period, Δt_{nc}. In practicality, this period can be anything from 1 hour to 1 day.

To summarize the algorithm:

Step 1: Calculate total reward values

Step 2: At exploration interval, if $(r_{transition} - r_{current}) > r_{exp,t}$, make transition to new state

Step 3: Wait for user feedback. If no feedback is provided, go to step 4, else go to step 5.

Step 4: Update r_c for new state and prepare for next exploration. Go to Step 1.

Step 5: Update r_c for all states from explored state to user selected state.

Step 6: Wait for Δt_{nc}, and then go to Step 1.

4.3.2.4.4. RL Algorithm with Ambient Light Sensor

If an ambient light sensor is available, and the lighting load is not the only source of lighting in the space, then the agent changes states on the basis of ambient light values. The minimum lighting level is determined based on the minimum illumination. Instead of assigning rewards to the brightness level, the reward values are assigned to the illumination ranges, so that the RL can learn the user preference of illumination.

4.3.2.4.5. RL Algorithm without Occupancy Sensor

If an occupancy sensor is not available, then the RL algorithm can be used to detect occupancy. This is done by expanding the state-space to time-based state space. This means, that the state-action-reward space is different for different time period. To reduce the storage complexity, a period of 1 hour is selected, so that there are only 24 state-action-reward space per day. Then, at each hour, the RL agent attempts exploration to find user feedback. If no user feedback is received, the agent can assume no occupancy, and can lower energy consumption by exploring further. The trained rewards also assist other RL agents to detect occupancy for that hour.

4.3.3. Learning Algorithm for Plug loads

Commercial miscellaneous electric loads can also be controlled using plug load controllers. The plug loads in a commercial building fall into different categories and load shedding opportunities are dependent on the category of loads.

The PNNL technical report on small and medium–sized commercial building monitoring and control needs lists the key plug load types in target commercial buildings as shown in Table 4-4 [49].

Table 4-4. Key Plug Loads in Target Building Types [49]

Office	**Retail/Service (Non-Food)**	**Education**	**Public Assembly, Public Order and Religious Worship**
Personal Computer (PC)	Cooking	PC	Cooking
Monitor	PC	Monitor	PC
Office Equipment	Walk-in Refrigeration	Office Equipment	Landscape Irrigation
Cooking	Vending Machine	Cooking	Walk-in Refrigeration
Residential Refrigeration	Monitor	Walk-in Refrigeration	Fitness Equipment
Distribution Transformer	Distribution Transformer	Vending Machine	Arcade
Vending Machine	Laundry	Distribution Transformer	Vending Machine
Vertical Transport	Unit Cooler	Ice Machines	Monitor
Unit Cooler	TV	Unit Cooler	Non-road Vehicles
	Automated Teller Machine (ATM)	Vertical Transport	Unit Cooler
	Residential Refrigeration	Residential Refrigeration	Residential Refrigeration

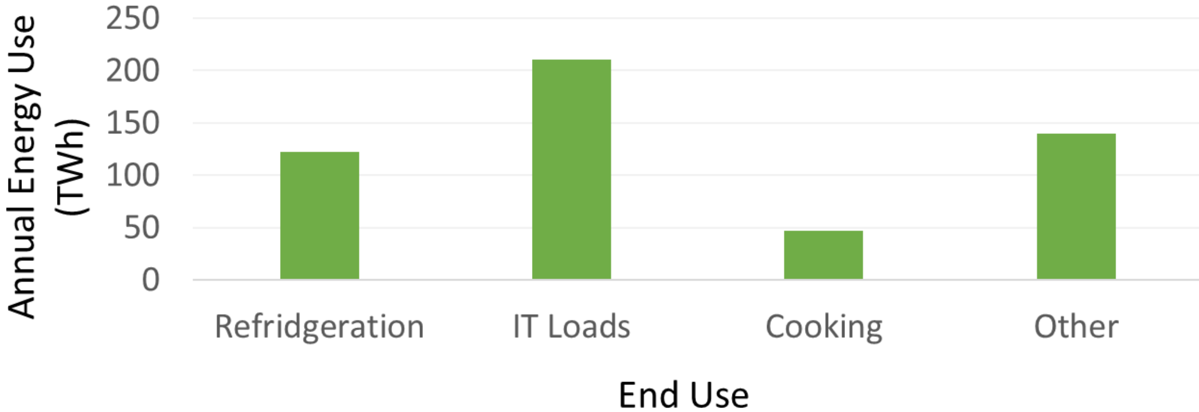


Figure 4.5. Energy usage of end-use plug loads by category [49]

Figure 4.5 [49] shows the energy usage of plug loads by category for all commercial building types, which shows that IT loads is the largest category in terms of energy usage, followed by refrigeration and others.

The plug loads are mostly process loads (loads that are required for the business operation) and therefore are of critical importance. Hence, plug loads offer very low potential of energy saving during occupied hours. But, they can save energy if controlled during unoccupied hours. The control algorithm for plug loads are therefore mostly tied to occupancy detection.

For IT loads, computer loads in an office environment cannot be turned OFF at any time, even during unoccupied hours. In many working environments, computers may be preferred to run by the user during unoccupied hours for running critical processes. To prevent unwanted work or data loss, computers are therefore not controlled in the algorithm. They can be configured to sleep/hibernate during unoccupied hours to save energy, but this is also dependent on user preference and manual control. Other IT loads, on the other hand, like printers, scanners etc., can be turned OFF during unoccupied hours. They can also be turned OFF during occupied hours if they are not in use for a long duration. The reinforced learning agent can experiment with control actions at different time periods and accumulate feedback over time to learn when these loads can be turned OFF without causing user discomfort.

For refrigeration type loads, the learning agent can control these loads based on occupancy and DR requirement. For example, if the set-point in a refrigeration unit can be controlled, it can be increased to higher values during unoccupied hours (up to a maximum allowed set-point). Navigant [198] estimates that these types of control can save between 20 and 35% of annual energy use in beverage merchandisers and refrigerated vending machines. These set-point control can also be used to pre-cool or deep freeze before a DR event and then increase set-point during DR event to save power by using the thermal storage of the devices.

For other plug loads, unless continuous operation is of critical importance, the loads can be turned OFF during unoccupied hours to save energy. Examples of this type of loads include electric cooking loads like oven, coffee-maker etc.

5. Incentive-based Algorithms for Building Demand Response

In this chapter, the step-by-step methodology is discussed for developing an incentive-based demand response algorithm for buildings. It starts by discussion of improving an incentive-based algorithm with priorities and preferences of residential loads fixed by the user and validating it in field implementation to prove its efficacy. Then, a learning-based DR algorithm is proposed in the following sub-chapter to control loads based on learned user comfort preferences instead of fixed priorities. Later, an incentive-based algorithm is proposed for learning agents in multiple buildings.

5.1. Incentive-based DR Algorithm with Fixed Priority and Preferences

The initial work of this dissertation started based on the ongoing work in Partnership For Innovation (PFI) project in Advanced Research Institute, Virginia Tech. An incentive-based algorithm for energy management was developed at Virginia Tech [89]. This algorithm controls power intensive loads based on a DR signal received from the utility specifying a DR event with time period and demand limit to be maintained during that time period. This work builds on this algorithm and proposes improved algorithms for robust communication, inclusion of smart thermostat control, and utilization of DER, specifically PV and storage, if available. The proposed improved algorithms are later validated to prove the efficacy of a priority based algorithm for demand response.

5.1.1. Algorithm to Address Communication Failures/Data Errors

During laboratory experiments with this previously developed algorithm, issues of communication failures and data errors were faced. ZigBee-based communication had been used in this system between a central energy management unit and the load controllers to collect sensor data and transmit DR decisions. The reasons underlying the occurrence of communication failures/data errors were found to be either communication errors from failure to communicate between the central unit and load controllers due to package drop or corrupted frames; or data errors due to microcontroller read errors. The laboratory experiment indicated that the communication failure/data error rates of these load controllers are in the range of 0.5-3.8% [199].

To avoid errors in operation, the energy management algorithm was improved to incorporate robustness against communication failures and data errors by incorporating a popular method used in communication networks: automatic repeat request (ARQ) [200, 201]. Although another popular method named forward error correction (FEC) is very effective in fixing bit errors by adding redundancy bits and error correction codes, it cannot be implemented in this case, as it compromises throughput due to limited resources in the smart plugs. On the other hand, ARQ methods [202] are simple to implement in sensor networks with low resources, and achieve reasonable throughput. A combination of the stop-and-wait (SW) and selective repeat (SR) ARQ schemes [200, 201] was implemented with some modifications of our own. In SW, sender waits for acknowledgement after transmitting packets, and in case of negative acknowledgement or no response time-out, re-sends the packet. In the basic SW scheme, this re-transmission continues until a positive acknowledgement is received. But for this case, one cannot indefinitely continue retransmission as that will stop the BEM from making decisions.

Therefore, in the modified algorithm [199], the BEM sends data request to each smart plug, and stops and waits for its response for a time-out of 5 seconds. If no/erroneous response is detected, the BEM uses the last recorded correct data for that smart plug and goes forward to send data request to the next one. At each cycle, the BEM keeps a counter of failed communications for each smart plug, and in case of three consecutive failures with a particular smart plug, it announces a warning signal indicating communication lost with that smart plug. The improved algorithm is shown in Figure 5.1. This improvement, in spite of its simplicity, successfully prevents BEM from using incorrect appliance data for calculations, thereby making it more resilient to handle communication failures/data errors.

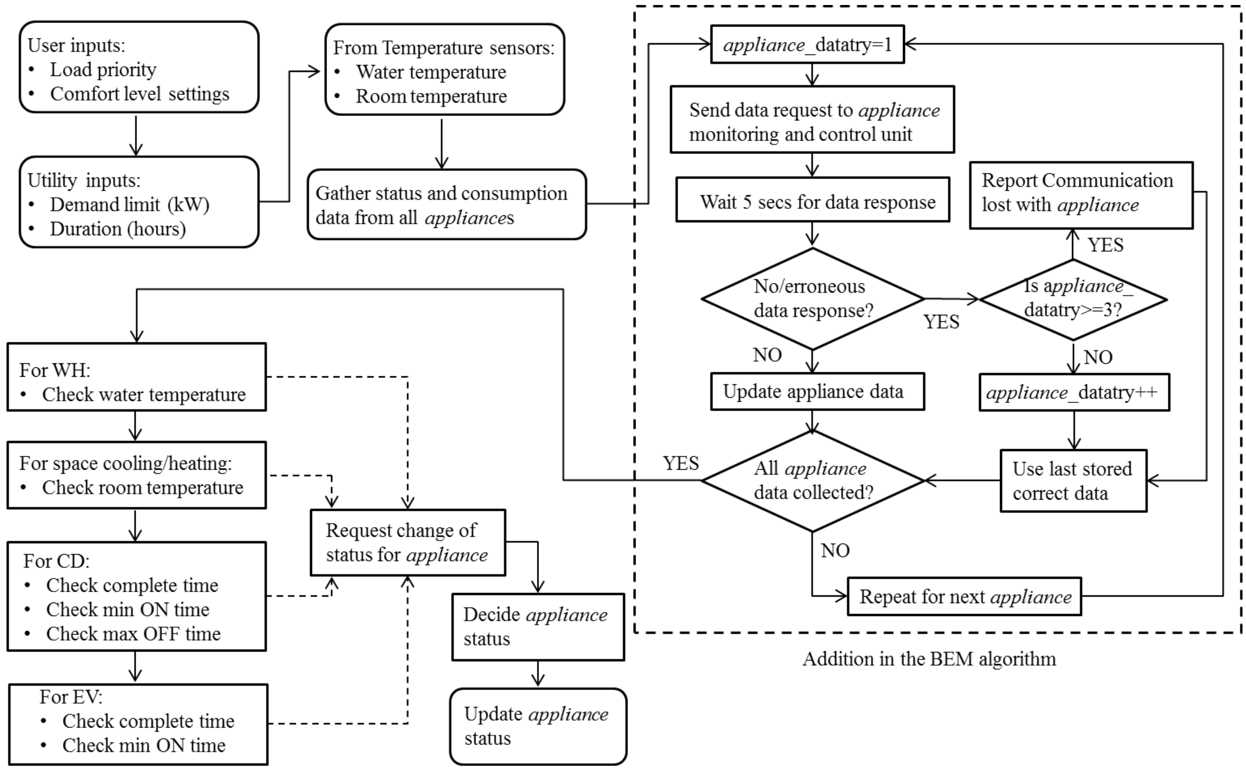


Figure 5.1. Improved BEM algorithm to handle communication failures/data errors.

The algorithm in Figure 5.1 is the BEM algorithm for residential applications. It focuses on power-intensive loads commonly available at households: water heater (WH), space cooling/heating unit (AC), clothes dryer (CD) and electric vehicle (EV). The demand response is implemented through DR events designed by utilities. A DR event is defined as a period of time during which the total power consumption of a household is to be limited within a demand limit specified by the utility. Another variation of this implementation is possible where the utility specifies the amount of power (kW) to curtail instead of the demand limit. The customer can be motivated to participate in DR events through monthly or annual monetary incentives, besides savings due to lower consumption during a peak-pricing period. To implement this, the utility sends a signal to the customer with demand limit and DR event duration prior to the DR event. Also, the customer selects the priorities and comfort level settings for the power-intensive loads. These settings can also be factory-preset considering load types. The BEM algorithm starts by collecting customer inputs of load priority and comfort level settings and utility inputs of demand limit and DR event duration. Then it starts collecting data from the sensors and the smart plugs/ smart appliances. While collecting data, it employs the improvement just discussed. Once data from all appliances have been collected, the algorithm checks if the current ON/OFF status of appliances violates any comfort level settings or the demand limit. If no demand limit has been imposed, the algorithm turns ON/OFF the appliances to maintain the comfort level settings with efficient energy usage. On the other hand, during DR event, the algorithm restricts the total power consumption within the specified demand limit. In this case, it uses the preset/customer-defined priority settings with the comfort level settings to decide which loads to interrupt in case of demand limit violation. This is discussed in more details in [89]. Decisions are then conveyed to the smart appliances/smart plugs to implement changes in load status.

Although the algorithm shown in Figure 5.1 is for residential customers, this can be applied to commercial/industrial buildings as well. For example, for commercial buildings, Heating, Ventilation and Air Conditioning (HVAC) loads can be controlled by changing set-points on smart thermostats. The lighting loads are considered critical loads in residential buildings, but they can be partially controlled in commercial buildings based on occupancy and ambient lighting. Plug-loads in residential buildings are also considered critical loads, whereas in commercial buildings, selected plug-loads can be controlled during unoccupied hours (i.e., office printers can be turned off after office hours). Electric Vehicle chargers in commercial buildings can also be considered for load curtailment using this BEM algorithm. Validation of this algorithm is presented in the case study chapter of this dissertation.

5.1.2. Algorithm to Include Set-point Control of Smart Thermostat

A HVAC unit is one of the most power-intensive loads in a building. It is also a very crucial load for customers, because room temperature has to be maintained within an acceptable range of customer comfort. For this reason, room temperature has to be continuously monitored by a BEM system. Direct ON-OFF control of the HVAC unit by the BEM system is effective for maintaining demand limit, but might not be effective in maintaining room temperature. Also, direct ON-OFF control may impose risk of permanent equipment damage or may reduce lifetime of the unit. Inclusion of a smart thermostat in the BEM system to control HVAC gives the flexibility of temperature set-point control in the DR algorithm without the need of additional hardware. A BEM software can communicate with the smart thermostat to a) constantly monitor the current temperature, and b) change the temperature set-point based on algorithm decision.

The BEM algorithm is modified to incorporate the use of temperature set-point control that is enabled by the inclusion of smart thermostat in the system [203]. In the improved algorithm, the temperature set-point for a HVAC unit control is set to some default value, i.e., 76ºF (±2ºF) (This value can be adjusted by the customer). If the demand limit of the DR event and the priority of the HVAC with respect to other appliances forces the BEM algorithm to decide to cut-off HVAC unit to provide power for a higher priority load, instead of directly turning the HVAC OFF, the BEM changes the set-point temperature of the thermostat to a higher/lower temperature based on mode (cooling/heating). This causes the thermostat to turn OFF the HVAC unit. Thus the HVAC unit is shed to provide power to higher priority loads. However, if the temperature goes beyond the acceptable range for customer, the HVAC unit is served by shedding other appliances of lower priority if necessary to meet the demand limit requested.

When the HVAC unit is of the highest priority, the temperature set point is sent back to the default set point. This way, the room temperature can be kept within an acceptable range even when the HVAC unit is not the highest priority load.
Algorithms for other appliances in the system (WH, CD and EV) remain the same as before.

5.1.3. Algorithm to Include DER if Available

This chapter proposes a method to allow integration of DER in an incentive-based BEM system. The proposed algorithm takes advantage of batteries to store energy from PV sources, and dispatch from batteries during peak load hours. The modified BEM algorithm [204] with DER is depicted in Figure 5.2.

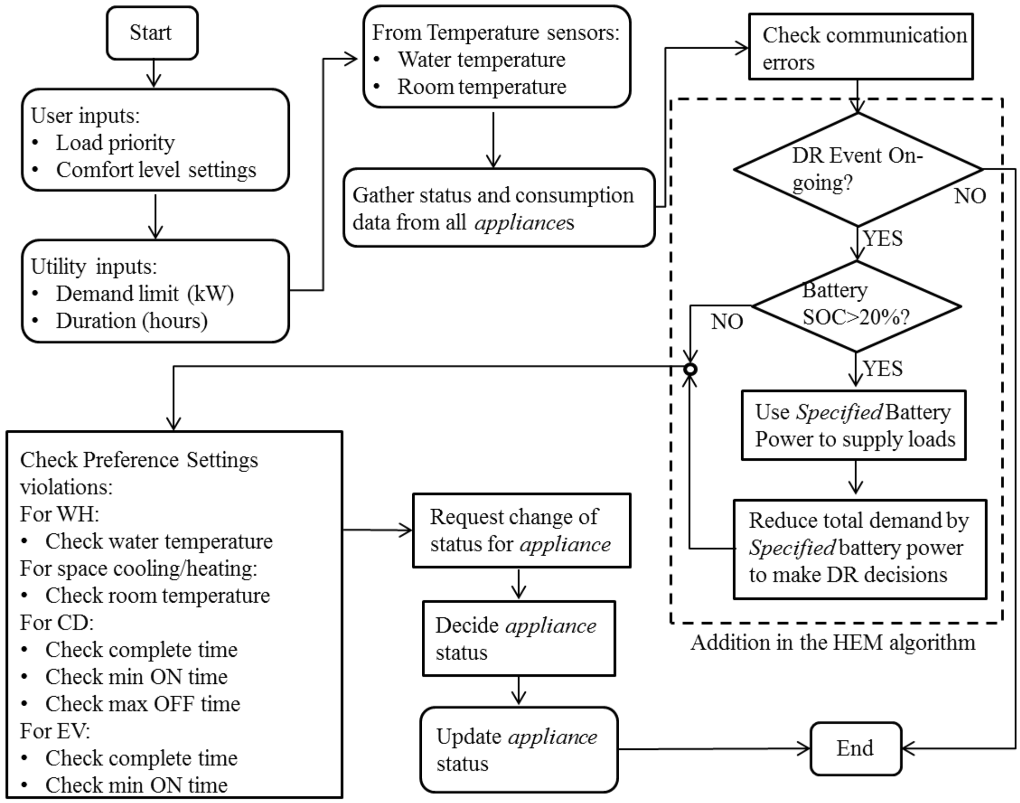


Figure 5.2. Modified BEM algorithm with DER.

As shown, the modified algorithm first gathers all data: customer priorities and preferences, utility signal inputs, sensor data, and current status and consumption of loads. Then it checks for communication errors during data collection and filters out bad data. Then the algorithm checks if a DR event has been initiated. If there is a DR event and the battery charge is available, the fixed amount of battery power (e.g., 1kW, 2kW, etc.) is drawn to supply loads. This decreases the total building demand by the specified battery power. The algorithm then checks for preference setting violations and operates load according to their priorities so that the specified demand limit is maintained. During periods with no DR event, loads are allowed to operate normally. If the battery SOC is full when there is no DR event and there is excess solar power available, then the inverter can be configured to feed the excess power to the grid.

5.2. Incentive-based DR Algorithm with Learning

Assumptions behind this incentive-based algorithm are similar to that associated with the algorithm described in section 5.1. The customer is considered to have a contract with the utility to respond to DR signals. This DR signal from the utility specifies a period of time T and the demand limit, DL [205]. During this period, the building has to reduce its power consumption below the specified DL. An alternative form of this implementation is possible where the utility specifies the amount of load to curtail. This is decided by the utility based on their data on the peak load of a building and can be written in the contract between the utility and the customer. If the historic peak load of the building is also known to the DR algorithm, then the amount of power to curtail can also be converted to a DL by:

DL = Historical peak load – amount of power to curtail.

So, the problem can be simplified to a scheduling problem where the total power consumption of the building has to be kept below the DL during DR event. If storage resources are available, then DL can be considered as DL + Ps(t), as the available power from storage can be used to serve loads.

The algorithm proposed here uses Q-learning for HVAC loads to reduce consumption during DR event through predictive pre-cooling/pre-heating. For other loads, the algorithm uses a mixed integer liner programming based optimization to find out the operation states with least user discomfort.

5.2.1. Algorithm for HVAC Loads

The power consumption of HVAC can be minimized during DR event, through pre-cooling/pre-heating before the DR event and operating at a state of minimum consumption during DR event.

Let us consider the following:

$T_{preferred}$ = Preferred set-point of user learned through RL

T_{pre} = Pre-cooling/pre-heating set-point

T_{DR} = Set-point during DR event

If the room temperature can be brought to T_{pre} before the start of the DR event, and during the DR event the set-point is set to T_{DR}, then during the DR event, the temperature is allowed to change by only the environmental factors from T_{pre} to T_{DR}.

For colder days, when heating is used, $T_{pre} > T_{DR}$, and the room temperature is allowed to drop during DR event.

For warmer days, when cooling is used, $T_{pre} < T_{DR}$, and the room temperature is allowed to rise during DR event.

In both cases, the consumption of the HVAC unit is minimum during the temperature change due to environment. This helps to reduce the consumption from HVAC units during DR event.

The RL agent can be manipulated to follow the pre-heating/pre-cooling schedule by temporarily changing the energy and comfort reward values for pre-DR, DR and post-DR periods. If the rewards are heavily biased for changing the state from T_{pre} to T_{DR} during the DR event, the Q-learning agent selects the pre-cooling/pre-heating schedule in order to maximize total rewards.

To avoid discomfort before and during the DR event, the values of T_{pre} and T_{DR} are selected based on the maximum and minimum set-point within the comfort range of the customer previously learned by the RL.

The rebound peak after DR event can be prevented by selecting different DR event ending times for different HVAC units, so that they do not start consumption at the same time after DR event.

5.2.2. Algorithm for Lighting and Plug Loads

For lighting loads, the power consumption can be reduced by reducing the brightness level below the preferred level. The comfort rewards keep decreasing as the brightness level is lowered. Same is true for plug loads, but in their case they only have two states ON and OFF.

Therefore, it is an optimization problem, where the constraint is:

$$\sum_{j} Pj(t) \leq DL$$

Where, $Pj(t)$ is the consumption of j-th load at time t.

The cost function of the problem is the total discomfort. In other words, the optimal solution is to maximize $\sum_j r_{c,j}(t)$ where, $r_{c,j}(t)$ is the comfort reward of j-th load at time t. This a convex optimization problem and can be solved by mixed integer linear programming.

The solution is implemented by the control agents to keep the total consumption always below demand limit, while maximizing total user comfort rewards, or in other words, minimizing total user discomfort.

At the end of the DR event, the algorithm goes back to the reward based control learnt by the agents. The optimization algorithm is run at a separate DR agent and the decisions are communicated back to the device agents.

5.3. Incentive-based DR Algorithm for Multiple Buildings

The algorithm proposed in this chapter considers the demand response scenario in a bigger perspective of a community of buildings so that demand response objectives can be met through coordination of multiple buildings. These buildings can be both commercial and residential. Before delving deeper into any multi-building demand response algorithm, it is crucial to acknowledge the fact that each building has its own set of energy efficiency goals and consumption requirements, therefore each building in a multi-building coordination algorithm is essentially an individual player with a target to maximize its own benefits from the demand response algorithm. These players can be cooperative or competitive based on the demand response scenario, and they can communicate and run distributed algorithms to reach the optimal solution for individual and community DR goals. In this dissertation, a cooperative incentive-based algorithm is proposed.

This algorithm assumes that a group of buildings sign up with the utility for an incentive-based program, where utility can send demand limit or curtailment signals during peak load period. Instead of the individual demand response scenario, where each building is provided a demand limit by the utility, in this scenario, it is considered that the utility sends a combined total demand limit for a group of buildings. Let us consider this combined demand limit as DL_{total}.

The utility may send the same signal to all buildings with the total demand limit. Then the buildings act upon this signal by running the coordinated algorithm. The algorithm is described step by step below:

Step 1:

The first step in this algorithm is to find the demand limit potential at each building. This is the inverse problem of the one described in section 4.4, where a demand limit was given to find possible scheduling. An important consideration at determining this is the possibility and amount of demand restrike after the DR event is over. For multi-building scenario this issue is more pronounced as demand restrike of multiple building at the same time can amount to a very high peak of demand restrike, which can be stressful to the power system. Therefore, the demand limit potential at each building is found by setting the constraint of maximum allowable demand restrike peak.

To calculate the demand limit potential for a residential building, the load models from [125, 205] can be used, and critical load models can be found from [206]. This load models are used to first simulate the base case scenario, and then simulating the total load restricted within a demand limit. Considering fixed priority for schedulable loads, the demand limit can be decreased in steps from peak load and then in each step the demand restrike is calculated. Demand limit potential is the value of demand limit which causes the demand restrike to be marginally below the maximum allowable demand restrike value. The same methodology is followed for commercial buildings, but in this case, building load models developed in Virginia Tech can be used.

So, a demand limit potential DLPi is found for each building i, so that demand restrike peak DRPi < DRPmax (maximum allowable demand restrike for the building). To simplify the problem, the demand limit potentials can be considered to have been determined by utility using historical data and customer negotiation and written in the contract.

Step 2:

Now, the demand limit potentials of all buildings are combined to check if it is sufficient to meet the DR limit set by the utility. Two cases may happen:

Case A: Total demand limit potential is more than provided demand limit, $\boldsymbol{\sum_i DLP_i \geq DL_{total}}$:

In this case, as the combined demand limit potential of all buildings is higher than the required demand limit, the algorithm divides the total demand limit to buildings by:

$$DL_i = \frac{DLP_i}{\sum_i DLP_i} . DL_{total}$$

Hence, each building is given a part of the total demand limit as DLi by ratio of their demand limit potential. Then, each building uses their own demand limit to run their own algorithm as described in section 5.2.

Case B: Total demand limit potential is less than provided demand limit, $\boldsymbol{\sum_i DLP_i < DL_{total}}$:

In this case, the total demand limit potential is not sufficient to meet the demand curtailment need set by utility. This case can happen in case of Emergency Demand Response (EDR) scenario, where utility may have to request more load curtailment. This situation should also be written in the contract between the utility and the customer with proper incentive opportunities. In this case, dividing demand limit to buildings is not suitable. The algorithm proposed in this case is a cooperative coordination based algorithm using pre-defined priorities of the loads in all buildings. For notation, $p_{ij}(t)$ will be used to denote the global pre-defined priority factor of load j in building i at time t. Then, the reward values are calculated by:

$$r_{ij}(t) = r_{j,i}(t).\, p_{ij}(t)$$

Where,

$r_{j,i}(t)$ = learned comfort reward for load j in building i at time t

$r_{ij}(t)$= adjusted comfort reward based on pre-defined priority factor

This scaling factor is used to signify comparative importance of loads in building with respect to others. For example, a lighting load in a hospital building will have a higher priority factor than a lighting load in a nearby office building.

After the reward values have been updated to a global set of reward values, the algorithm described in section 5.2.2 is implemented considering all the loads in all buildings in the same picture. The optimization solution is then conveyed to each building using the multi-building communication framework, and the decisions are implemented by each control agent. The priority factors ensure that discomfort is lower for building loads with higher priority.

6. Case Study

The chapter presents the case studies for validating the software system and the demand response algorithms proposed in this dissertation. The case studies are designed with progressively increasing complexity, so that case studies are designed to validate component algorithms at different steps of the gradual development of a large and complex algorithm.

6.1. Case Study 1: Validation of Learning Algorithms for Energy Efficiency

This case study presents the validation of the core RL algorithm for lighting agents. It shows how user feedback can train the agent about comfort preferences.

6.1.1. Case Study Assumptions

The case study is conducted on a simulated model of a lighting load. The lighting load model has been validated by the experimental brightness and consumption data from WattStopper lighting with LED lights, as shown in chapter 4. For this case study, the following assumptions are made:

1. Occupancy sensor is available; hence, occupancy can be accurately considered.
2. The zone occupied hours are from 6:30 AM to 6:00 PM.
3. The initial schedule is pre-programmed into the software.
4. The customer provides feedback anytime the brightness level goes below comfort level. The customer adjusts it to preferred brightness level.

6.1.2. Case Study Results

The results are evaluated on a case by case basis. The following are the cases and the corresponding results:

Case 1: Initial Schedule with no training:

The initial schedule is considered to be 100% brightness level during occupied hours and 0% during unoccupied hours. So, the brightness level throughout the day looks like Fig 6.1.

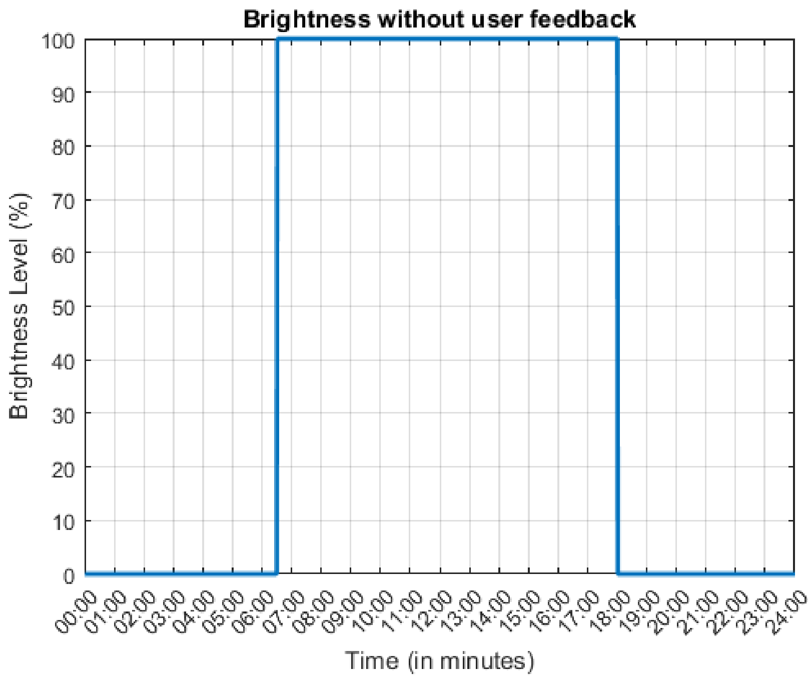


Figure 6.1. Brightness level without learning.

The same results are found if the energy comfort balance factor, $A_{ec} = 0$. This prevents the agent from exploring, and keeps is satisfied with the initial schedule of 100%, as it starts with some initial reward values, and energy reward is completely ignored.

Case 2: Training with user feedback:

In this case, the energy balance factor, $A_{ec} = 0.5$, to provide equal weight to both energy and comfort rewards. As the initial comfort reward is not high enough to prevent exploration, the agent explores lower brightness states with higher energy rewards. Fig 6.2 shows the training session.

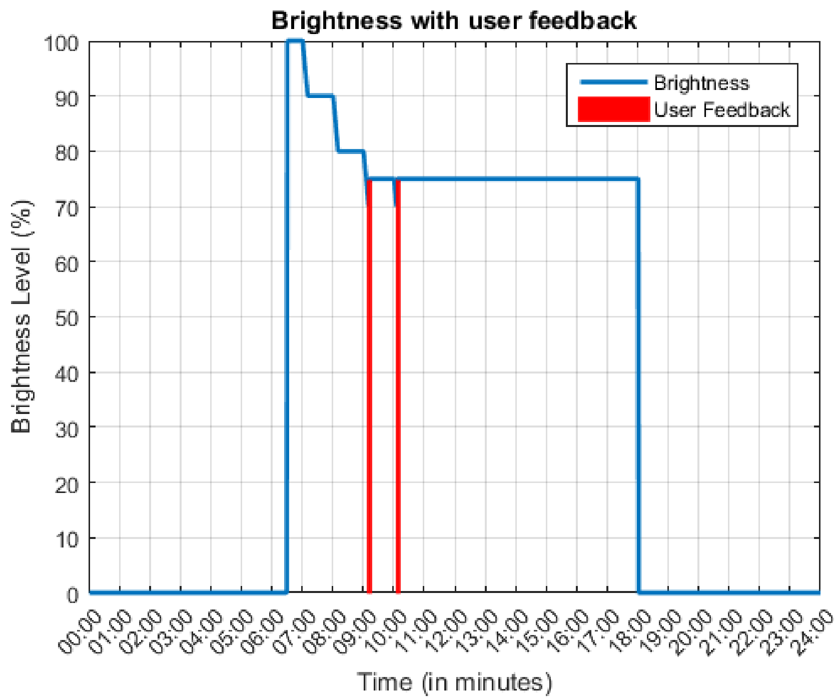


Figure 6.2. Brightness level with feedback (User preference=75%, $A_{ec} = 0.5$).

As can be seen from the figure, the agent explores at every clock hour. It only changes 1% brightness level per minute for a total of 10% change in 10 minutes at the start of each hour. This exploration strategy is chosen in order to prevent user inconvenience. The slow rate of change avoids unwanted shock for user. The 10% limit is to let the user get accustomed to the changed brightness level, so that they can provide feedback only if it is uncomfortable, and not just to stop changes.

In the figure, the agent explores by 10% every hour until 9:00 to 10:00 AM, when the brightness level has reached 70%. The simulated user feels discomfort below 75%, therefore they adjust it back to 75%, and this negative feedback updates the satisfaction reward values. The red bar shows the feedback. Although in practical case, after one such feedback, the next exploration should not be done on the same day, in this case, for a simplified theoretical study, the agent is allowed to make exploration in the next hour itself. Therefore, it tries to change the brightness again next hour, and met with another feedback from user. After these two feedbacks, the rewards are updated in a way that the agent does not explore further, and has reached the optimal state. Fig 6.3 shows the updated rewards.

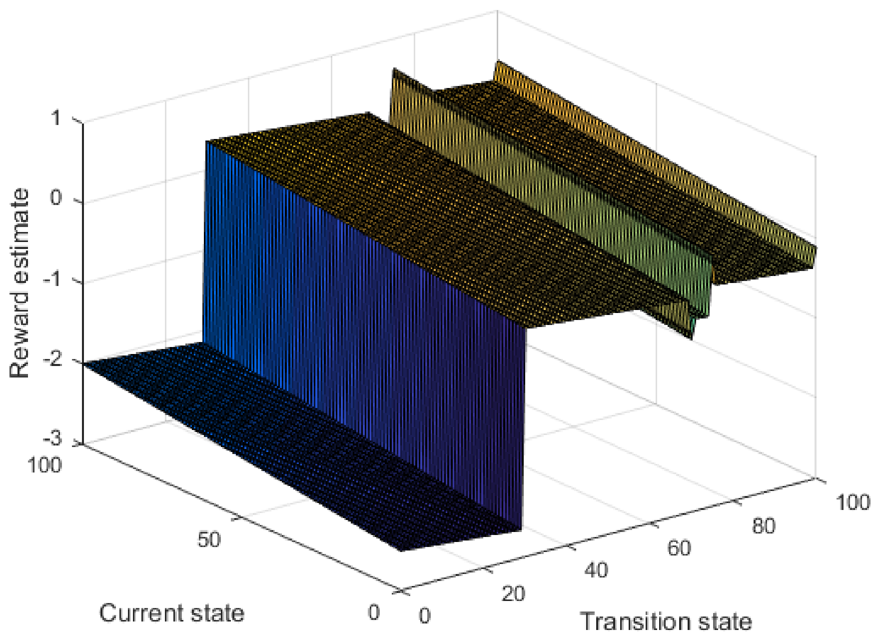


Figure 6.3. Brightness level with feedback (User preference=75%, $A_{ec} = 0.5$).

As can be seen from Fig 3, it is a surface plot of reward matrix. It plots the rewards for going from one state to another. The negative values of reward below 30% is pre-set to avoid the agent from ever exploring in those states, as they are below the standard minimum brightness level. From 30% to 100%, the base slope is coming from the energy reward values which show that rewards are higher for lower brightness level. The updated values around 75% are due to training of satisfaction reward values. As the reward slope below 75% is higher than 0.5, the agent is prohibited from exploring further.

Finally, Fig 6.4 shows the operation before and after training.

Figure 6.4. Brightness level before and after training.

As can be seen from the figure, the optimal operation saves roughly 25% of energy, while staying within user comfort level. This value of energy savings is totally dependent on each user's preference.

Case 3: Training with different A_{ec}:

In this case, an analysis is performed on the effect of A_{ec} on the learning. If $A_{ec} = 0$, the agent does not explore at all, as mentioned earlier. The other extreme is $A_{ec} = 1$, when the agent completely ignores user comfort. The result is the agent prefers the highest reward from energy, which comes from brightness level of 30%, without violating the minimum standard, as shown in Fig 6.5.

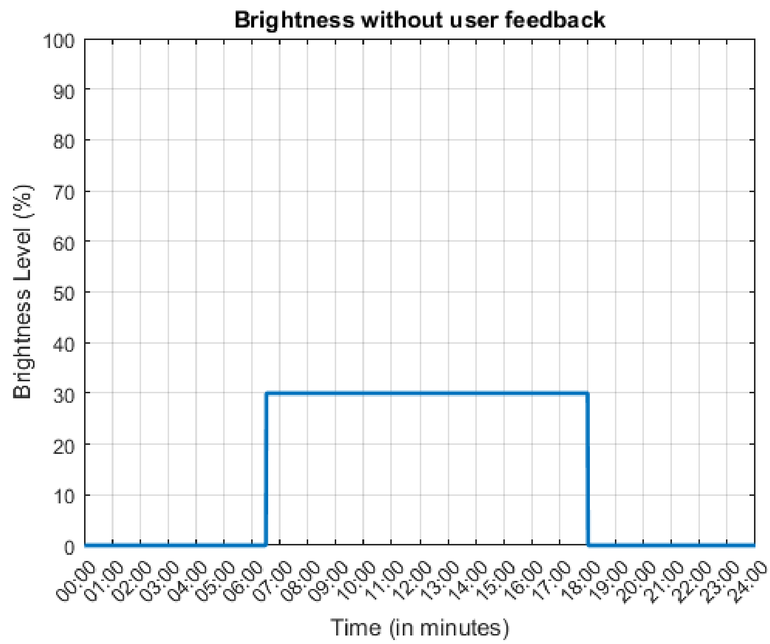


Figure 6.5. Brightness level avoiding user feedback through $A_{ec} = 1$.

As the A_{ec} is increased beyond 0.5, more emphasis is given on exploration and less on user comfort. Hence, it takes more feedbacks to adjust the reward values. Fig 6.6 shows such a case with adjusted satisfaction reward weights and $A_{ec} = 0.6$.

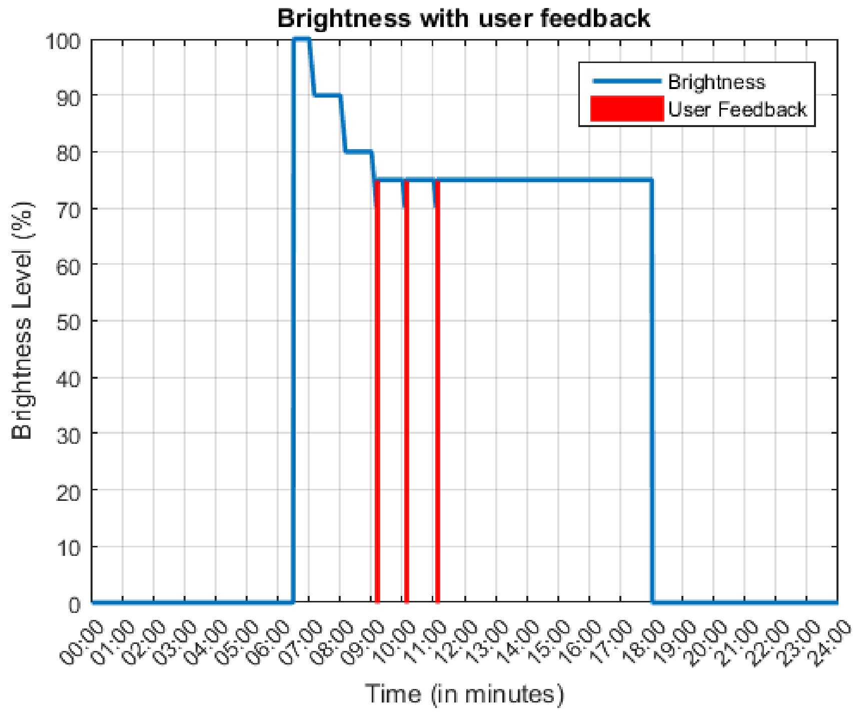


Figure 6.6. Brightness level with user feedback and $A_{ec} = 0.6$.

As can be seen from Fig 6.6, the same algorithm requires three user feedbacks instead of two in the previous case to stop exploration. This shows how the weights can affect the learning. For the case studies in the DR section, A_{ec} is considered to be 0.5 to ensure balance between exploration and exploitation.

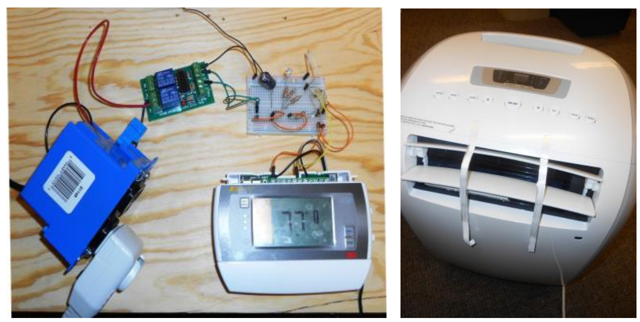
6.2. Case Study 2: Validation of Priority Based DR Algorithm in Smart House

6.2.1. Case Study 2a: BEM with Smart Thermostat Control

This part presents the hardware demonstration case studies of the BEM system with smart thermostat to showcase the applicability of the system to address demand limit and customer preference.

6.2.1.1. BEM System Setup

Figure 6.7 shows the smart thermostat setup in the laboratory environment. A portable AC unit was used to be controlled by the thermostat. A USNAP module was used with the thermostat for Wi-Fi connectivity. The thermostat was configured to be connected to the wireless local area network of the lab environment. In an actual household implementation, a router can be used instead to provide Wi-Fi connectivity to the thermostat. The BEM unit consisted of a laptop with internet connectivity that runs the BEM program. The BEM unit could communicate with the thermostat through the Internet. A local area network implementation is also possible if the Internet is unavailable for a certain household.



(a) (b)

Figure 6.7. The BEM system setup in the laboratory: (a) the smart thermostat with its control circuit, (b) the portable AC unit controlled by the thermostat.

6.2.1.2. Case Study Assumptions

6.2.1.2.1. House & Load

The case studies are based on the parameters of a hypothetical average single family house with a size of 2500 sq. ft. [207]. Only four power-intensive loads are controlled by the BEM system: a water heater, an AC unit, a clothes dryer and an electric vehicle. Critical loads are not controlled and their power consumptions are taken from the RELOAD database [206]. In the laboratory environment, due to unavailability of the actual loads of WH, CD and EV, representative loads were used. WH and EV were replaced by baseboard heaters and CD was represented by a hair dryer, due to similar load characteristics. The power consumption of these representative loads used in the laboratory are multiplied by corresponding scale factors to match the power consumption of actual loads.

6.2.1.2.2. Load Priority

The customer priorities for loads are assumed to be as follows: WH has a higher priority than AC, AC has a higher priority than CD and CD has a higher priority than EV.

6.2.1.2.3. Comfort Level Setting

The comfort settings are assumed to be as follows: hot water temperature has to be maintained between 110°F-120°F. Room temperature has to be maintained within ±2°F of the set-point temperature, which initially by default is 76 °F (can be changed to 80 °F by the BEM algorithm). Both the minimum ON time limit and OFF time limit for the heating coils of the clothes dryer are set to be 15 minutes, and CD must finish its job by midnight. For EV, minimum 15 minutes EV charging time is specified before it can be shed, and it has to finish its job by 8am in the morning.

6.2.1.2.4. Variation in Temperature

The laboratory environment for this experiment was a controlled-temperature environment, i.e., the temperature of the room was centrally controlled to be fixed at a certain set-point temperature (i.e., 74°F). This limitation was overcome by using a space heater with constant heating in the room. The distance of the heater from the thermostat and the speed of the fan of the heater were adjusted to represent temperature rising profile similar to the real profile for the hypothetical household on a hot summer day.

6.2.1.3. Actual AC Measurement Data During a No-DR Event

The AC unit was controlled by the BEM system with the smart thermostat for 3 hours from 5pm to 8pm without any DR event to see the actual power consumption of the AC and the room temperature profile. Figure 6.2 shows these results.

As can be seen from the figure, the portable AC unit consumed power in two steps when it was ON. This is because, when the AC was turned ON by the thermostat, first the fan of the AC unit was ON, which consumes low power, and then after some minutes the cooling compressor was ON, increasing the power consumption. The maximum power consumption of the AC was around 585 W. A scale factor was used to scale-up this value to match the actual power consumption of the AC unit for the hypothetical household. Also, it can be seen in Figure 6.8 that the temperature increased in steps. This is due to the limitation of the thermostat, which could monitor and provide temperature data in 0.5 °F steps.

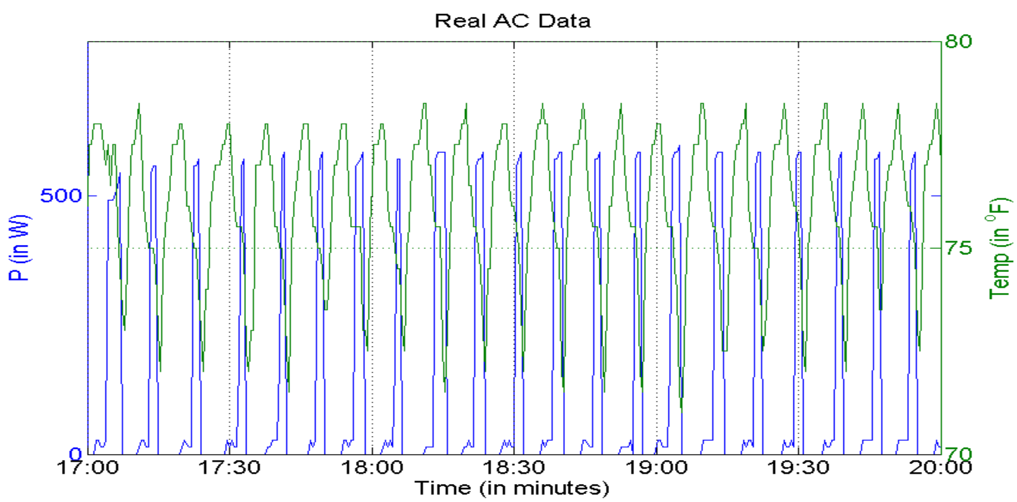
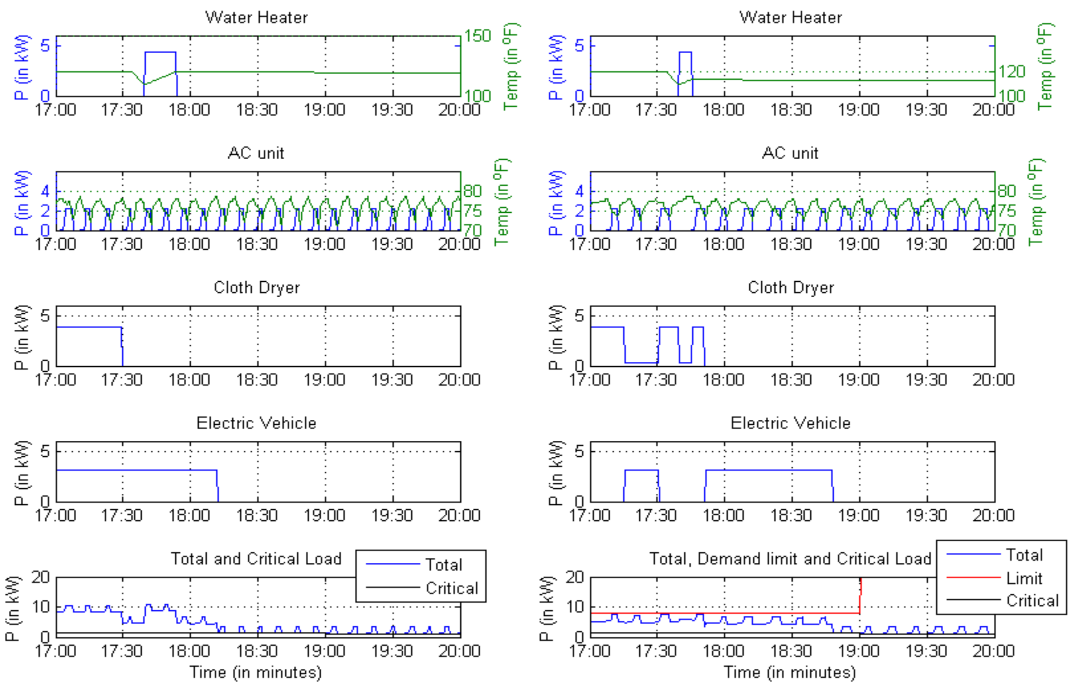


Figure 6.8. Actual AC unit measurement data.

6.2.1.4. Case Scenarios

To see the effect of demand response in household load control by the BEM system, the DR event was considered to be scheduled for 2 hours from 5pm to 7pm. A demand limit of 8kW was considered within the DR event period. This is the limit that the total household consumption should not exceed. The BEM program was run in real time for 3 hours from 5 pm to 8pm for two different cases. Case 1 considered no DR event. Case 2 was with the DR event as mentioned earlier. Figure 6.9 shows the comparison between temperature profiles and power consumption of different appliances for the two case scenarios.



(a) (b)

Figure 6.9. Demand response demonstration results for a hypothetical 2,500 sqft home with scaled-up power consumption: (a) without a DR event; (b) with a DR event between 5pm and 7pm.

<u>Case 1: No DR event:</u> Figure 6.9(a) shows the power consumptions of different appliances along with temperature profiles from 5pm to 8pm without any DR event. As there is no demand limit, the BEM system does not have to shed any load. Thus all appliances are turned ON and OFF based only on their preset comfort level specified by a homeowner. This causes the peak consumption at 5:40pm to 5:48pm, when all the appliances were ON. The thermostat operates at the set-point temperature of 76 °F and this set-point temperature is maintained throughout the 3-hour period of the experiment, as no demand limit existed during that time.

<u>Case 2: With DR event:</u> As can be seen in Figure 6.9(b), the DR event was imposed between 5pm and 7pm with a demand limit of 8kW. During the DR event, from 5:40 pm to 5:48 pm, the WH was ON to bring the water temperature back to the specified limit. This caused the BEM to change the set-point temperature of thermostat to 80 °F. This is why during the time WH is ON, AC was kept OFF by the thermostat, even though room temperature increased up to 80 °F. The set-point was brought back to 76 °F when the WH turned OFF and AC was next in priority. Hence thermostat

turned the AC ON at 5:48 pm and brought the room temperature back to the comfort limit. The total power consumption was always within the specified demand limit of 8kW. This shows that the temperature set-point control of thermostat can be used to cap the highest allowable room temperature and thus further address customer preference while keeping the total household power consumption within a specified limit.

6.2.2. Case Study 2b: Validation of BEM in Smart House

The BEM algorithm discussed in section 5.1 was validated in field implementation, by designing realistic case study scenarios in the smart house located at Yildiz Technical University (YTU) – Davutpasa campus in Istanbul, Turkey. This was a collaborative project between YTU and Virginia Tech Advanced Research Institute. The data presented here came from experiments conducted by teams from both YTU and VT-ARI.

6.2.2.1. Smart House and Its Integration with the BEM Algorithm with Further Modification

The test-bed was a pilot smart house project at Yildiz Technical University (YTU) Davutpasa Campus, Istanbul, Turkey, which is financially supported by Istanbul Development Agency (ISTKA) [208].

6.2.2.1.1. Loads in the YTU Smart House

The 35 square meter smart house prototype was built inside a research laboratory building at the YTU Davutpasa campus. The prototype consists of a living room with a kitchen, a study room and a bathroom. It is equipped with typical household appliances. An electric vehicle (EV) charging station is also a part of the smart house. Note that both solar and wind electricity available from the building roof-top can be integrated into the smart house, together with a power conditioning system and a battery bank. The smart house network is shown in Figure 6.10.

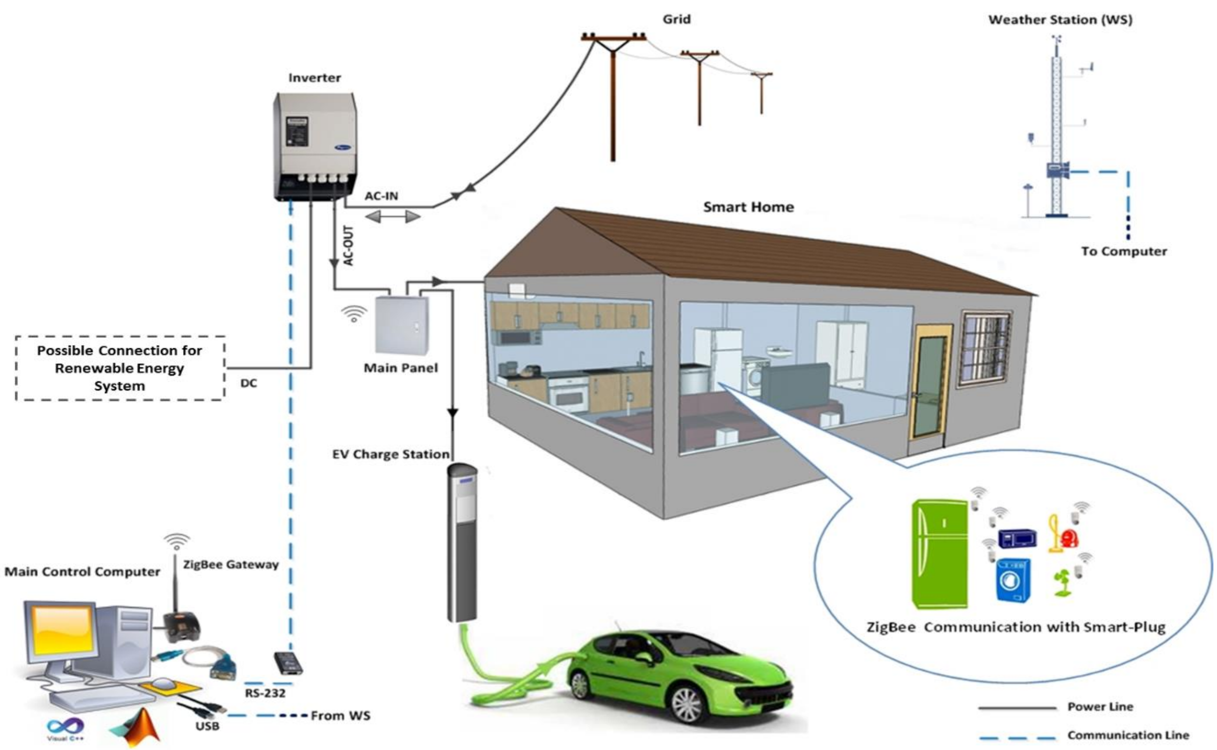


Figure 6.10. General concept of the YTU smart home project.

6.2.2.1.2. 4-Noks® Smart Plugs for Load Monitoring and Control

In the smart house, 4-noks® smart plugs are used as load control devices [209]. Each appliance is connected to an addressable 4-noks® smart plug that uses ZigBee wireless communication to allow its monitoring and control from the central computer. While power-intensive loads are to be controlled by the smart plug, all critical loads are only monitored. In addition to smart plugs, smart temperature sensors are also installed in the smart house, which provide room temperature data to the central computer. The EV charging station can also be turned ON/OFF from the main computer using a 4-noks® ZR-HMETER.D-M [210].

6.2.2.1.3. Integration of the BEM to the YTU Smart House

To allow the BEM algorithm to control appliances in the smart house, the improved BEM algorithm discussed before is embedded in the main computer that communicates with all other devices. The central computer collects electrical consumption data from all appliances and implements control decisions by communicating with selected smart plugs that provide necessary interface between appliances and electrical outlets.

In order to enable the implementation of BEM algorithm in a smart house, some other modifications in the algorithm were necessary to control the smart house loads and smart plugs, as discussed below.

Modification of the BEM Algorithm Taking into Account Deferrable, Non-Interruptible Loads

Some selected loads in a smart house environment are deferrable/non-interruptible loads, which include a washing machine and a dishwasher. These loads are not suitable for interruption by the BEM during a DR event. This is because, for some washing machine and dishwasher models, if their operation is interrupted, the whole washing cycle must be restarted. Hence, the BEM algorithm has been modified to defer the start time of load cycle in this category until the end of a DR event.

Modification of BEM Algorithm Taking into Account Data Collection of 4-noks® Smart Plugs

In the laboratory experiments, each load controller was sent data requests once every minute, hence if there was a communication failure during a single communication request/response, it resulted in missed data for that particular minute only. On the other hand, in the YTU smart house, data was collected from the 4-noks® smart plugs several times during one-minute intervals. Hence, even one successful communication with a smart plug during a minute was sufficient for BEM to make decisions. Due to this fact, the BEM algorithm was further modified to ignore communication failures with a smart plug during a given minute, if there was at least one successful communication with that smart plug during that minute.

6.2.2.2. Design of Realistic Case Scenarios

In this subchapter, assumptions and data used in the formation of case studies are discussed, including surveys of appliance usage, customer preferences and load priorities, assumptions for a DR event and load profiles of different appliances.

6.2.2.2.1. Survey of Appliance Usage in Turkish Households

In order to design valid case studies, the consumption profile of typical Turkish customer building is required. A survey was conducted among 10 Turkish customers to build such a profile. Customers were asked to provide the time and duration of usage of each appliance, including water usage, on a regular work day. All participants had working hours from 09:00 to 17:00 regardless of the season. Based on the cumulative survey results, average usage profiles of all smart house appliances were generated to represent typical appliance usage profiles of Turkish customers. Figure 6.11 depicts typical usage times and durations of selected appliances on a regular weekday in summer and winter.

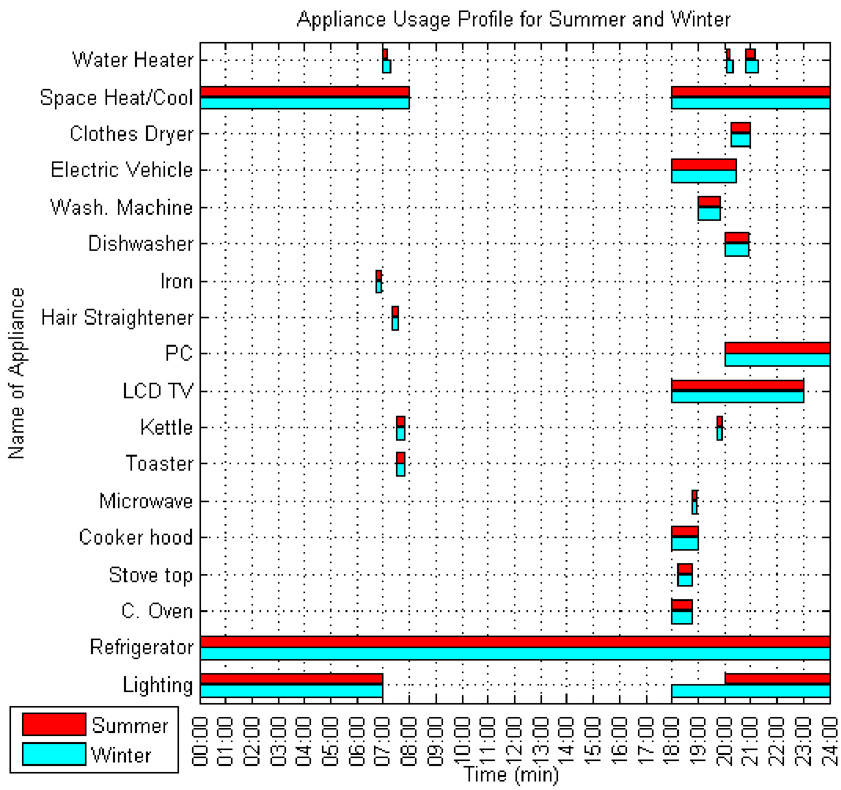


Figure 6.11. Appliance usage time for an average Turkish household, determined from the survey, for summer and winter.

Notice a minimal appliance usage during customers' absence from home (08:00-18:00 including travel time to and from work), and sleeping hours (midnight to 06:00) where partial lighting, i.e. porch/corridor lighting, is ON. Most appliances are used either in the morning from 06:00 to 08:00, or in the evening from 18:00 to midnight. The hot water usage profile from the survey was used to determine the operation of an electric water heater during both summer and winter seasons. Lighting was used for shorter duration during summer due to extended daytime.

6.2.2.2.2. Survey of Customer Preference and Load Priorities

A separate survey was conducted on the same group of customers to determine typical load priority and preference settings for the four power-intensive loads to be controlled by the BEM. For priority settings, most customers responded with the following priority sequence: WH>AC>CD>EV. The average preference setting for a water heater (WH) from the survey was to maintain the hot water temperature within 43-46^{o}C (~110-115^{o}F) range. And the room temperature set point was preferred to be 23^{o}C (~74^{o}F) during winter and 24^{o}C (~76^{o}F) during summer. The customer preference was for the clothes dryer (CD) to finish its job by midnight. Also, both its maximum OFF time and minimum ON time was specified to be 30 minutes. An electric vehicle (EV) was preferred to finish its job by 08:00 in the morning with a minimum charging time of 30 minutes before any interruption could occur.

6.2.2.2.3. Assumptions for DR Events

To design the demand limit and duration of a DR event for the case studies, the total household consumption without DR event was generated using the appliance usage profiles from the above surveys. Figure 6.12 shows the total household consumption during morning and evening hours for summer and winter when customers are at home.

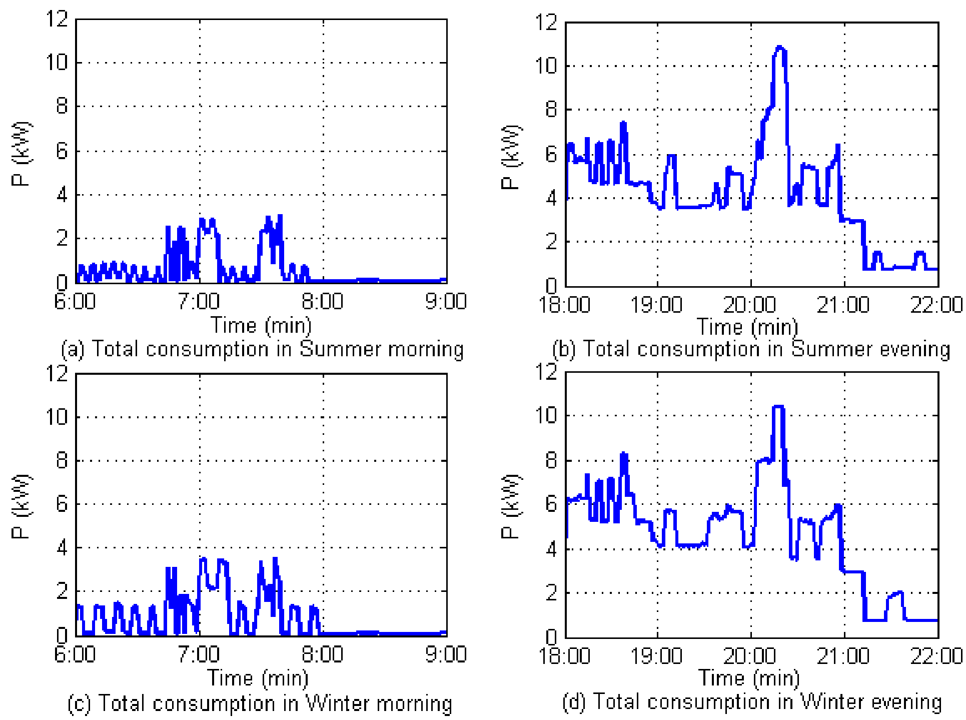


Figure 6.12. Total household consumption profiles without DR for summer and winter.

The morning peak demand was observed between 06:00 and 08:00 both in the summer and winter, with the summer peak being 3.6kW and winter peak being 3.8kW. The evening peak demand was observed between 18:00 and 21:00, with a summer peak of 10.8kW and winter peak of 10.2kW. Morning peak loads are much lower than evening peak loads, hence a morning peak period is not a candidate for a DR event. Since typical evening peaks occur between 18:00 and 21:00, this study considers that a local utility imposes a demand reduction target to limit the peak demand of residential customers from 18:00-21:00.

Both in winter and summer, a demand limit of 6.7 kW (or, approx. 33% load reduction) is imposed during a DR event. This value of demand limit is selected for the case studies based on the simulation results to avoid the rebound of peak demand (i.e., demand restrike) during an off-peak period after a DR event ends.

6.2.2.2.4. Load Profiles

Water Heater

A space heater with the same power rating as a typical electric water heater in Turkey was used to represent the electric water heater in the smart house. The measured load profile of this space heater is shown in Figure 6.13(a). It was placed outside the smart house during the experiment so that the additional heat generated did not affect the room temperature inside smart house. It is considered the highest priority load.

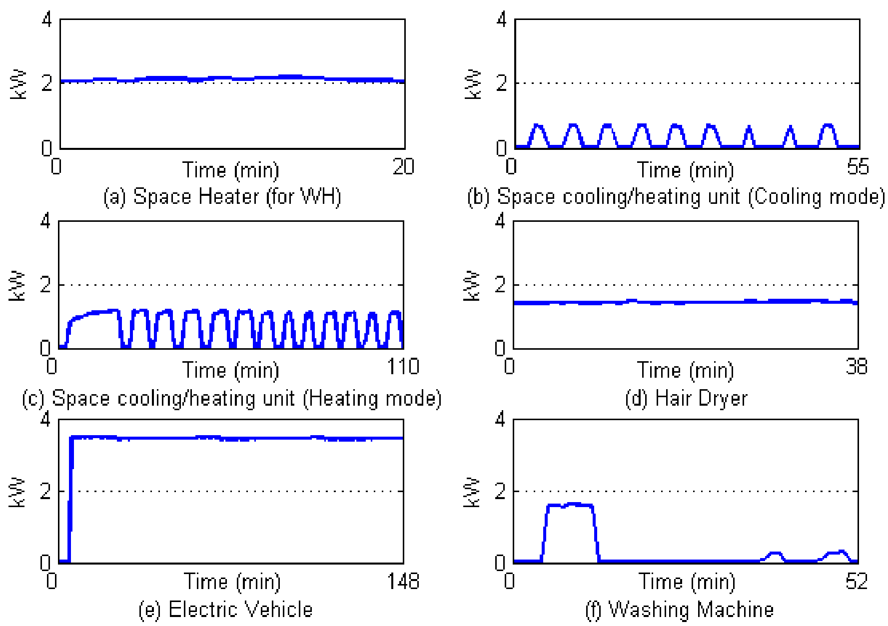


Figure 6.13. Measured load profiles of selected appliances.

Space Cooling/Heating Unit

The space cooling/heating unit was operated in the heating mode for winter case studies with the room temperature set point of 23°C (~74°F) and a dead-band of 1°F. For summer case studies, it was operated in the cooling mode with the set point of 76°F (~24°C). From the measured load profiles in Figure 6.13(b) & 6.13(c), it can be seen that the unit consumed 1.14 kW in its heating mode and 0.7 kW in its cooling mode. The space cooling/heating unit is considered the second highest priority load for the case study demonstration.

Clothes Dryer

As a clothes dryer is unavailable in the YTU smart house environment, it was represented by a hair dryer to conduct the case studies. The hair dryer has a similar load profile (Figure 6.13(d)) as a clothes dryer, but has lower power consumption. Hence, a scale factor of 2.0 was used to scale-up the consumption of the hair dryer to represent the consumption of a clothes dryer. The unit is set with a load priority just below the space cooling/heating unit.

Electric Vehicle

Due to the limited availability of EV for the experiments, a one-time measurement was conducted to determine the power consumption of an EV at the YTU smart house. The EV was charged from 45% to 85% state-of-charge and it had an almost constant consumption throughout the experiment. The recorded EV consumption profile (Figure 6.13(e)) was used for case studies instead of its real-time operation. The EV is considered the lowest priority load in the house.

Deferrable, Non-Interruptible Loads

These loads, for example, a washing machine, are not interrupted during their operation, but will be deferred if a homeowner starts their operation during a DR event. Figure 6.13(f) shows the measured power consumption profile of a washing machine.

Table 6-1 summarizes the power-intensive load representations, their power consumption (kW), scale factors, and their priority and preference settings used during the case study demonstration.

Table 6-1. Load Representations, Scale factors, Priority and Preference Settings

Load	Actual load used in smart home demonstration	Actual load power consumption (kW)	Scale factor used	Load priority	Preference settings
Water heater	Space heater	2.1	1	1	Hot water temperature: 110-115^{o}F
Space cooling/ heating unit	Space cooling/ cooling unit	1.14 (Heating); 0.7 (Cooling)	1	2	Room temperature: summer: 76^{o}F($\pm$1^{o}F); winter: 74^{o}F($\pm$1^{o}F)
Clothes dryer	Hair dryer	1.45	2	3	Finish job by midnight; Max OFF/Min ON Time: 30 min
EV charger	Recorded profile	3.3	1	4	Finish job by 8 AM; Min charging time: 30 min

6.2.2.3. Discussions of Case Studies

This subchapter presents the description of case studies, basis for temperature adjustments in the smart house, results and discussions, as well as lessons learned from the field experiment.

6.2.2.3.1. Case Study Description

To observe the impact of the proposed BEM algorithm in a real-world smart house environment, four different case studies were conducted:

a) Summer season with no DR event;

b) Summer season with a DR event from 18:00-21:00;

c) Winter season with no DR event; and

d) Winter season with a DR event from 18:00-21:00.

For all these cases, the BEM program was run for 4 hours to observe the impact of DR on total household loads during the 3-hour DR event and the impact of load restrike for one hour after the DR event ends.

6.2.2.3.2. Basis for Temperature Adjustments in the Smart House

As the smart house is built inside the YTU Electrical Engineering building and all experiments (including both winter and summer case studies) were conducted in winter, the temperature profile outside the smart house had to be manipulated to represent typical temperature profiles for winter/summer cases of a real house.

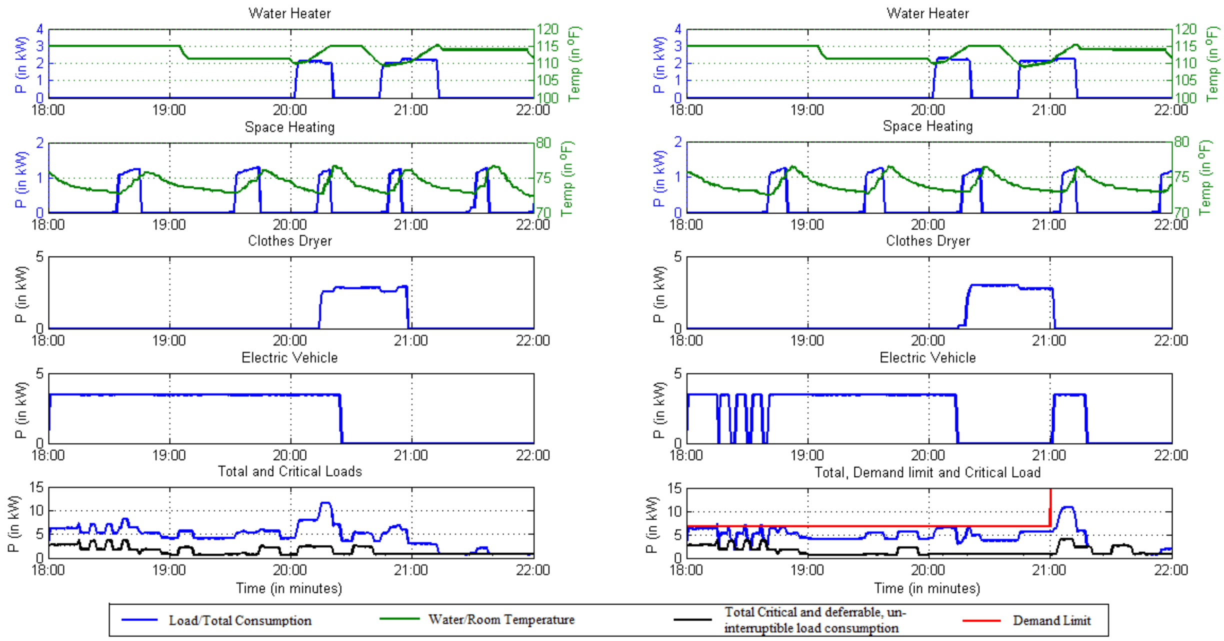
For winter cases, building zone heaters were turned OFF, and windows were left open so that the environment outside the smart house (but inside the building) represents ambient conditions during the winter in Turkey.

For summer cases, building zone heaters were kept ON, keeping the building temperature at around 24^{o}C (~76^{o}F), and an additional space heating unit was used outside the smart house to increase the ambient temperature, representing a hot summer day in Turkey. The additional space heating was kept constantly ON during the whole duration of the summer case.

6.2.2.3.3. Results & Discussions

Results of the four different case studies are discussed as follows:

Case 1: Winter case – No DR event: This case was run in the afternoon on Jan 11, 2013. This is the base case for winter season without any DR event in effect. Figure 6.14(a) shows the power consumption of different appliances. Temperature profiles for this case are also shown. Critical loads include uninterruptible loads, such as lighting, TV, refrigerator, PC, etc. Deferrable/uninterruptible loads include a washing machine and a dishwasher. As there is no DR event, all appliances are turned ON and OFF based only on their preset comfort settings as specified earlier. The space cooling/heating unit functioned in the heating mode to maintain the room temperature within ±1^{o}F around the set point of 74^{o}F. As can be seen from the power consumption profile, the operation cycle of the space cooling/heating unit is different between 18:00-20:00 and 20:00-22:00. This is attributable to the fact that there were more people present in the smart house between 18:00-20:00. This caused the duration of both the heating and cooling functions to be longer than those during 20:00-22:00. The other loads were operated according to schedule derived from the Turkish survey. The peak consumption of almost 10.4 kW occurred from 20:16 to 20:21 when all the appliances were ON simultaneously. The critical load profile is also shown at the bottom-most subplot of Figure 6.14(a), together with the total household power consumption.

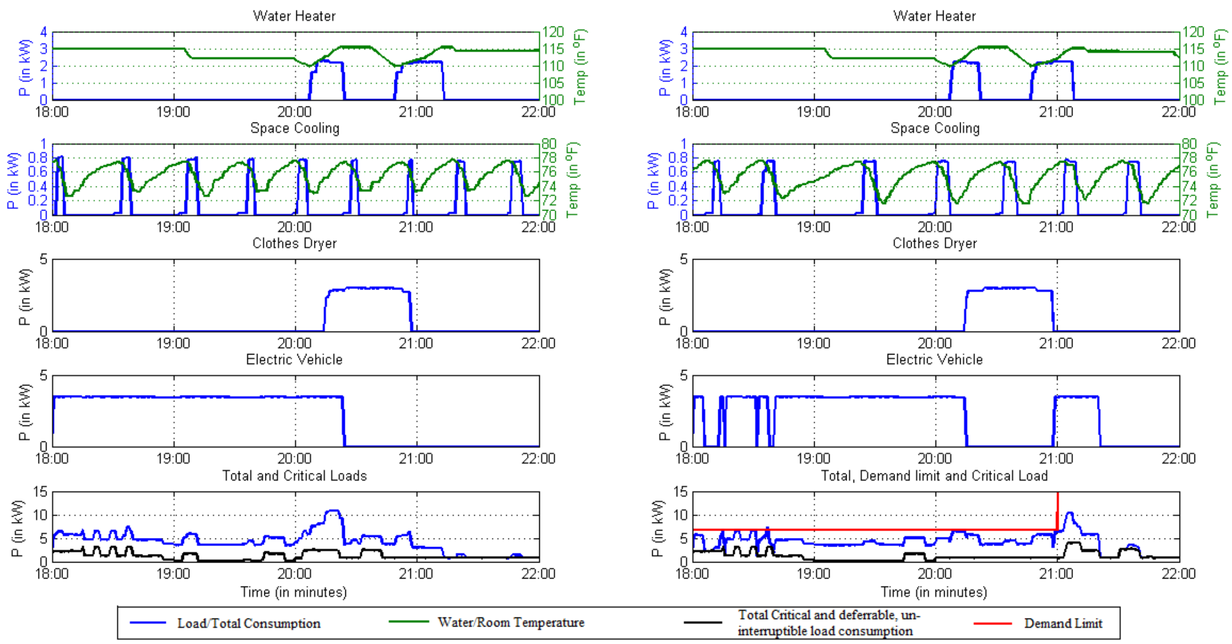


(a) (b)

Figure 6.14. Demand response demonstration results: (a) winter case without DR event; and (b) winter case with a DR event between 18:00 and 21:00.

Case 2: Winter case – With DR event: This case was run in the evening on Jan 11, 2013. As can be seen in Figure 6.14(b) – the bottom-most subplot, a demand limit of 6.7 kW was imposed between 18:00 and 21:00. During the DR event, the BEM algorithm continuously tried to maintain the total consumption below the demand limit. In order to do so, it shed lower priority loads if the total consumption exceeded the demand limit. For example, from 18:15 to 18:45, the EV was turned OFF a number of times to allow critical loads to run. It was also turned OFF from 20:16 to 21:00 in order to keep the clothes dryer ON, which has higher priority than the EV. In short, the BEM algorithm ensured that the demand limit was not surpassed at any time during the DR event. The deferrable loads: washing machine and dishwasher were deferred until after 21:00 when the DR event was over. Due to load curtailments and deferrals during the DR event, load restrike was observed for almost 15 minutes after the DR event ended, where peak consumption reached the value of almost 10.4 kW. A comparison between Figure 6.14(a) and Figure 6.14(b) clearly demonstrates the efficacy of the BEM algorithm to control total household consumption during a DR event.

Case 3: Summer case – No DR event: The base case for summer season is the case without any DR event. See Figure 6.15(a). This case was run in the afternoon on Jan 15, 2013. The space cooling/heating unit operated in the cooling mode, with a set point temperature of 76°F. As mentioned earlier, an additional space heater was turned ON to increase the temperature inside the smart house, representing summer season. Usage schedules of appliances were modified according to the result of the survey conducted. When comparing Figure 6.14 and Figure 6.15, it can be seen that the water heater was operated for shorter periods in summer cases than those in winter cases. Without a DR event, the BEM algorithm operated appliances inside the smart house to maintain preset customer comfort level. The total household consumption reached a peak value of 10.5 kW between 20:16 to 20:20.



(a) (b)

Figure 6.15. Demand response demonstration results: (a) summer case without DR event; and (b) summer case with a DR event between 18:00 and 21:00.

Case 4: Summer case – With DR event: A demand limit of 6.7 kW was applied for this case for

3 hours from 18:00 to 21:00, as shown in Figure 6.15(b). The case was run in the late evening on Jan 15, 2013. To keep the total consumption below demand limit, the BEM algorithm turned OFF and ON the EV a number of times from 18:00 to 18:45 to allow critical loads to operate. The EV was also turned OFF during the clothes dryer operation from 20:15 to 21:00. An irregularity in the operation cycle of the space cooling/heating unit was observed around 19:00. This is because the door of the smart house was open during that time, causing the additional space heater (used for representing summer case) to take longer than usual to warm up the room. Overall, the demonstration shows that the BEM algorithm keeps the total household consumption below the demand limit of 6.7 kW between 18:00 and 21:00. The washing machine and the dishwasher were deferred until the end of DR event. This resulted in 15-minute load restrike after the DR event ended at 21:00, with a peak consumption of about 10.2 kW.

6.2.2.3.4. Summary of Results

To summarize the results from the four cases discussed above, the BEM algorithm was able to successfully control the household consumption below demand limit while maintaining customer priority and comfort settings, for both winter and summer cases. Although, winter and summer cases do not offer significant differences in terms of implementation results, both are presented here for sake of completeness of the study and proof of year-round effectiveness. Another point to note from the results is: in both cases, the rebound peak after DR event was almost equal to the original peak without DR. This shows that the imposed demand limit (i.e., 6.7 kW) is the marginal value for this specific smart house and a lower value of demand limit would have resulted in a higher restrike peak than original peak. This marginal value is important to consider for a utility before choosing a demand limit for a particular household.

6.2.2.3.5. Lessons Learned from Field Implementation

From the field implementation of BEM in a smart house environment, the following issues and associated mitigation approaches are worth mentioning: issues of different data collection rates of 4-noks® and handling of communication failures and data errors.

Data Collection of 4-noks®

The 4-noks® smart plugs used in the smart house communicate with the centralized PC in a request-response type protocol. That is, these smart plugs provide sampled data in response to a request from the PC. If the communication channel was busy for any reason, a 10-second delay will be imposed before a new request is issued. Due to this reason, the number and timing of data collected by a particular smart plug within a particular minute varied highly from data within another minute and also from data collected by other smart plugs. As data is used in one-minute intervals for the analysis, this issue is resolved by averaging all data collected by a smart plug within a 60-second time window, and using the average as the representative data for that minute. There is at least one data collected during each minute by each smart plug, therefore the solution is deemed sufficient to resolve the data collection rate variation issue.

Handling of Communication Failures and Data Errors

As can be expected, communication failures and data errors with smart plugs existed during the case studies. The failure rates for smart plugs connected to loads ranged from a minimum of 0% to maximum 2.11%. The experiment shows that given the smart plug failures, the revised algorithm is able to make correct decisions based on the sampled data. This proves the robustness of the algorithm and its suitability for application in regular households that may have variety of appliances to monitor/control.

6.2.3. Case Study 2c: Validation of BEM in Smart House with DER

This section presents simulation and experimental case studies based on the algorithm described in section 5.1.3.

6.2.3.1. Smart House and Case Study Setup

The smart house environment used in this study is the same smart house at Yildiz Technical University (YTU) Davutpasa Campus, Istanbul, Turkey. The prototype is equipped with typical household appliances and an electric vehicle (EV) charging station. Solar PV (16 Kyocera panels, each rated at 210W) along with a power conditioning system and a battery bank (22kWh) is integrated with the smart house. The YTU smart house project with DER is shown in Figure 6.16.

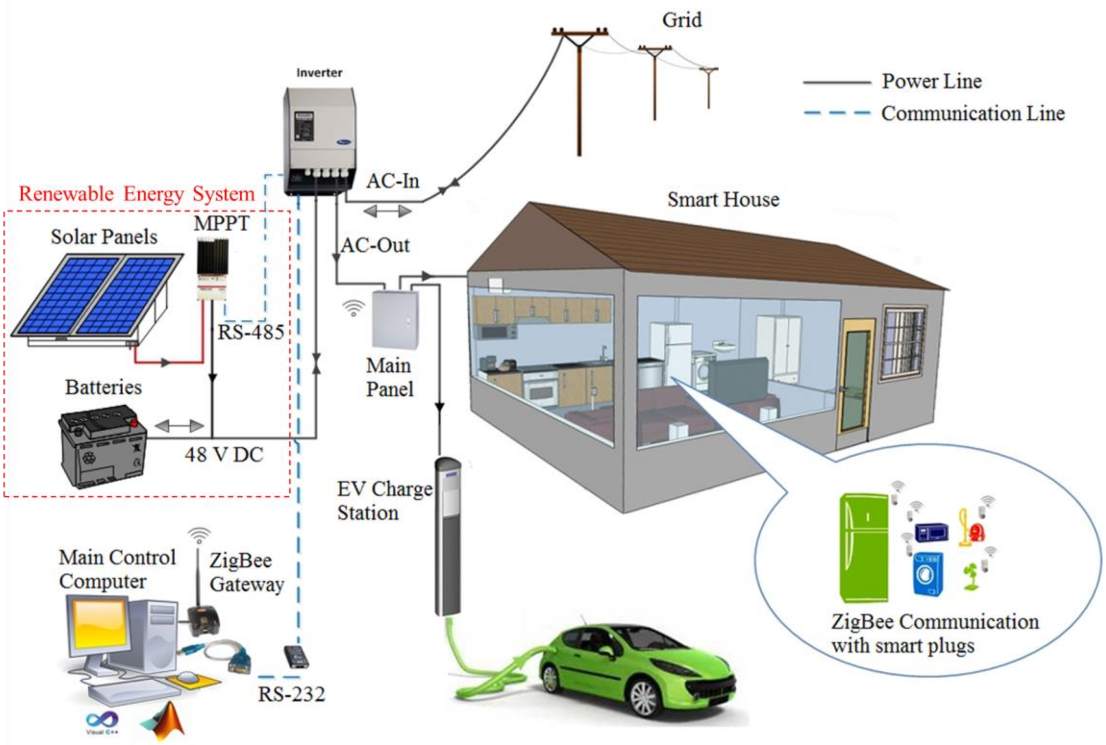


Figure 6.16. General concept of the YTU smart house project with DER.

The assumptions and load representations for this case study is the same as described in Case Study 2b in section 6.2.2.

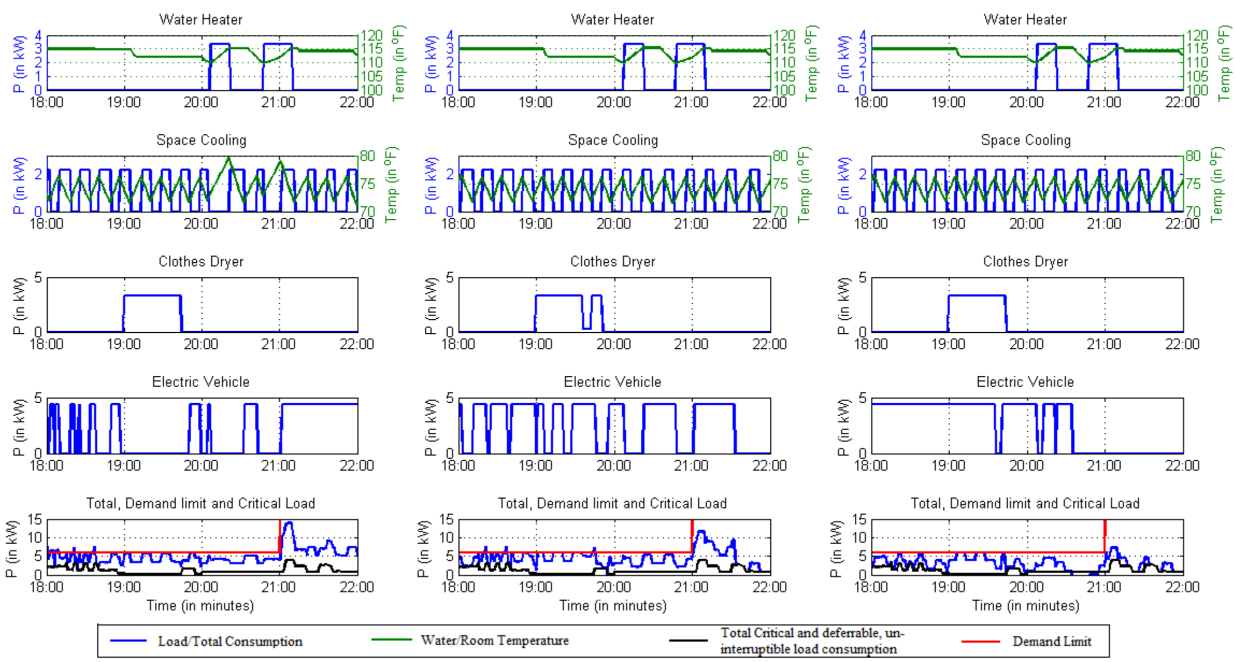
6.2.3.2. Simulation Case Studies and Results

To study effects of the modified BEM algorithm with renewable energy, the developed PV rooftop model and battery model were integrated with the BEM algorithm to run simulation case studies. These case studies were conducted to determine the effect of amount of battery power drawn during a DR event and to determine a suitable amount of battery power for real case studies.

For simulation case studies to resemble to actual scenario at the YTU smart house, the PV and battery model were adjusted to match characteristics of the YTU renewable energy system. Also, loads and their power ratings in the case studies were considered to be the same as loads in the smart house. A 6.7 kW demand limit was considered to be imposed for a 3-hour DR event from 18:00-21:00.

The BEM simulation was conducted for six different cases with different values of power drawn from the batteries (0, 1, 2, 3, 4 or 5 kW) during the DR event, 0 kW being the base case. The battery SOC was considered to be the same (i.e., full charge) at the beginning of the DR event in each case. Load priority was considered to be WH>AC>CD>EV for all cases; and a sunny summer day with clear sky for PV is assumed.

Figure 6.17(a), (b) and (c) shows simulation results for the base case, and cases with 2 kW and 4 kW of battery power used during DR event, respectively. As can be seen from the figures, the BEM can allow more loads to operate during the DR event as the battery power is increased. This results in reduced rebound peak after the DR event ends. The base case shows a demand restrike peak of 14.8 kW, whereas using 2 kW and 4 kW battery power bring it down to 10.5 kW and 7.2 kW, respectively. This benefit is counteracted by more state-of-charge lost as battery power is increased. Figure 6.18 shows the SOC of batteries during and after DR event for different power drawn from the batteries (1, 2, 3, 4 and 5kW).



(a) (b) (c)

Figure 6.17. Demand response simulation results with battery power: (a) 0 kW (base case), (b) 2 kW, and (d) 4 kW.

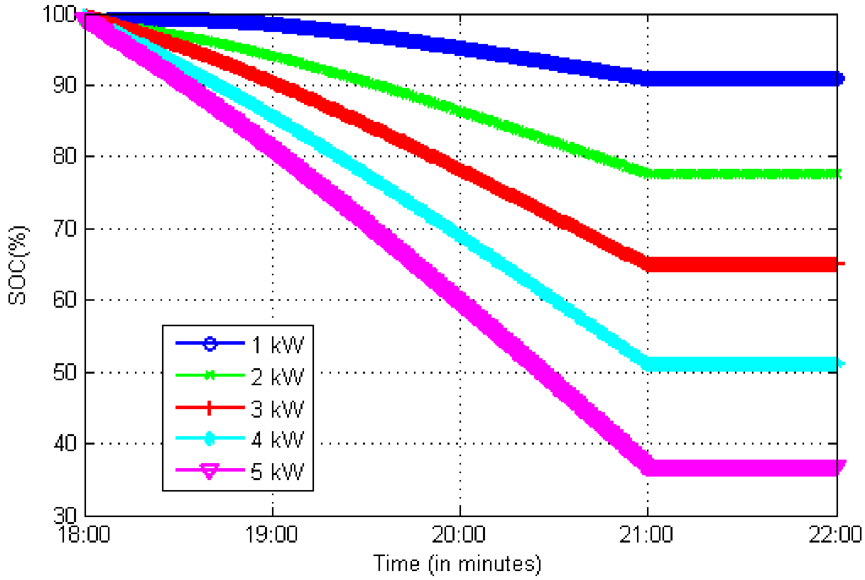


Figure 6.18. Battery state of charge (SOC) during and after DR event for different battery power (shown in legends).

To see further impact on battery SOC, the operation of BEM was simulated from the start of the DR event to the next day for the total of 24 hours. Using the same sunny solar output profile for each case, it was found that for the first two cases (1, and 2 kW of battery power drawn), the batteries could get enough energy to charge back to full SOC during the next day. For 3 kW of battery power drawn, they were marginally able to charge back to full charge. But with 4 kW and 5kW power, the SOC could not be recharged back to its full level.

Hence, in this scenario, up to 3 kW from batteries can be used during DR event. But considering possibility of cloudy days, 2 kW of battery power drawn is selected for the experimental study at the YTU smart house. Note that this threshold will be different for different sizing of DER systems and residential loads, and can be found using similar analysis.

6.2.3.3. Discussion of Experiments at YTU Smart House

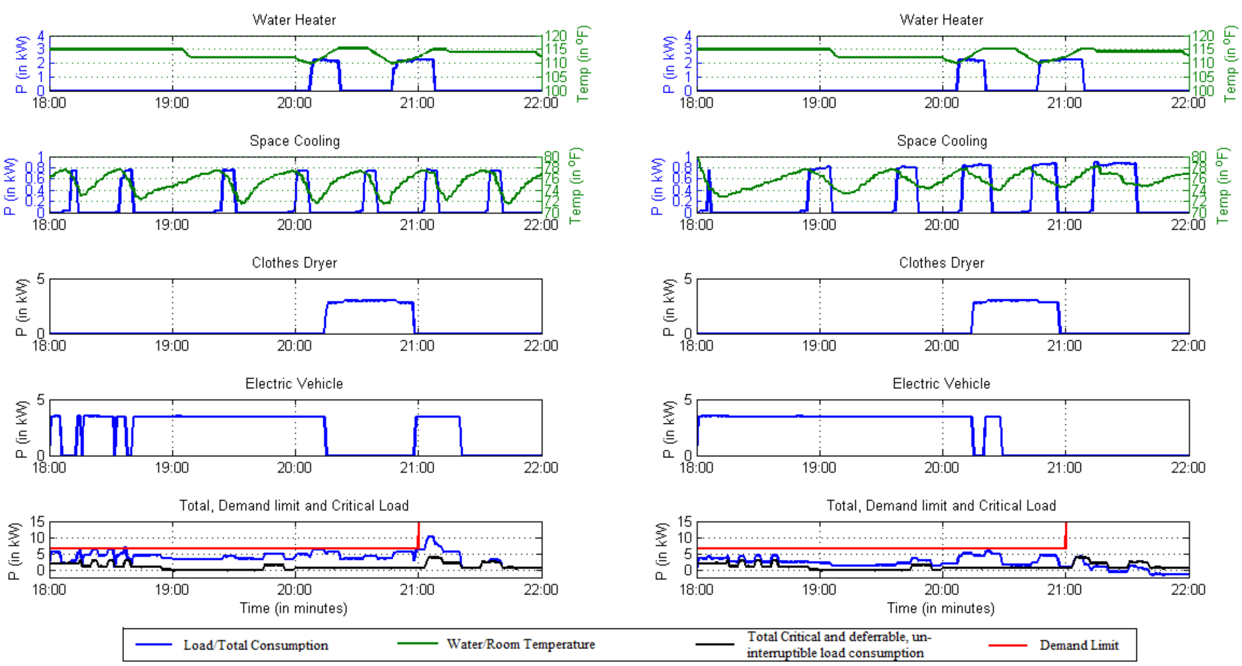
To observe the impact of the proposed BEM algorithm with DER in the real-world smart house environment, two experiments were conducted:

a) Summer case w/o DER;

b) Summer case w/ DER;

For these cases, the BEM program was run for 4 hours to observe the impact of DR on total household loads during the 3-hour DR event (18:00 - 21:00) and the impact of load restrike for one hour (21:00 - 22:00) after the DR event ends.

Case 1: Summer case w/o DER: The demand limit of 6.7 kW was applied for this case for 3 hours from 18:00, as shown in Fig 6.19(a). To keep the total consumption below demand limit, the BEM algorithm turned OFF and ON the EV a number of times from 18:00 to 18:45 to allow critical loads to operate. The EV was also turned OFF during the clothes dryer operation from 20:15 to 21:00. Overall, the demonstration shows that the BEM algorithm keeps the total household consumption below the demand limit of 6.7 kW between 18:00 and 21:00. The washing machine and the dishwasher were deferred until the end of DR event. This resulted in 15-minute load restrike after the DR event ended at 21:00, with a peak consumption of about 10.2 kW.



(a) (b)

Figure 6.19. Demand response demonstration results: (a) summer case w/o DER; and (b) summer case w/ DER.

Case 2: Summer case w/ DER: For this case, the PV and batteries were connected to the system, and inverter was programmed to force-feed 2kW power to the grid/load. Hence, use of DER system in this case resulted in lower amount of load reduction by BEM system during the DR event, as shown in Figure 6.19(b). The EV was only turned OFF for 5 minutes during the peak consumption period of 8:16 pm to 8:20 pm when WH, AC and CD were all ON simultaneously. As expected, the load compensation was also lower (slightly less than 5 kW) due to force-feed power from PV and batteries. Comparing this with Figure 6.13(a), the efficacy of modified algorithm can be easily observed. The period of space cooling operation in this case is longer than the other summer case due to more people moving in and out of the smart house during this case run, which caused the space cooling unit to run for longer duration to maintain the temperature inside the smart house. Note that the difference in rebound peak between the simulation and the experiment (Figure 6.17(b) vs Fig. 6.19(b)) can be attributed to the actual load usage in real case studies (for example, space cooling unit usage), which was different from the simulation load profile.

Results of this case study demonstrate how the BEM algorithm integrated with DER can be useful for residential load management with an incentive-based demand response program.

6.3. Case Study 3: Validation of Learning-Based DR Algorithms

This case study is designed to show the effectiveness of the proposed learning-based DR algorithm on a commercial building.

6.3.1. Case Study Setup

For this case study, a hypothetical commercial building is considered with three zones. Each zone has an HVAC unit, a lighting unit and some plug loads. For sake of simplicity, the plug loads are considered of critical importance during the business hours and therefore not controlled. The lighting loads at each zone has dimming control.

Each of the learning agents for these loads have been individually trained by simulated user feedback for comfort ranges.

The rating of HVAC and lighting loads and their learned comfort preferences are given in Table 6-2.

Table 6-2. Power Rating and Learned Comfort Ranges of the Hypothetical Building Loads

Zone	Load	Power Rating (kW)	Comfort Range
1	HVAC unit	3.3	Preferred: 73°F Allowed: 70-80°F
	Lighting	1.4	Preferred: 73% Allowed: > 50%
2	HVAC unit	4.7	Preferred: 73°F Allowed: 70-80°F
	Lighting	1.8	Preferred: 75% Allowed: > 50%
3	HVAC unit	4.7	Preferred: 75°F Allowed: 70-80°F
	Lighting	1.8	Preferred: 75% Allowed: > 50%

The learning is assumed to have been completed before this case study. Therefore, the learning agents no longer explore, instead only exploit the knowledge of optimal comfort settings for the loads.

6.3.2. Assumption of DR event

A DR event is considered to be scheduled from 3 pm to 5 pm on the day of the case study. A DR signal is received by the BEM one day prior to the day of DR event. A demand limit of 3 kW is imposed during the 2 hour DR event.

Two cases are simulated:

1) Base case: Case without any DR event, when the loads run according to learning agents
2) Case with DR event: A DR event is imposed, and load curtailment is performed.

According to the DR algorithm, the HVAC units are ordered pre-cool before the DR event and the set-point is raised to the highest allowable set-point during DR event. After DR event, the set-points are brought back to original learned values. The pre-cooling start and post-DR return are set at different times for different zones in order to avoid peak load before and after DR event.

For lighting loads, the MILP algorithm finds out the brightness levels with minimum user discomfort, while maintaining total demand within demand limit.

6.3.3. Case Study Results

Zone 1:

Fig 6.20, 6.21, 6.22, and 6.23 show the results for loads in zone 1.

Fig 6.20 is the base case scenario for the HVAC load, which shows the normal operation with the learned set-point of 73^{o}F.

Fig 6.21 shows the AC set-point control and corresponding consumption for the case with DR event. As can be seen from the figure, the pre-cooling algorithm successfully reduces the HVAC load during DR event.

Fig 6.22 shows base case and DR case on the same figure for comparison.

Fig 6.23 shows the results for lighting control in zone 1. It shows the brightness level and power consumption for base case and DR case. As can be seen from the figure, the base case brightness level follows the learned user preference. During the DR event, the brightness is reduced to 52% based on the DR algorithm, which is still within the allowable comfort range.

Similar results are shown for zone 2 and zone 3, which show successful load curtailment for HVAC and lighting loads, while staying within comfort range.

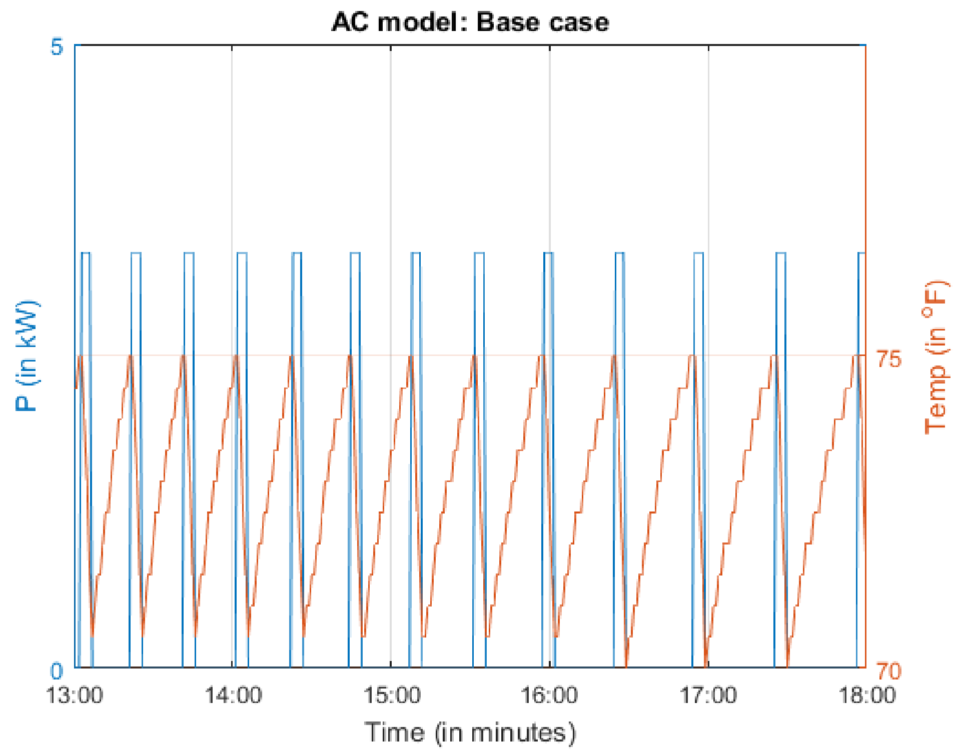


Figure 6.20. Base case HVAC consumption and temperature for Zone 1.

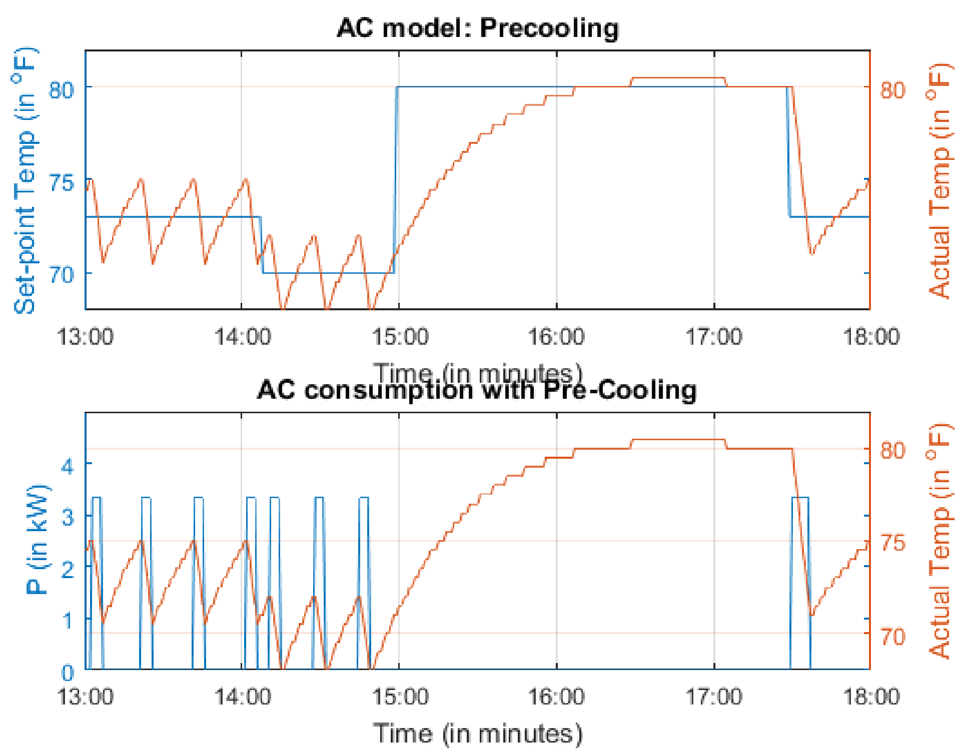


Figure 6.21. Pre-cooling HVAC set-point, consumption, and temperature for DR case for Zone 1.

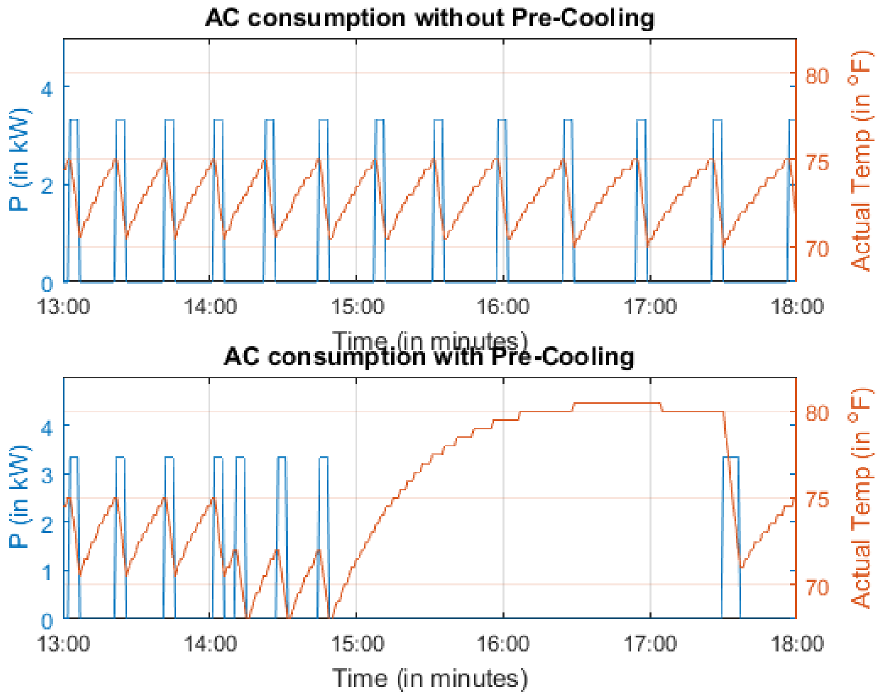


Figure 6.22. Comparison of HVAC consumption between base case and DR case for Zone 1.

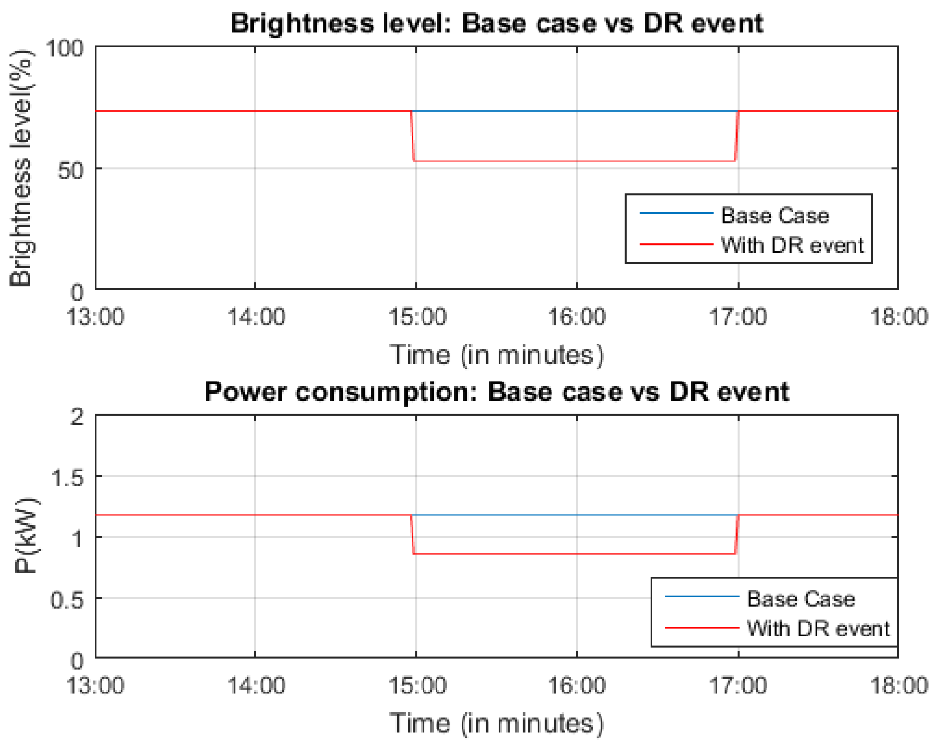


Figure 6.23. Comparison of lighting brightness level and consumption between base case and DR case for Zone 1.

Zone 2:

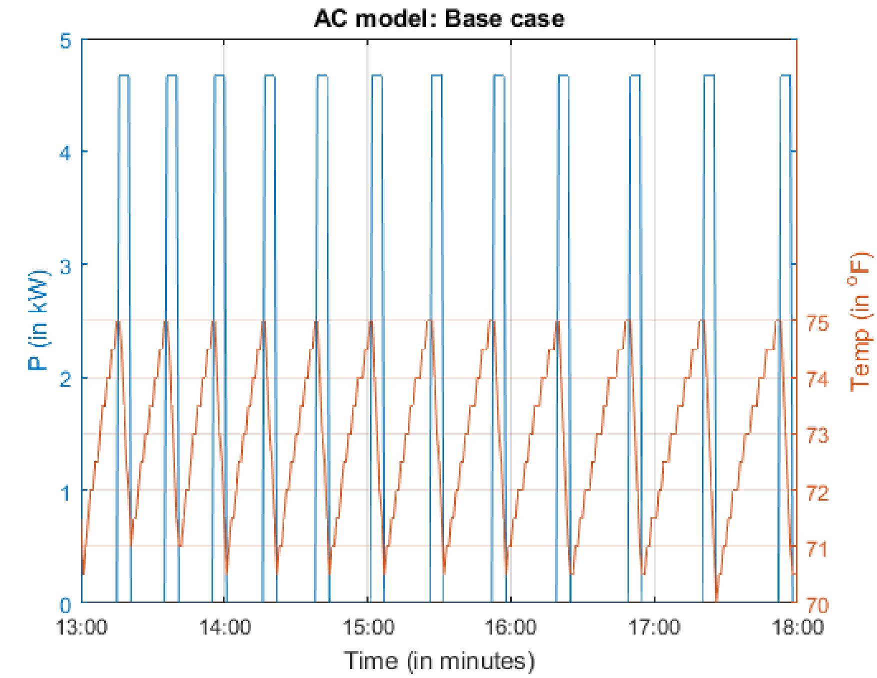


Figure 6.24. Base case HVAC consumption and temperature for Zone 2.

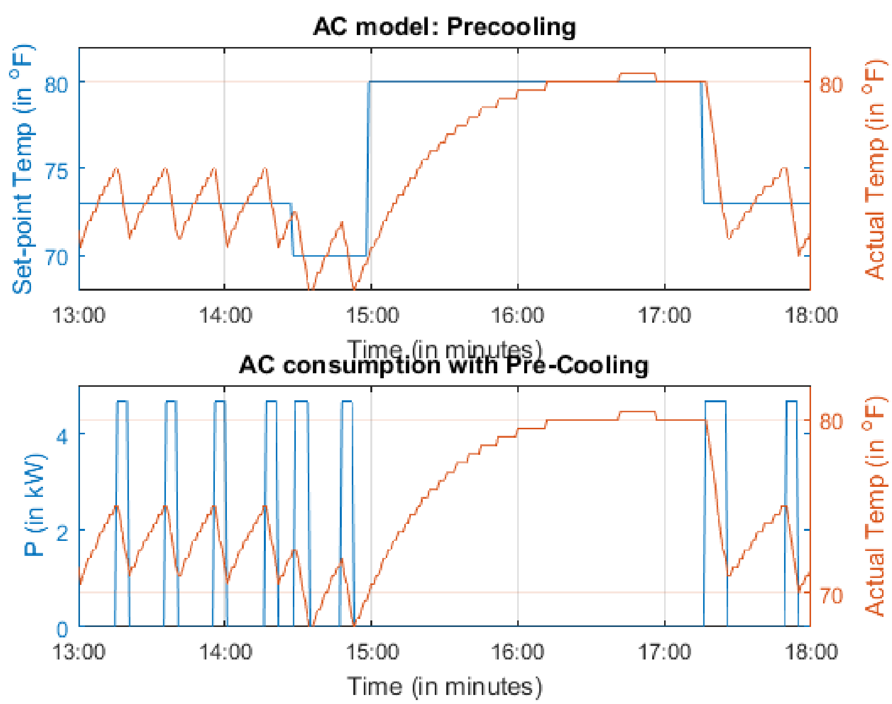


Figure 6.25. Base case HVAC consumption and temperature for Zone 2.

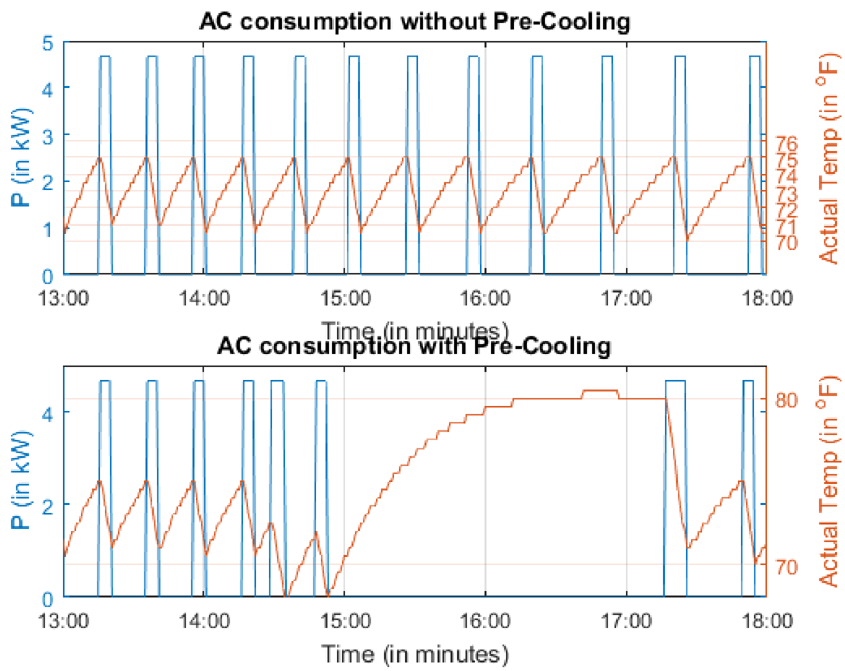


Figure 6.26. Comparison of HVAC consumption between base case and DR case for Zone 2.

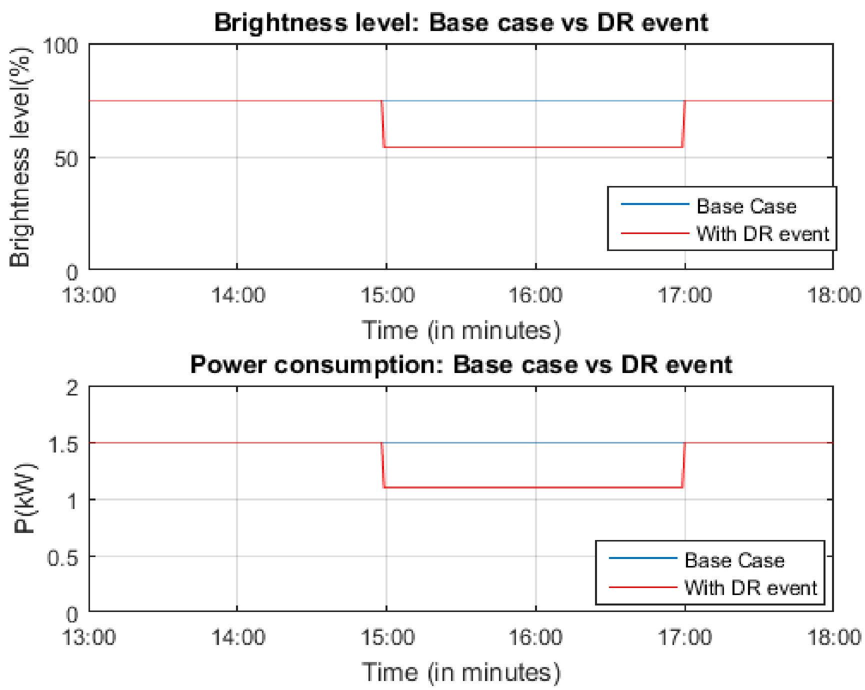


Figure 6.27. Comparison of lighting brightness level and consumption between base case and DR case for Zone 2.

Zone 3:

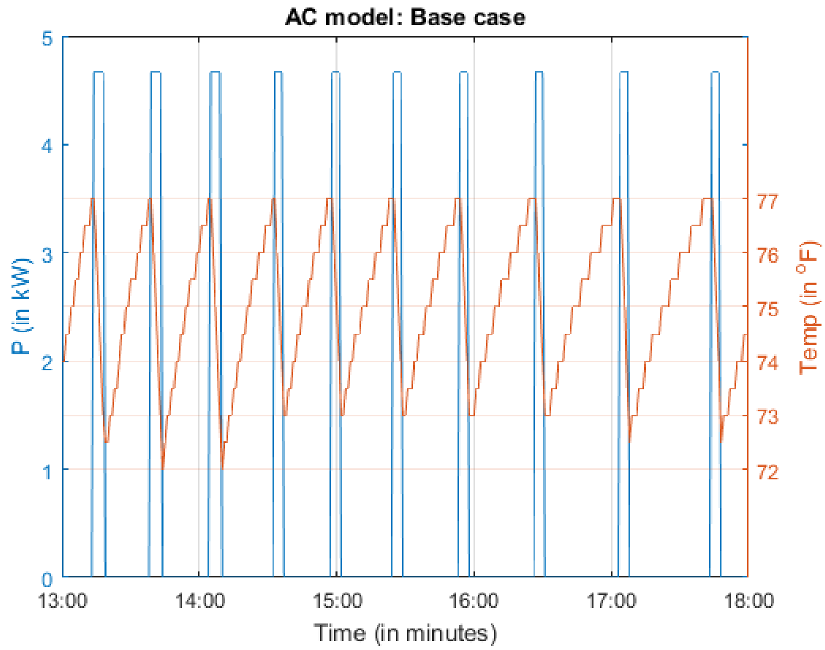


Figure 6.28. Base case HVAC consumption and temperature for Zone 3.

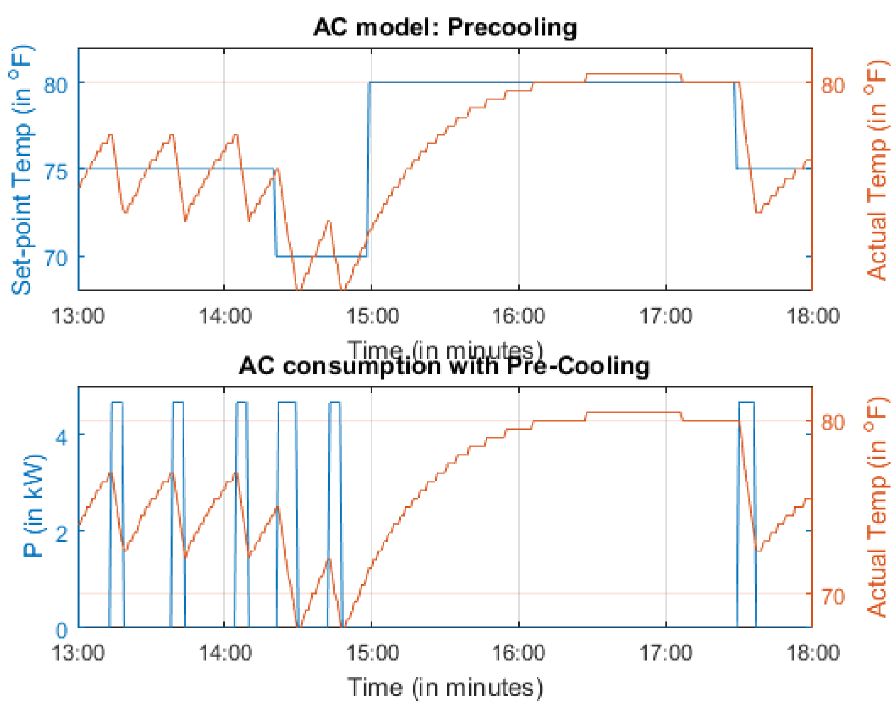


Figure 6.29. Base case HVAC consumption and temperature for Zone 3.

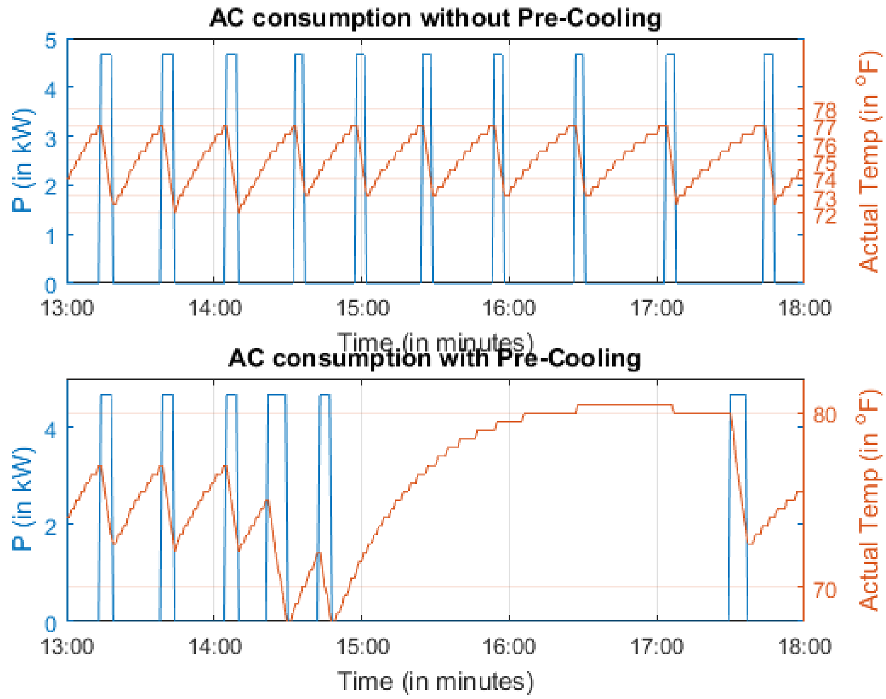


Figure 6.30. Comparison of HVAC consumption between base case and DR case for Zone 3.

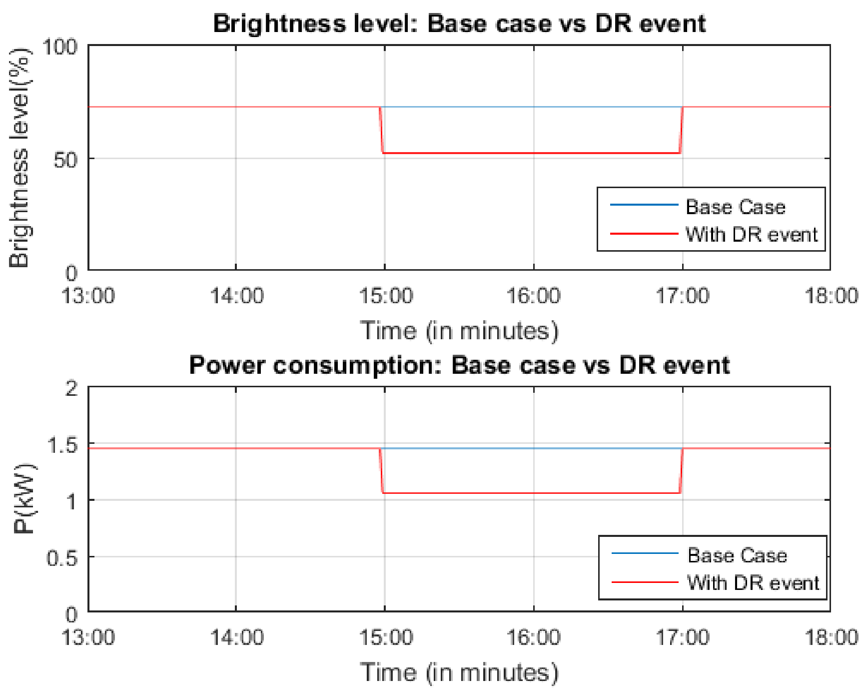


Figure 6.31. Comparison of lighting brightness level and consumption between base case and DR case for Zone 3.

Total demand:

Fig 6.32 shows the total consumption for base case and DR case. As can be seen from the figure, in base case, the peak load from 3-5 pm is almost 14 kW and the base load is always higher than 3kW. During DR case, the proposed algorithm can successfully limit the total demand within the demand limit of 3kW, while keeping the load settings within allowable comfort ranges. This validates the efficacy of the proposed algorithm.

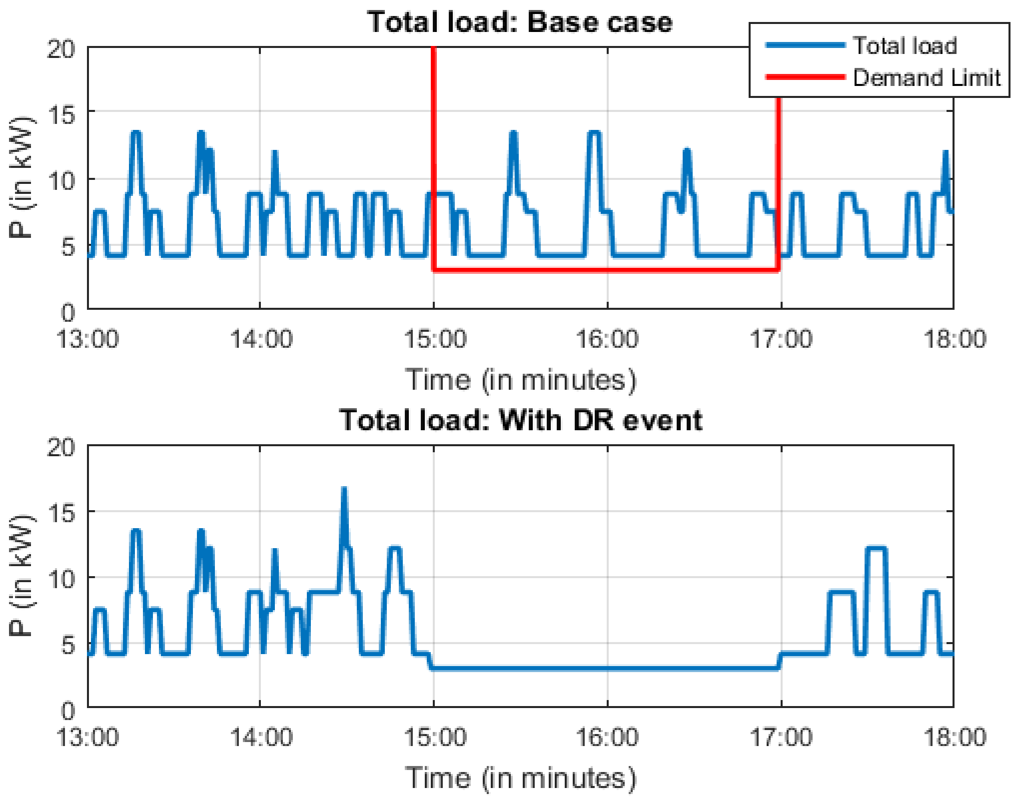


Figure 6.32. Comparison of total consumption between base case and DR case.

7. Summary, Conclusion and Future Work

7.1. Summary

The dissertation reviews existing literature in the field of energy management and demand response in small to medium-sized buildings. The findings from this review indicate a gap in availability of low-cost options for software technology to automatically manage energy in these buildings. The untapped DR potential in this category of buildings can be utilized with an open source software platform with automated energy saving and DR algorithms. The literature review also presents a knowledge gap in availability of self-learning algorithms that can automatically learn user comfort preferences and find opportunities of energy savings while ensuring comfort. The use of reinforcement learning algorithms for building energy management is found to be an idea not fully utilized for energy efficiency and DR. Also, incentive-based DR programs are less studied in literature than their time-based counterparts, and their large peak-load reduction potentials are not fully explored.

The development of software platform within the scope of this dissertation addresses some key issues like: secure plug and play and interoperability of building loads within a software platform, readiness for DR signals through OpenADR, secure communication between software deployments in multiple zones and multiple buildings, and a distributed learning agent architecture that can manage control of loads in a distributed manner for individual energy savings as well as a coordinated manner in case of DR signals. The security strengths and vulnerabilities of the software platform is analyzed for the key components and communications of the system, and some security measures are proposed for identified vulnerabilities.

The second part of the dissertation focuses on development of distributed RL algorithms for automated learning of user comfort preferences and control of loads for efficient energy usage. The building loads are then classified into HVAC, lighting and plug loads; and for each type of load, the standards for comfort are studied and RL algorithms are proposed for automatically learning user comfort preferences. The balance between exploration and exploitation is ensured to find and implement optimal operation of loads that provide the maximum energy savings within user comfort range.

Two sets of DR algorithms are proposed for a building to participate in an incentive-based DR program. First, a priority based algorithm is proposed for residential buildings that uses set-point control on smart thermostats and ON/OFF control for other power intensive loads. The algorithm is made robust against communication issues, and algorithms are proposed to utilize DER (specifically PV and storage) if available. Next, a set of DR algorithms is proposed for commercial buildings that uses the learning from the learning agents and utilizes them for a coordinated operation of loads to reduce peak load in a demand constrained DR scenario. The goal is to find the optimum solution that minimizes user discomfort while maintaining total demand within a specified demand limit. The idea is expanded to a multi-building algorithm which can provide a coordinated peak load reduction for loads in multiple buildings in a campus type facility.

Case studies are presented for the learning and DR algorithms to validate their efficacy in accomplishing the objectives of the dissertation. The effects of user feedback on learning agents are studied for each type of load. Sensitivity of the algorithms on the tuning parameters is also presented. A set of case studies conducted in a smart house are presented to prove the efficacy of the priority based algorithms. Finally, a simulated case study is presented for learning-based DR algorithm in a hypothetical commercial building with loads that are modeled from load profiles from real buildings. The study shows the effectiveness of the proposed DR algorithm for peak load reduction in an incentive-based DR program.

7.2. Conclusion

Contributions to the development of the software platform ensure that the building load controlling devices can be seamlessly integrated into the system. For device integration - discovery APIs, API translators and corresponding agents have to be developed within the BEMOSS™ platform. Laboratory experiments with off-the-shelf commercial devices integrated to BEMOSS™ running on a low-cost single board computer show that, the devices can be easily discovered, monitored and controlled. Both manual user control and automated learning-based control have been demonstrated. Although the discovery process may take longer for some protocols like Modbus, it doesn’t affect in the long run, as discovery is usually performed only once at the time of inclusion of the device into the system. The controls during subsequent operation are prompt, and only marginally affected by communication issues, especially if the device is a cloud-based device. Analysis of the security of the software shows that vulnerabilities have been properly taken care of through counter-measures built within the system. The approval process for devices and nodes helps prevent unauthorized actors within the system. The sniffing-based detection further strengthens the security by detecting unauthorized devices in the network. A round-trip OpenADR control experiment which involves getting signal through OpenADR, reacting to the signal during the DR period, restoring base case settings after the end of the DR period, and sending feedback through OpenADR, shows that it is feasible to use BEMOSS™ for OpenADR implementation.

Results from the case studies with learning agents show that the agents always look for energy saving opportunities within the customer comfort range. The agents explore the states of more energy saving until customer provides negative feedback and so can find the optimal energy saving states before customer discomfort starts. Results also show that individual energy savings during energy efficient operation of the learning agents can be highly dependent on an individual user and their preferences for comfort. The more flexible the user's comfort preferences are; the more energy savings are possible. Therefore, an estimate of energy savings cannot be made without interacting with the user environment first. In any case, the learning agents are shown to be providing maximum energy efficiency by reducing energy wastage that is not essential for user comfort. Also, the learning agents are shown to be effective in finding optimal comfort settings for different occupants in different zones. Results from the DR case study show that the proposed DR algorithm is effective in maintaining demand constraints during a DR event, while minimizing user discomfort. For the learning-based algorithm, the discomfort caused is only based on individual user's comfort preferences, and even in the worst case scenario, the algorithm does not violate minimum comfort standards.

Overall, the work presented in this dissertation is expected to encourage participation of small to medium-sized buildings in energy efficiency and DR activities. The ease-of-use of the software platform and the automated operation of learning agents can provide energy saving benefits for the user without much effort required from the user. The incentive-based DR algorithm can assist the utilities to alleviate power system stress conditions through peak load reduction.

7.3. Recommendations for Future Work

This work can be expanded to build a deployment-ready software technology for energy management in commercial buildings. The following are some recommendations for future research directions:

1. Interoperability of building loads: This work proposes a framework of API translators and automated API translator generation for standard APIs to allow interoperability of devices within the software platform. Following the current research trends for Internet of Things (IoT), this can be expanded to include most building loads in the IoT framework. The open source nature of the software platform encourages contributions from developers to include API translators for devices. The more devices are brought within the software platform, the more extensive the interoperability standard of the platform will become. The interoperability standards are still being developed, and this work can contribute to that development.

2. Learning algorithms for residential loads: The RL algorithms developed in this work focus on three types of loads that are most common in commercial buildings: HVAC, lighting and plug loads. As residential loads are more varied in nature in terms of power consumption profiles and user comfort preferences, concepts in this work can be extended to develop RL algorithms for various residential loads as well. This can make the DR algorithm proposed in this work applicable for residential buildings.

3. Occupancy detection: The RL algorithms developed in this work assume that the occupancy in a given building zone is known to the agent through use of occupancy sensors. In the section for RL algorithms for lighting loads, a methodology is proposed to detect occupancy in the absence of occupancy sensors through the exploration of the lighting agents. This concept can be extended to develop a comprehensive learning algorithm for detection of occupancy. Also, occupancy is only considered a binary parameter in this work. The ability to detect the number of occupants in a room or zone can be very helpful for learning algorithms to predict optimal comfort conditions. This can be developed through the use of image deconstruction and pattern recognition algorithms, similar to ones used in Microsoft Kinect. Then this detection can be incorporated in the learning algorithms for better comfort control.

4. Performance analysis: The performance of the RL algorithms developed in this work can be analyzed for hardware with different computational resources. The usage of programming languages (like C or FORTRAN) closer to machine language to code computation-intensive algorithms and then linking the results to the python-based software platform is an option that can be explored to find opportunities of better performance.

References

[1] “Data from PJM’s hourly load data” [Online]. Available: http://pjm.com/markets-and-operations/energy/real-time/loadhryr.aspx Feb. 2012

[2] DOE-EIA, “Energy efficiency and renewable energy. International energy outlook,” 2011 [Online]. Available: http://205.254.135.24/forecasts/ieo/pdf/0484(2011).pdf

[3] Albadi, M.; and El-Saadany, E.; "Demand response in electricity markets: An overview," in Proc. Power Eng. Soc. Gen. Meet., Jun. 2007, pp. 1-5.

[4] C. W. Gellings and J. H. Chamberlin. *Demand Side Management: Concepts and Methods*. (2nd edition).Tulsa, OK: PennWell Books. 1993.

[5] C.W. Gellings. “The concept of demand-side management for electric utilities” in *Proc. of the IEEE*, vol.73, no.10, pp. 1468- 1470, October, 1985

[6] D.R. Limaye. “Implementation of demand-side management programs”. in *Proc. of the IEEE.* vol.73, no.10, pp. 1503- 1512, October, 1985

[7] C.W. Gellings. and W.M. Smith. “Integrating demand-side management into utility planning”. in *Proc. of the IEEE.*, vol.77, no.6, pp.908-918, June, 1989

[8] R. Bhatnagar and S. Rahman. “Direct Load Control: Relationships between Electric Utility Experiences/Assessments and System Characteristics”. *IEEE Transactions on Power Apparatus and Systems*. vol. PAS-104, no.8, pp.2168-2175, August, 1985

[9] U.S. Department of Energy. (2006, February). *Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them: A Report to the United States Congress Pursuant to Section 1252 of the Energy Policy Act of 2005*. [On-line] Available: http://eetd.lbl.gov/ea/EMP/reports/congress-1252d.pdf.

[10] The Federal Energy Regulatory Commission Staff. (2010, June 17th). *National Action Plan on Demand Response*. [On-line]. Available: www.ferc.gov/legal/staff-reports/06-17-10-demand-response.pdf.

[11] Federal Energy Regulatory Commission, Assessment of Demand Response and Advanced Metering, February 2011 [Online]. Available: http://www.ferc.gov/legal/staff-reports/2010-dr-report.pdf.

[12] M. A. Piette, D.Watson, N.Motegi, S. Kiliccote, and E. Linkugel, “Automated demand response strategies and commissioning commercial building controls,” in *Proc. 14th Natl. Conf. Building Commissioning*, San Francisco, CA, Apr. 2006.

[13] J. F. Kelly, M. Meiners and G. Rouse. (2007, October 15th). *IIT Perfect Power Prototype Final Report*.
[On-line]. Available: www.galvinpower.org/files/IIT_Perfect_Power_Prototype.pdf.

[14] Xcel Energy. (2008). *SmartGridCityTM: Design Plan for Boulder, Colo*. [On-line]. Available: http://smartgridcity.xcelenergy.com/media/pdf/SmartGridCityDesignPlan.pdf.

[15] PG&E, Final Report: 2010 Load Impact Evaluation for PG&E Company's SmartAC Program, April 2011 [Online]. Available: http://www.sgiclearinghouse.org/node/4499.

[16] Southern California Edison, Program Impact Evaluation of 2004 SCE Energy Smart Thermostat Program Final Report, 2005 [Online]. Available: http://www.sgiclearinghouse.org/node/4458.

[17] KEMA, 2005 Smart Thermostat Program Impact Evaluation, 2006 [Online]. Available: http://www.sgiclearinghouse.org/node/2406.

[18] PowerCentsDC Program Final Report, 2010 [Online]. Available: http://www.sgiclearinghouse.org/node/4457.

[19] S. Lutzenhiser, J. Peters, M. Moezzi and J. Woods, Beyond the price effect in time-of-use programs: results from a municipal utility pilot, 2007-2008, 2009 [Online]. Available: http://www.sgiclearinghouse.org/node/1759.

[20] Smarthouse-smartgrid, Online: http://www.smarthouse-smartgrid.eu/

[21] BeyWatch, Online: http://www.beywatch.eu/

[22] BeAwarw, Online: http://www.energyawareness.eu/beaware/

[23] SmartWatts, Online: http://www.smartwatts.de

[24] Sesame, Online: http://sesame.ftw.at/

[25] Sesame-S, Online: http://sesame-s.ftw.at

[26] Demand Response Initiative, Online: http://www.ieso.ca/imoweb/consult/demandresponse.asp

[27] SmartCamp, Online: http://arc.gov.au/pdf/LP10_R2/Curtin_University_of_Technology.pdf

[28] EcoGrid, Online: http://energinet.dk/EN/FORSKNING/Energinetdk-research-anddevelopment/EcoGrid/Sider/EU-EcoGrid-net.aspx

[29] DiYSE, Online: http://dyse.org:8080/pages/viewrecentblogposts.action?key=hometest

[30] Virginia Tech – Advanced Research Institute. "Smart Grid Information Clearing House". Online: http://www.sgiclearinghouse.org/demo_projects.

[31] PJM, https://www.pjm.com

[32] PJM, Demand Response End Use Customer Fact Sheet, online: https://www.pjm.com/~/media/markets-ops/dsr/end-use-customer-fact-sheet.ashx

[33] PJM, List of Curtailment Service Providers, online: http://www.pjm.com/markets-and-operations/demand-response/csps.aspx

[34] PJM, Reliability Pricing Model Fact Sheet, online: http://www.pjm.com/~/media/about-pjm/newsroom/fact-sheets/rpm-fact-sheet.ashx

[35] PJM Stuff Whitepaper, Price Responsive Demand, online: http://www.pjm.com/~/media/documents/reports/pjm-whitepaper-on-price-responsive-demand.ashx

[36] OpenADR, http://www.openadr.org

[37] David G Holmberg, “OpenADR Advances”, LBNL peer-reviewed paper, online: http://escholarship.org/uc/item/3dn3426x

[38] OASIS Energy Interoperation Technical Committee (EI---TC), information available at: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=energy-interop

[39] Smart Grid Interoperability Panel, SGIP: http://collaborate.nist.gov/twiki-sggrid/bin/view/SmartGrid/WebHome

[40] PJM, Advanced Technology Pilot Program – Project Guidelines, online: www.pjm.com/~/media/markets-ops/advanced-tech-pilots/advanced-technology-pilot-program-project-guidelines.ashx

[41] PJM, OpenADR 2.0 ATR Pilot, online: http://www.pjm.com/~/media/Images/ctc-display/modules/pilots/ipkeys/pjm-pilot-materals.ashx

[42] IPKeys press release, online: http://move.ipkeys.net/06-05-2012/

[43] James S. Boch, “PJM Interconnection OpenADR 2.0 Advanced Technology Resource Pilot,” online: http://energy.ipkeys.com/wp-content/uploads/2013/11/AEEP_PJM_OpenADR_ATR_Pilot.pdf

[44] LBL, News on OpenADR 2.0b market pilot, online: http://drrc.lbl.gov/news/ipkeys-pjm-walmart-berkeleylab-and-schneider

[45] OpenADR, News on OpenADR 2.0b pilot, online: http://www.openadr.org/index.php?option=com_content&view=article&id=96:ipkeys-at-pjm-interconnection&catid=29:news&Itemid=129

[46] Buildings Energy Data Book, Buildings Sector, [Online]. Available: http://buildingsdatabook.eren.doe.gov/ChapterIntro1.aspx?1#4– November 1, 2011.

[47] EIA - Commercial Building Energy Consumption Survey (CBECS), Available: http://www.eia.gov/consumption/commercial/data/2003/index.cfm?view=consumption#e1a

[48] Building Technologies Office, Office of Energy Efficiency & Renewable Technology, Department of Energy [Online]. Available: http://energy.gov/eere/efficiency/buildings.

[49] PNNL, “Small- and Medium-Sized Commercial Building Monitoring and Controls Needs: A Scoping Study,” October 2012, Available: http://www.pnnl.gov/main/publications/external/technical_reports/PNNL-22169.pdf.

[50] SmartThings website, Available: http://www.smartthings.com/ [Online]

[51] Staples Connect powered by Linksys [Online]. Available: www. staples.com/sbd/cre/marketing/staples-connect/.

[52] GE Brillion Connected Appliances [Online]. Available: www.geapp liances.com/connected-home-smart-appliances/.

[53] Iris Smart Home Management System [Online]. Available: www.lowes.com/cd_Iris_239939199.

[54] Revolv: Life & Smart Home Awesomation [Online]. Available: http://revolv.com/.

[55] Freedomotic Open Source Building Automation [Online]. Available: http://freedomotic.com/.

[56] OpenRemote Open Source Automation Platform [Online]. Available: http://www.openremote.org/.

[57] openHAB empowering the smart home [Online]. Available: http://www.openhab.org/.

[58] Buildings Energy Data Book, Commercial Sector, [Online]. Available: http://buildingsdatabook.eren.doe.gov/ChapterIntro3.aspx.

[59] Biao Sun; Luh, P.B.; Qing-Shan Jia; Ziyan Jiang; Fulin Wang; Chen Song, "Building Energy Management: Integrated Control of Active and Passive Heating, Cooling, Lighting, Shading, and Ventilation Systems," IEEE Trans on Automation Science and Engineering, vol. 10, no. 3, pp. 588-602, July 2013.

[60] Kuzlu, M.; Pipattanasomporn, M., "Assessment of communication technologies and network requirements for different smart grid applications," *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES* , vol., no., pp.1,6, 24-27 Feb. 2013.

[61] S. M. Amin, "Smart grid security, privacy, and resilient architectures: Opportunities and challenges," *IEEE PES General Meeting, 2012*, pp. 1-2, San Diego, CA, July 2012.

[62] C. Karlof, and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Proc. Of 2003 IEEE Intl. Workshop on Sensor Network Protocols and Applications,* pp. 113-127, May 2003.

[63] P. A. Subrahmanyam, D. Wagner, D. Mulligan, E. Jones, U. Shankar, and J. Lerner, "Network Security Architecture for Demand Response/Sensor Networks," Consultant Report to California Energy Commission, October 2005.

[64] M. Paranjpe, "Security and Privacy in Demand Response Systems in Smart Grid," M.S. Report to California State University, Sacramento, 2010.

[65] M. Zillgith, D. Nestle, and M. Wagner, "Security Architecture of the OGEMA 2.0 Home Energy Management System," *Proc. of 2013 Intl. ETG-Congress: Symposium 1: Security in critical infrastructures today*, pp. 1-6, Nov 2013.

[66] Longhua Guo; Jianhua Li; Jun Wu; Zhengmin Xia; Shengjun Zheng, "A Security Mechanism for Demand Response Using RBAC and Pub/sub," 2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems (ISADS), pp.260-265, 25-27 March 2015

[67] Hongwei Li; Xiaodong Lin; Haomiao Yang; Xiaohui Liang; Rongxing Lu; Xuemin Shen, "EPPDR: An Efficient Privacy-Preserving Demand Response Scheme with Adaptive Key

Evolution in Smart Grid," IEEE Transactions on Parallel and Distributed Systems, vol.25, no.8, pp.2053-2064, Aug. 2014

[68] Mohan, A.; Mashima, D., "Towards Secure Demand-Response Systems on the Cloud," 2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), pp.361-366, 26-28 May 2014

[69] Cleveland, F.M., "Cyber security issues for Advanced Metering Infrasttructure (AMI)," 2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, pp.1-5, 20-24 July 2008

[70] Kamto, J.; Lijun Qian; Fuller, J.; Attia, J., "Light-weight key distribution and management for Advanced Metering Infrastructure," 2011 IEEE GLOBECOM Workshops (GC Wkshps), pp.1216-1220, 5-9 Dec. 2011

[71] Seung-Hyun Seo; Xiaoyu Ding; Bertino, E., "Encryption key management for secure communication in smart advanced metering infrastructures," IEEE International Conference on Smart Grid Communications (SmartGridComm) 2013, pp.498-503, 21-24 Oct. 2013

[72] Nian Liu; Jinshan Chen; Lin Zhu; Jianhua Zhang; Yanling He, "A Key Management Scheme for Secure Communications of Advanced Metering Infrastructure in Smart Grid," IEEE Transactions on Industrial Electronics, vol.60, no.10, pp.4746-4756, Oct. 2013

[73] Yang Liu; Shiyan Hu; Tsung-Yi Ho, "Vulnerability assessment and defense technology for smart home cybersecurity considering pricing cyberattacks," 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp.183-190, 2-6 Nov. 2014

[74] Komninos, N.; Philippou, E.; Pitsillides, A., "Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures," IEEE Communications Surveys & Tutorials, vol.16, no.4, pp.1933-1954, Fourthquarter 2014

[75] Yue Li, "Design of a Key Establishment Protocol for Smart Home Energy Management System," 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), pp.88-93, 5-7 June 2013

[76] Antonini, A.; Barenghi, A.; Pelosi, G.; Zonouz, S., "Security challenges in building automation and SCADA," 2014 International Carnahan Conference on Security Technology (ICCST), pp.1-6, 13-16 Oct. 2014

[77] Boyes, H.A., "Cyber security of intelligent buildings: A review," 8th IET International System Safety Conference incorporating the Cyber Security Conference 2013, pp.1-7, 16-17 Oct. 2013

[78] Boyes, H.A.; Isbell, R.; Norris, P.; Watson, T., "Enabling intelligent cities through cyber security of building information and building systems," IET Conference on Future Intelligent Cities, pp.1-6, 4-5 Dec. 2014.

[79] A.Brooks, E.Lu, D.Reicher, C.Spirakis, B.Weihl, "Demand dispatch," IEEE Power Energy Mag., vol.8, no.3, pp.20-29, May/June 2010.

[80] P. Zhang, F. Li, and N. Bhatt, "Next-generation monitoring, analysis, and control for the future smart control center," IEEE Trans. Smart Grid, vol. 1, no. 2, pp. 186-192, Sep. 2010.

[81] M. Erol-Kantarci and H. T.Mouftah, "Wireless multimedia sensor and actor networks for the next generation power grid," Ad Hoc Networks, vol. 9, no. 4, pp. 542-551, Jun. 2011.

[82] A.R.Al-Ali, A.El-Hag, M.Bahadiri, M.Harbaji, A.E.H.Yousef, "Smart home renewable energy management system," Energy Procedia, vol.12, pp.120-126, Sep. 2011.

[83] O.Ma, N.Alkadi, P.Cappers, and P.Denholm, "Demand response for ancillary services," IEEE Trans. Smart Grid, vol.4, no.4, pp.1988-1995, Dec. 2013.

[84] S.H.Li, D.Zhang, A.B.Roget, Z.O'Neill, "Integrating home energy simulation and dynamic electricity price for demand response study," IEEE Trans. Smart Grid, vol.5, no.2, pp.779-788, Mar. 2014.

[85] A. H. Mohsenian-Rad, and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," IEEE Trans. On Smart Grid, vol. 1, no. 2, 2010.

[86] Tsui, K.M.; Chan, S.C., "Demand Response Optimization for Smart Home Scheduling Under Real-Time Pricing," IEEE Transactions on Smart Grid, vol. 3, no. 4, pp. 1812-1821, Dec. 2012.

[87] T. Logenthiran, D. Srinivasan, and T. Z. Shun, "Demand side management in smart grid using heuristic optimization," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1244–1252, 2012.

[88] Mohsenian-Rad, A. H.; Wong, V. W. S.; Jatskevich, J.; and Schober, R.; "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid," in Proc. The Innovative Smart Grid Technologies (ISGT) Conference, Jan 2010, Gaithersburg, MD, pp. 1-6.

[89] M. Pipattanasomporn, M. Kuzlu, and S. Rahman, "An algorithm for intelligent home energy management and demand response analysis," in IEEE Trans. On Smart Grid, vol. 3, no. 4, pp.2166-2173, 2012.

[90] Li, J.; Chuang, J. Y.; Xiao, J.; Hong, J. W.; and Boutaba, R.; "On the design and implementation of a home energy management system," in Proc. The 6th International Symposium on Wireless and Pervasive Computing (ISWPC), Feb 2011, Hong Kong, pp. 1-6.

[91] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Hardware demonstration of a home energy management system for demand response applications," in IEEE Trans. On Smart Grid, vol. 3, no. 4, pp. 1704-1711, 2012.

[92] Han, D.; and Lim, J.; "Design and implementation of smart house energy management systems based on ZigBee," IEEE Trans. On Consumer Electronics, 2010, vol. 56, no. 3, pp. 1417-1425.

[93] Son, Y. S.; Pulkkinen, T.; Moon, K. Y.; and Kim, C.; "Home energy management system based on power line communication," IEEE Trans. On Consumer Electronics, 2010, vol. 56, no. 3, pp. 1380-1386.

[94] Han, J.; Choi, C. –S.; Park, W. –K.; Lee, I; Kim, S. –H., "Smart home energy management system including renewable energy based on ZigBee and PLC," IEEE Trans. On Consumer Electronics, vol. 60, no. 2, pp. 198-202, May 2014.

[95] Han, J. –S.; Choi, C. –S.; and Lee, I. –W.; "More efficient home energy management system based on ZigBee communication and infrared remote controls," IEEE Trans. On Consumer Electronics, 2011, vol. 57, no. 1, pp. 85-89.

[96] Huang, H. –Y.; Yen, J. –Y.; Chen, S. –L.; Ou, F. –C., "Development of an intelligent energy management network for building automation," IEEE Trans on Automation Science and Engineering, vol. 1, no. 1, pp. 14-25, Jul 2004.

[97] Byun, J.; Hong, I.; and Park, S.; "Intelligent cloud home energy management system using household appliance priority based scheduling based on prediction of renewable energy capability," IEEE Trans. on Consumer Electronics, 2012, vol. 58, no. 4, pp. 1194-1201.

[98] Chang, T. -H.; Alizadeh, M.; and Scaglione, A.; "Coordinated home energy management for real-time power balancing," 2012 IEEE PES General Meeting, Jul 2012, San Diego, CA, pp. 1-8.

[99] Chen Chen; Kishore, S.; Snyder, L.V., "An innovative RTP-based residential power scheduling scheme for smart grids," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.5956-5959, 22-27 May 2011.

[100] Na Li; Lijun Chen; Low, S.H., "Optimal demand response based on utility maximization in power networks," 2011 IEEE PES General Meeting, pp.1-8, 24-29 July 2011.

[101] Costanzo, G.T.; Guchuan Zhu; Anjos, M.F.; Savard, G., "A System Architecture for Autonomous Demand Side Load Management in Smart Buildings," IEEE Transactions on Smart Grid, vol. 3, no. 4, pp.2157-2165, Dec. 2012.

[102] C. Roe, S. Meliopoulos, R. Entriken, and S. Chhaya, "Simulated demand response of a residential energy management system," *Proc. IEEE Energytech*, pp. 1–6, 2011.

[103] Kato, T.; Yuasa, K.; Matsuyama, T., "Energy on demand: Efficient and versatile energy control system for home energy management," 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp.392-397, 17-20 Oct. 2011.

[104] M. Lissere, T. Sauter, and J. Y. Hung, "Future Energy Systems: Integrating Renewable Energy Sources into the Smart Power Grid Through Industrial Electronics," *IEEE Industrial Electronics Magazine*, vol. 4, no. 1, pp. 18-37, Mar 2010.

[105] C. Cecati, C. Citro, and P. Siano, "Combined Operations of Renewable Energy Systems and Responsive Demand in a Smart Grid," *IEEE Trans on Sustainable Energy*, vol. 2, no.4, pp. 468-476, Oct 2011.

[106] Y. Guo, M. Pan, and Y. Fang, "Optimal Power Management of Residential Customers in the Smart Grid," *IEEE Trans on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1593-1606, Sept 2012.

[107] H. Wu, M. Shahidehpour, and A. Al-Abdulwahab, "Hourly demand response in day-ahead scheduling for managing the variability of renewable energy," *IET Generation, Transmission & Distribution*, vol. 7, no. 3, pp. 226-234, March 2013.

[108] Y. Guo, M. Pan, Y. Fang, and P. P. Khargonekar, "Decentralized Coordination of Energy Utilization for Residential Households in the Smart Grid," *IEEE Trans on Smart Grid*, vol. 4, no. 3, pp. 1341-1350, Sept 2013.

[109] A. Papavasiliou, and S. S. Oren, "Large-Scale Integration of Deferrable Demand and Renewable Energy Sources," *IEEE Trans on Power Systems*, vol. 29, no. 1, pp. 489-499, Jan 2014.

[110] X. Liu; L. Ivanescu, R. Kang, and M. Maier, "Real-time household load priority scheduling algorithm based on prediction of renewable source availability," *IEEE Trans on Consumer Electronics*, vol. 58, no. 2, pp. 318-326, May 2012.

[111] S. Chen, N. B. Shroff, and P. Sinha, "Heterogeneous Delay Tolerant Task Scheduling and Energy Management in the Smart Grid with Renewable Energy," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1258-1267, July 2013.

[112] S. Squartini, D. Fuselli, M. Boaro, F. De Angelis, and F. Piazza, "Home energy resource scheduling algorithms and their dependency on the battery model," *2013 IEEE Symp on Computational Intelligence Applications in Smart Grid*, pp. 122-129, Apr 2013.

[113] A. J. Roscoe, and G. Ault, "Supporting high penetrations of renewable generation via implementation of real-time electricity pricing and demand response," *IET Renewable Power Generation*, vol. 4, no. 4, pp. 369-382, July 2010.

[114] A. R. Boynuegri, B. Yagcitekin, M. Baysal, A. Karakas, and M. Uzunoglu, “Energy management algorithm for smart home with renewable energy sources,” *2013 POWERENG*, pp. 1753-1758, May 2013.

[115] P. T. V. B. N. Kumar, S. Suryateja, G. Naveen, M. Singh, and P. Kumar, "Smart home energy management with integration of PV and storage facilities providing grid support," *2013 IEEE Power and Energy Society General Meeting (PES GM)*, pp. 1-5, 21-25 July 2013.

[116] C. –S. Choi, J. I. Lee, and I. –W. Lee, “Complex home energy management system architecture and implementation for green home with Built-in Photovoltaic and motorized blinders,” *2012 Intl. Conf. on ICT Convergence*, pp. 295-296, Oct 2012.

[117] Y. –M. Wi, J. –U. Lee, and S. –K. Joo, "Electric vehicle charging method for smart homes/buildings with a photovoltaic system," *IEEE Trans on Consumer Electronics*, vol. 59, no. 2, pp. 323-328, May 2013.

[118] D. Angeli and P. A. Kountouriotis, “A stochastic approach to dynamic demand refrigerator control,” *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 3, pp. 581–592, 2012.

[119] PJM’s Electricity Storage [Online]. Available: http://www.pjm.com/about-pjm/exploring-tomorrows-grid/electricity-storage.aspx

[120] S. Kishore and L. V. Snyder, “Control mechanisms for residential electricity demand in smart grids,” in *Proc. IEEE Smart Grid Comm*, 2010, pp. 443–448.

[121] F.Mangiatordi, E. Pallotti, P. D. Vecchio, and F. Leccese, “Power consumption scheduling for residential buildings,” in *Proc. IEEE Environ. Elect. Eng. (EEEIC)*, 2012, pp. 926–930.

[122] P. Du, and N. Lu, “Appliance commitment for household load scheduling,” *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 411–419, 2011.

[123] G. Koutitas, “Control of flexible smart devices in the smart grid,” *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1333–1343, 2012.

[124] J. Han, C. S. Choi, W. K. Park, and I. Lee, “Green home energy management system through comparison of energy usage between the same kinds of home appliances,” in Proc. the 15h IEEE Int. Symp. Consumer Electronics (ISCE), pp. 1-4, 2011.

[125] Pipattanasomporn, M.; Kuzlu, M.; Rahman, S.; Teklu, Y., "Load Profiles of Selected Major Household Appliances and Their Demand Response Opportunities," *IEEE Transactions on Smart Grid*, vol.5, no.2, pp.742-750, March 2014.

[126] Erol-Kantarci, M.; and Mouftah, H. T.; “Wireless sensor networks for cost efficient residential energy management in the smart grid,” IEEE Trans. On Smart Grid, 2011, vol. 2, no. 2, pp. 314-325.

[127] Schneider Electric, Wiser Home Management, Available: https://www.wiserenergy.com/ [Online]

[128] General Electric, GE Energy Management, Available: http://www.geenergymanagement.com/ [Online]

[129] Honeywell, Energy Management products, Available: http://honeywell.com/Solutions-Technologies/Pages/energy.aspx#Management [Online]

[130] Digi International, Digi ZBee Smart Plug, Available: http://www.digi.com/products/wireless-modems-peripherals/wireless-range-extenders-peripherals/xbee-smart-plug-zb [Online]

[131] EnTek International LLC, Remote Appliance Controller, Available: http://enteklc.com/wp-content/uploads/2011/02/mc140-datasheet-1-26-11.pdf [Online]

[132] Ecobee, Smart Plug, Available: http://www.ecobee.com/wp-content/themes/ecobee/etc/smartPlug_a4_april9.pdf [Online]

[133] Belkin, WeMo Switch, Available: http://www.belkin.com/us/p/P-F7C027/ [Online]

[134] The Energy Detective (TED) website [Online]. Available: www.theenergydetective.com.

[135] ThinkECO Inc website [Online]. Available: http://www.thinkecoinc.com/.

[136] Kill-A-Watt website [Online]. Available: http://www.p3international.com/products/special/p4400/p4400-ce.html.

[137] Plug Smart website [Online]. Available: http://www.plugsmart.com/.

[138] Alert Me website [Online]. Available: http://www.alertme.com/.

[139] Enistic website [Online]. Available: http://www2.enistic.com/.

[140] B. Li and L. Xia, "A multi-grid reinforcement learning method for energy conservation and comfort of HVAC in buildings," Automation Science and Engineering (CASE), 2015 IEEE International Conference on, Gothenburg, 2015, pp. 444-449.

[141] K. Dalamagkidis, D. Kolokotsa, K. Kalaitzakis, G.S. Stavrakakis, “Reinforcement learning for energy conservation and comfort in buildings,” Building and Environment 42 (2007) 2686–2698.

[142] P. Fazenda, K. Veeramachaneni, P. Lima and U. –M. O’Reilly, “Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems,” Journal of Ambient Intelligence and Smart Environments 6 (2014) 675–690.

[143] Gomes, C.; Silva, C.A., “Intelligent Energy Management Systems in Office Buildings”, [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/844820067124003 /Resumo%20Alargado.pdf.

[144] J. Zhu, F. Lauri, A. Koukam, V. Hilaire, "Managing ventilation systems for improving user comfort in smart buildings using reinforcement learning agents", Proceedings of European

Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2014), pp. 149-155.

[145] Simeng Liu, Gregor P. Henze, Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 1. Theoretical foundation, Energy and Buildings, Volume 38, Issue 2, February 2006, Pages 142-147.

[146] Simeng Liu, Gregor P. Henze, Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis, Energy and Buildings, Volume 38, Issue 2, February 2006, Pages 148-161.

[147] A. A. Beilskis, E. Guseinoviene, "Ambient Lighting Controller Based on Reinforcement Learning Components of Multi-Agent", Electronics and Electrical Engineering, vol. 121, no. 5, pp. 79-84, 2012.

[148] X. Pan and B. Lee, "An Approach of Reinforcement Learning Based Lighting Control for Demand Response," PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, Nuremberg, Germany, 2016, pp. 1-8.

[149] A. T. Kaliappan, S. Sathiakumar and N. Parameswaran, "Flexible power consumption management using Q learning techniques in a smart home," 2013 IEEE Conference on Clean Energy and Technology (CEAT), Lankgkawi, 2013, pp. 342-347

[150] F. Ruelens; B. J. Claessens; S. Vandael; B. De Schutter; R. Babuška; R. Belmans, "Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning," in IEEE Transactions on Smart Grid , vol.PP, no.99, pp.1-11

[151] H. Berlink, A. H. R. Costa, "Batch Reinforcement Learning for Smart Home Energy Management," Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), pp. 2561-2567.

[152] D. O'Neill, M. Levorato, A. Goldsmith and U. Mitra, "Residential Demand Response Using Reinforcement Learning," Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, Gaithersburg, MD, 2010, pp. 409-414.

[153] Z. Wen, D. O'Neill and H. Maei, "Optimal Demand Response Using Device-Based Reinforcement Learning," in IEEE Transactions on Smart Grid, vol. 6, no. 5, pp. 2312-2324, Sept. 2015.

[154] Rathnayaka, A.J.D.; Potdar, V.M.; Hussain, O.; Dillon, T., "Identifying prosumer's energy sharing behaviours for forming optimal prosumer-communities," *Cloud and Service Computing (CSC), 2011 International Conference on* , vol., no., pp.199,206, 12-14 Dec. 2011

[155] Rathnayaka, A.J.D.; Potdar, V.M.; Dillon, T.S.; Hussain, O.K.; Chang, E., "A Methodology to Find Influential Prosumers in Prosumer Community Groups," *Industrial Informatics, IEEE Transactions on* , vol.10, no.1, pp.706,713, Feb. 2014
[156] Ciuciu, I.G.; Meersman, R.; Dillon, T., "Social network of smart-metered homes and SMEs for grid-based renewable energy exchange," *2012 6th IEEE International Conference on Digital Ecosystems Technologies (DEST)*, pp.1-6, 18-20 June 2012
[157] Brusco, G.; Burgio, A.; Menniti, D.; Pinnarelli, A.; Sorrentino, N., "Energy Management System for an Energy District With Demand Response Availability," *IEEE Transactions on Smart Grid*, vol.5, no.5, pp.2385-2393, Sept. 2014
[158] Kahrobaee, S.; Rajabzadeh, R.A.; Leen-Kiat Soh; Asgarpoor, S., "A Multiagent Modeling and Investigation of Smart Homes With Power Generation, Storage, and Trading Features," *IEEE Transactions on Smart Grid*, vol.4, no.2, pp.659-668, June 2013
[159] Gamauf, T.; Leber, T.; Pollhammer, K.; Kupzog, F., "A generalized load management gateway coupling smart buildings to the grid," *AFRICON, 2011*, pp.1-5, 13-15 Sept. 2011
[160] Mocanu, E.; Aduda, K.O.; Nguyen, P.H.; Boxem, G.; Zeiler, W.; Gibescu, M.; Kling, W.L., "Optimizing the energy exchange between the Smart Grid and Building Systems," *Power Engineering Conference (UPEC), 2014 49th International Universities* , vol., no., pp.1,6, 2-5 Sept. 2014
[161] Safdarian, A.; Fotuhi-Firuzabad, M.; Lehtonen, M., "A Distributed Algorithm for Managing Residential Demand Response in Smart Grids," *IEEE Transactions on Industrial Informatics*, vol.10, no.4, pp.2385-2393, Nov. 2014
[162] ByungChul Kim; Lavrova, O., "Optimal power flow and energy-sharing among multi-agent smart buildings in the smart grid," *Energytech, 2013 IEEE*, vol., no., pp.1-5, 21-23 May 2013
[163] Roscia, M.; Longo, M.; Lazaroiu, G.C., "Smart City by multi-agent systems," *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*, pp.371-376, 20-23 Oct. 2013.
[164] Chourabi, H.; Taewoo Nam; Walker, S.; Gil-Garcia, J.R.; Mellouli, S.; Nahon, Karine; Pardo, T.A.; Scholl, Hans Jochen, "Understanding Smart Cities: An Integrative Framework," *2012 45th Hawaii International Conference on System Science (HICSS)*, pp.2289-2297, 4-7 Jan. 2012
[165] Ziming Zhu; Lambotharan, S.; Woon Hau Chin; Zhong Fan, "A Game Theoretic Optimization Framework for Home Demand Management Incorporating Local Energy Resources," *Industrial Informatics, IEEE Transactions on*, vol.11, no.2, pp.353-362, April 2015.
[166] Atzeni, I.; Ordonez, L.G.; Scutari, G.; Palomar, D.P.; Fonollosa, J.R., "Demand-Side Management via Distributed Energy Generation and Storage Optimization," *IEEE Transactions on Smart Grid*, vol.4, no.2, pp.866-876, June 2013

[167] He Chen; Yonghui Li; Louie, R.H.Y.; Vucetic, B., "Autonomous Demand Side Management Based on Energy Consumption Scheduling and Instantaneous Load Billing: An Aggregative Game Approach," *IEEE Transactions on Smart Grid*, vol.5, no.4, pp.1744-1754, July 2014

[168] Soliman, H.M.; Leon-Garcia, A., "Game-Theoretic Demand-Side Management With Storage Devices for the Future Smart Grid," *IEEE Transactions on Smart Grid*, vol.5, no.3, pp.1475-1485, May 2014.

[169] Kai Ma; Guoqiang Hu; Spanos, C.J., "Distributed Energy Consumption Control via Real-Time Pricing Feedback in Smart Grid," *IEEE Transactions on Control Systems Technology*, vol.22, no.5, pp.1907-1914, Sept. 2014

[170] Tsung-Hui Chang; Alizadeh, M.; Scaglione, A., "Real-Time Power Balancing Via Decentralized Coordinated Home Energy Scheduling," *IEEE Transactions on Smart Grid*, vol.4, no.3, pp.1490,1504, Sept. 2013.

[171] Kawashima, H.; Kato, T.; Matsuyama, T., "Distributed mode scheduling for coordinated power balancing," *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp.19-24, 21-24 Oct. 2013

[172] Salinas, S.; Ming Li; Pan Li, "Multi-Objective Optimal Energy Consumption Scheduling in Smart Grids," *IEEE Transactions on Smart Grid*, vol.4, no.1, pp.341-348, March 2013

[173] Zhao Tan; Peng Yang; Nehorai, A., "An Optimal and Distributed Demand Response Strategy With Electric Vehicles in the Smart Grid," *IEEE Transactions on Smart Grid*, vol.5, no.2, pp.861-869, March 2014

[174] Peng Yang; Chavali, P.; Gilboa, E.; Nehorai, A., "Parallel Load Schedule Optimization With Renewable Distributed Generators in Smart Grids," *IEEE Transactions on Smart Grid*, vol.4, no.3, pp.1431-1441, Sept. 2013

[175] Adika, C.O.; Lingfeng Wang, "Demand-Side Bidding Strategy for Residential Energy Management in a Smart Grid Environment," *IEEE Transactions on Smart Grid*, vol.5, no.4, pp.1724-1733, July 2014

[176] Gatsis, N.; Giannakis, G.B., "Residential Load Control: Distributed Scheduling and Convergence With Lost AMI Messages," *IEEE Transactions on Smart Grid*, vol.3, no.2, pp.770-786, June 2012

[177] Rahbari-Asr, N.; Ojha, U.; Ziang Zhang; Mo-Yuen Chow, "Incremental Welfare Consensus Algorithm for Cooperative Distributed Generation/Demand Response in Smart Grid," *IEEE Transactions on Smart Grid*, vol.5, no.6, pp.2836-2845, Nov. 2014

[178] Rahimiyan, M.; Baringo, L.; Conejo, A.J., "Energy Management of a Cluster of Interconnected Price-Responsive Demands," *IEEE Transactions on Power Systems*, vol.29, no.2, pp.645-655, March 2014

[179] Kunwar, N.; Yash, K.; Kumar, R., "Area-Load Based Pricing in DSM Through ANN and Heuristic Scheduling," *IEEE Transactions on Smart Grid*, vol.4, no.3, pp.1275-1281, Sept. 2013

[180] Dong-Min Kim; Jin-O Kim, "Design of Emergency Demand Response Program Using Analytic Hierarchy Process," *IEEE Transactions on Smart Grid*, vol.3, no.2, pp.635-644, June 2012

[181] Xiao-Min Hu; Zhi-Hui Zhan; Ying Lin; Yue-Jiao Gong; Yu Wei-Jie; Yao-Xiu Hu; Jun Zhang, "Multiobjective genetic algorithm for demand side management of smart grid," *2013 IEEE Symposium on Computational Intelligence in Scheduling (SCIS)*, pp.14-21, 16-19 April 2013

[182] Colombo, G.; Ferrero, F.; Pirani, G.; Vesco, A., "Planning Local Energy Communities to develop low carbon urban and suburban areas," *2014 IEEE International Energy Conference (ENERGYCON)*, pp.1012-1018, 13-16 May 2014

[183] Strobbe, M.; Vanthournout, K.; Verschueren, T.; Cardinaels, W.; Develder, C., "Large-scale residential demand response ICT architecture," *2014 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp.1-6, 12-15 Oct. 2014

[184] Dimeas, A.; Drenkard, S.; Hatziargyriou, N.; Karnouskos, S.; Kok, K.; Ringelstein, J.; Weidlich, A., "Smart Houses in the Smart Grid: Developing an interactive network.," *IEEE Electrification Magazine*, vol.2, no.1, pp.81-93, March 2014

[185] Wei Wu; Aziz, M.K.; Hantao Huang; Hao Yu; Hoay Beng Gooi, "A real-time cyber-physical energy management system for smart houses," *2011 IEEE PES Innovative Smart Grid Technologies Asia (ISGT)*, pp.1-8, 13-16 Nov. 2011

[186] Kahrobaeian, A.; Ibrahim Mohamed, Y.A.-R., "Networked-Based Hybrid Distributed Power Sharing and Control for Islanded Microgrid Systems," *IEEE Transactions on Power Electronics*, vol.30, no.2, pp.603-617, Feb. 2015

[187] Weigang Zhong; Zhichuan Huang; Ting Zhu; Yu Gu; Qingquan Zhang; Ping Yi; Dingde Jiang; Sheng Xiao, "iDES: Incentive-driven distributed energy sharing in sustainable microgrids," *2014 International Green Computing Conference (IGCC)*, pp.1-10, 3-5 Nov. 2014

[188] Meegahapola, L.G.; Robinson, D.; Agalgaonkar, A.P.; Perera, S.; Ciufo, P., "Microgrids of Commercial Buildings: Strategies to Manage Mode Transfer From Grid Connected to Islanded Mode," *IEEE Transactions on Sustainable Energy*, vol.5, no.4, pp.1337-1347, Oct. 2014

[189] Khamphanchai, W.; Saha, A.; Rathinavel, K.; Kuzlu, M.; Pipattanasomporn, M.; Rahman, S.; Akyol, B.; Haack, J., "Conceptual architecture of building energy management open source software (BEMOSS)," *2014 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp.1-6, 12-15 Oct. 2014.

[190] BEMOSS Continuation Report to DOE, August 2014.

[191] BEMOSS Quarterly Presentation to DOE, June 25, 2015.

[192] Richard S. Sutton; Andrew G. Barto, “Reinforcement Learning: An Introduction”, MIT Press, Cambridge, MA, 1998.

[193] The Standards for Ventilation and Indoor Air Quality, ANSI/ASHRAE Standard 62.1-2013, American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) [Online]. Available: https://www.ashrae.org/resources--publications/bookstore/standards-62-1--62-2. Retrieved: February 2016.

[194] P. Ple Fanger (1970). Thermal Comfort: Analysis and applications in environmental engineering. McGraw-Hill.

[195] T. Hoyt, S. Schiavon, A. Piccioli, D. Moon, and K. Steinfeld, CBE Thermal Comfort Tool. Center for the Built Environment, University of California Berkeley [Online]. Available: http://cbe.berkeley.edu/comforttool/. Retrieved: February 2016.

[196] Hoyt, T., E. Arens, and H. Zhang., "Extending air temperature setpoints: Simulated energy savings and design considerations for new and retrofit buildings", Building and Environment, September, 2014. Online: https://escholarship.org/uc/item/13s1q2xc.

[197] The Lighting Handbook, 10th Edition, Illuminating Engineering Society of North America (IESNA) [Online]. Available: http://www.ies.org/handbook/. Retrieved: Jan. 2016.

[198] Goetzler, W. G., Shalom, S. Jasinski, R. Legett, H. Lisle, A. Marantan, M. Millard, D. Pinault, D. Westphalen, and R. Zogg., 2009. Energy Savings Potential and R&D Opportunities for Commercial Refrigeration: Final Report. Washington, DC: Navigant Consulting, Inc. September 23.
http://apps1.eere.energy.gov/buildings/publications/pdfs/corporate/commercial_refrig_report_10-09.pdf

[199] Saha, A.; Kuzlu, M.; Pipattanasomporn, M.; Rahman, S.; Elma, O.; Selamogullari, U.S.; Uzunoglu, M.; Yagcitekin, B., "A Robust Building Energy Management Algorithm Validated in a Smart House Environment," *Intelligent Industrial Systems*, DOI 10.1007/s40903-015-0004-y, May 2015.

[200] Liu, H.; Ma, H.; Zarki, M. E.; and Gupta, S.; "Error Control Schemes for Networks: An overview," in Mobile Networks and Applications 2, 1997, pp. 167-182.

[201] Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y.; and Cayirci, E.; "A Survey on Sensor Networks," in IEEE Comm. Magazine, 2002, vol. 40, no. 8, pp. 102-114.

[202] Lin, S.; Costello, D. J.; and Miller, M. J.; "Automatic-Repeat-Request Error-Control Schemes," in IEEE Comm. Magazine, 1984, vol. 22, no. 12, pp. 5-17.

[203] Saha, A.; Kuzlu, M.; Pipattanasomporn, M., "Demonstration of a home energy management system with smart thermostat control," *2013 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp.1-6, 24-27 Feb. 2013

[204] Saha, A.; Kuzlu, M.; Khamphanchai, W.; Pipattanasomporn, M.; Rahman, S.; Elma, O.; Selamogullari, U.S.; Uzunoglu, M.; Yagcitekin, B., "A home energy management algorithm in a smart house integrated with renewable energy," *2014 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp.1-6, 12-15 Oct. 2014

[205] Shao, Shengnan. An approach to demand response for alleviating power system stress conditions due to electric vehicle penetration. Diss. Virginia Polytechnic Institute and State University, 2011

[206] RELOAD Database Documentation and Evaluation and Use in NEMS [Online]. Available: http://www.onlocationinc.com/LoadShapes-Reload 2011.pdf

[207] “Median and average square feet of floor area in new single-family houses completed by location,” [Online]. Available: http://www.census.gov/const/C25Ann/sftotalmedavgsqft.pdf.

[208] “Development of A Smart Grid-Compatible Smart house Prototype with the aims of Improving Energy Efficiency and Lowering Carbon Emission”, Available online: http://www.akilliev.yildiz.edu.tr/, accessed Aug 2013.

[209] 4-noks® ZR-PLUG-XX-M Smart plugs. Available online: http://www.zb-connection.com/zigbee-wireless-plug-play-network/devices-for-energy-saving/zigbee-electricity-meter-and-remote-control-plug, accessed Jan 2015.

[210] 4-noks® ZR-HMETER.D-M Energy meter on din rail. Available online: http://www.zb-connection.com/zigbee-wireless-plug-play-network/devices-for-energy-saving/energy-meter-on-din-rail, accessed Jan 2015.