The OpenADR standard and development of new Demand Response algorithms in the Smart Grid

Konstantinos Koliopoulos

SID: 3301120008

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

NOVEMBER 2013

THESSALONIKI – GREECE

The OpenADR standard and development of new Demand Response algorithms in the Smart Grid

Konstantinos Koliopoulos

SID: 3301120008

Supervisor:	Dr. Costas Tzaras
Supervising Committee Members:	Prof. Ioannis Vlahavas Dr. George Koutitas

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

NOVEMBER 2013

THESSALONIKI – GREECE

Abstract

This dissertation was written as a part of the MSc in ICT Systems at the International Hellenic University. This dissertation can be dissected into two parts. The first part includes chapters one to five and is a rich bibliographical research about Smart Grid, Demand Response, Demand Response standards and Admission Control. The second part includes chapters six to eight and is about the proposed by this work Demand Response algorithm for grid reliability using Admission Control for a fair scheme.

The first chapter is a short introductory chapter that elaborates upon the Thesis motivation, contribution and organization. Chapters two to five are a thorough bibliographical research about Smart Grid, Demand Response, Demand Response standards and Admission Control. More specifically, the current state of affairs of the Smart Grid and Demand Response are showcased. Also, the state of the art Demand Response standards are analyzed in depth and various implementation cases are explored. Finally, the theory of Admission Control algorithms is documented.

Chapters six to eight are about the proposed by this work Demand Response algorithm for grid reliability using Admission Control for a fair scheme. More specifically, a complete description and mathematical foundation is offered. Also, the simulation of the proposed algorithm is showcased and finally an in depth discussion about the results derived from the simulation is held.

Konstantinos Koliopoulos

06/11/2013

To Despoina

To the universe

"*Concepts that have proven useful in ordering things easily achieve such authority over us that we forget their earthly origins and accept them as unalterable givens.*"

"Ἀποδεδειγμένων τινῶν ἐννοιῶν ὠφελίμων εἰς τὴν τῶν πραγμάτων ταξινόμησιν καὶ ὡς ἐκ τούτου ἀρχομένων τοῦ συνειδότος ἡμῶν ὥστε ἐπιλανθάνειν ἡμᾶς τῆς ἐκ τῆς γῆς προϊούσης αὐτῶν φύσεως, τυγχάνει συνήθης ἡ ἀποδοχή των ὡς ἀναλλοιώτων καταλειπομένων."

Albert Einstein

Dedications and acknowledgments

I would like to dedicate this work to my wife, Dr. Despoina Kassa Koliopoulou, DDS, MDentSci, for unconditionally supporting all my endeavors and for giving me the strength to break the limits of myself and stride above and beyond.

I would like to sincerely express my gratitude towards the School of Science and Technology for harboring my academic and scientific quests and for supporting my studies by awarding me the compensatory scholarship of the School for the academic year 2012 - 2013.

I would like to express my appreciation to my supervisor Dr. Costas Tzaras, professor at the School of Science and Technology of the International Hellenic University, for his trust and support. Also, I would like to express my appreciation to Dr. George Koutitas, academic assistant at the School of Science and Technology of the International Hellenic University, for his guidance, support, patience and for providing valuable suggestions, thought provoking comments and motivation throughout the duration of this dissertation as well as throughout the duration of this Masters course.

Also, I would like to convey special thanks and deep appreciation to Dr. Costas Psychalinos, assistant professor at the Physics Department of the University of Patras, for being a wonderful and inspiring teacher and advisor, for engulfing my first scientific quests and for forging into me the lifelong lesson of searching the physical meaning behind any phenomenon.

Moreover, I would like to thank Prot. Charalampos Karadimos for utilizing his expertise in ancient Greek in order to provide the translation of Einstein's phrase from English to ancient Greek.

Last but not least, I wish to thank my mother, Artemis Kouremeti, my father, George Koliopoulos, and my sister, Dr. Maria Koliopoulou. For very different reasons, each one of them is a raw model for me.

Contents

1 Introduction

In this short introductory chapter the motivation for this Thesis well as the contribution and organization of this work are discussed.

1.1 Thesis Motivation

The Smart Grid and Demand Response is a revolution that the globe will witness in the near future. It is a radical change made possible by resend ICT advances and forged under the threat of climatic changes.

In a few words, the Smart Grid enable Demand Response to provide automation, intelligence and control over the power grid network under the umbrella of various Demand Response standards.

The motivation of this thesis is the exploration of ICT for Green and especially the Smart Grid, the Demand Response and Demand Response Standards. Also, we employ the newfound Demand Response framework and combine it with well-established control mechanisms from the cellular networks. This result in the proposal of a novel algorithm that can provide grid reliability.

1.2 Thesis contribution

The contribution of this Thesis is the creation of a novel Demand Response algorithm that utilizes Admission Control in order to provide grid reliability. Admission Control is a very effective way for utilizing the resources that a network provides and is heavily used in mobile cellular networks.

This is the first time in the literature that an algorithm is proposed for power grid networks and operates under the Demand Response framework and utilizes Admission Control in order to provide grid reliability. The proposed algorithm includes three different implementation schemes and a plethora of scenarios that all provide grid reliability and fairness for the actors that partake in it.

For the proposed algorithm we provide an extensive description and a complete mathematical foundation. Also, we simulate the proposed algorithm and extract all valuable performance metrics that are needed in order to evaluate the effectiveness and performance of the proposed algorithm in all of its implementation schemes. Finally we provide an in-depth analysis of the extracted performance metrics and compare all the implementation schemes in order to provide the best solution in respect to the actors that partake in the algorithm.

1.3 Thesis organization

This work can be dissected into eight chapters. The first chapter is an introduction. Chapters two to five compose a rich bibliographical research about Smart Grid, Demand Response, Demand Response Standards and Admission Control. Chapters six to eight provide the description, analysis, simulation and evaluation of the proposed Demand Response algorithm.

Chapter two is about the Smart Grid. More specifically, the incentives that lead to the development of the Smart Grid are showcased as well as the solutions and advantages that the Smart Grid provides. Later on, a discussion about the drivers, the goals and the evolution of the Smart Grid is held. Finally, the role of ICT, the Smart Grid architecture, the entities that partake in it and the cost of the Smart Grid are analyzed.

Chapter three is about Demand Response. More specifically, the need for Demand Response is showcased as well as an overview and a definition of Demand Response. Later on, a discussion about the types of loads and Comfort as a metric is held. Also, Demand Response mechanisms, Demand Response architecture and programs are analyzed. Next, the Demand Response benefits are thoroughly explained. Finally, the limitations of Demand Response and various security issues are examined.

Chapter four is about Demand Response standards. More specifically, the OpenADR standard is thoroughly showcased as well as its history definition and deployment status. Later on, a discussion about the evolution of the OpenADR standard, its actors and services is held. Also, the OpenADR standard profiles and events are analyzed. Next, the OpenADR standard security and certification are thoroughly explained. Moreover, the OpenADR standard communication architecture and implementation configuration are examined. Later on, the OpenADR standard Demand Response programs are explained. Next, the residential case in the OpenADR standard implementation is showcased as well as the commercial case. Finally, the OpenADR 2.0 A and B profile are thoroughly discussed. Apart from the OpenADR standard, the SEP standard is also briefly examined and the differences between the OpenADR standard and the SEP standard are pinpointed.

Chapter five is about Admission Control. A thorough discussion about Admission Control algorithms is held.

Chapter six is about the proposed by this Thesis Demand Response algorithm for grid reliability using Admission Control for a fair scheme. More specifically, the actors are showcased as well as the modelling of the Subscribers, the Utility and the Demand Response Automation Server. Later on, a discussion about other rules and about providing fairness and defining the optimum utility function for the Utility and the Subscribers is held. Also, Pre-emptive and non pre-emptive Demand Response events are analyzed. Next, the proposed algorithm flowchart is thoroughly explained. Finally, the proposed algorithm pseudocode is examined.

Chapter seven is about the simulation and results of the proposed algorithm. More specifically, the simulation of the Subscribers is showcased as well as the simulation the Utility and the Demand Response Automation Server. Finally, the results for all extracted performance metrics are explained.

The last chapter, chapter eight, is about the conclusions drawn from the simulation of the proposed algorithm and about future work. More specifically, the conclusions derived from all extracted performance metrics are thoroughly discussed and explained. Finally, a discussion about future work is held.

2 Smart Grid

As the global community faces critical problems like global warming, depletion of traditional energy sources and continuous raise of the value of electricity, the industry is forced to react. It is now, more than ever, more obvious that there is an evolution on the electric industry, that a new energy paradigm is emerging [9]. The vessel of this evolution is the Smart Grid. The Smart Grid will fundamentally transform the traditional power grid network in an attempt to meet the challenges that threaten the global lifestyle [3, 10]. It has been foreseen that the design and implementation of this next-generation smart, green grid will occur in the next decade and will create infrastructure that will last for the next century and beyond [10].

In this chapter we will attempt to provide a deep understanding of the inherited flaws of today's power green network, the incentives that lead to the creation of the Smart Grid, the solutions and advantages that it offers and the drivers, goals and evolution of it. Furthermore, we will provide an exhausting analysis on the role and cost of ICT, the architecture of the Smart Grid and the new entities that emerge from it.

2.1 Towards the Smart Grid, incentives, solutions and advantages

The current state of the power grid network consists of its traditional and well established components. Power is generated at the power plants mostly by employing thermal energy by burning fossil fuels, coal or natural gas [1, 10]. Harvesting power by renewable energy sources (*RES*) (hydroelectric, wind turbines, solar power) or nuclear energy sources is utilized way less than burning fossil fuels. Then, the power is transmitted to the consumers via a transmission and distribution network by using high voltage and medium voltage stations respectively. The consumer receives and uses power at low voltage. Also, the power plants generate alternating current (*AC*) which is transformed to direct current (*DC*) within the supply unit of the equipment. All voltage transformations, from high voltage that is generated at the power plants and is transferred via the transmission network, to medium voltage that is transferred via the distribution network to the consumer that receives low voltage, are performed by transformers [1, 3]. Figure 2.1 shows the architecture of the power grid network.

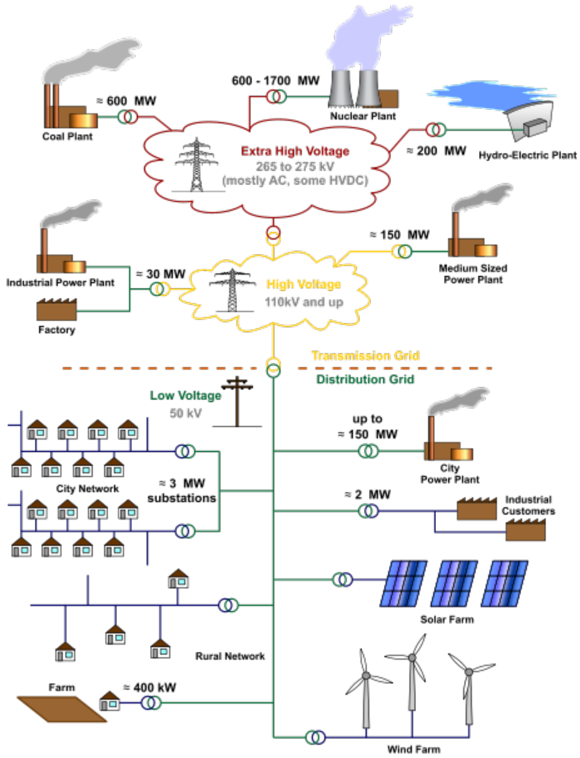

Figure 2.1: *The power grid network (Source: MBison under Wikipedia Creative Commons)*

Today's power grid network implements an one-way flow of electricity. It is centralized, with bulk production, designed for peak production and uses mainly coal and natural gas to produce power [1, 10]. Thus, it is responsible for 40% of human caused CO_2 production. Also, the power generation is controllable, with predictable load but with limited automation and situational awareness [1, 3, 10]. Finally, there is a plethora of customized proprietary systems but lack of customer-side data to manage and reduce energy use [3, 8]. Figure 2.2 shows today's power grid network.

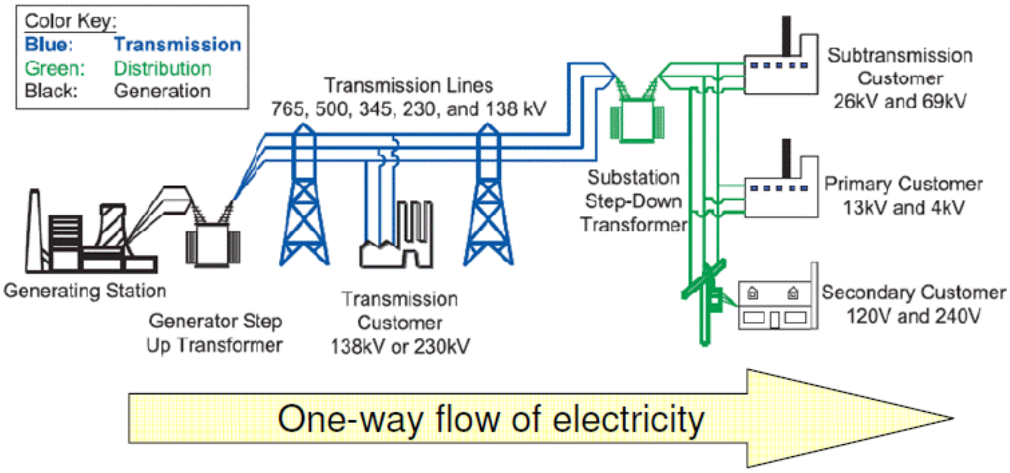

Figure 2.2: *Today's power grid network [3]*

The process of transmission and conversion of power from the power plants to the consumer has inherited flaws: the transmission and the conversion losses [1, 10]. This results to about 33% efficiency of the electricity supply system [1, 10].

In an attempt to overcome those drawbacks, a multidimensional and interdisciplinary solution has been proposed that addresses different aspects of the cause of the problems. By producing power locally at medium or low voltages the conversion and transmission losses will be minimum [1, 10]. By giving intelligence to the network in order to adapt consumption according to production, the gains by harvesting time variant renewable energy sources will be maximum. Finally, by storing power to battery banks or electric vehicles (*EVs*) and by giving intelligence to the network, the power consumption adaptation will be more effective [1, 3, 10].

Summarizing the above, two are the key elements for providing a solution. Give intelligence and automation [1, 3]. Thus, the term *Smart Grid* emerges. The Smart Grid is the next generation of power grid network. The Smart Grid will transform the current hierarchical energy generation and transition network to a distributed system [1, 10]. In the same time, intelligence and automation will give to the Smart Grid self organizing properties. The Smart Grid will be a Self Optimized Network (*SON*) capable to self adapt to load changes, self optimized in new changes and self heal from faults [1]. The Smart Grid will use protocols and hardware that integrates information technology and advanced communications into the power grid in order to increase system efficiency and cost effectiveness, provide customer tools to manage energy use, improve reliability, resiliency and power quality and enable use of innovative technologies including renewable energy sources, storage and electric vehicles within a multi-stakeholder interactions scheme [3, 8, 9].

The Smart Grid will utilise a two-way flow of electricity and information by combining the power grid network with a communication network. This combination requires interoperability. Thus, there is a need for reliable standards and validated performance [3, 8]. Figure 2.3 shows the two-way flow of electricity and information in the Smart Grid.

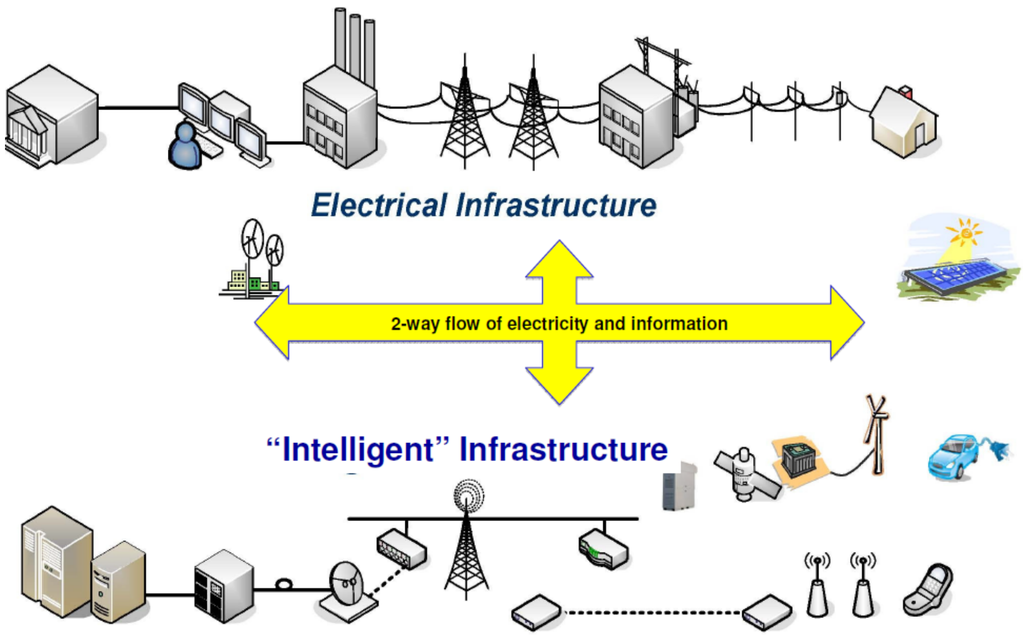

Figure 2.3: *Two-way flow of electricity and information in the Smart Grid [3]*

The Smart Grid advantages are:

● Decreases black out likelihood by rendering the power grid more stable [5].

● Decreases congestion by reconfiguring the network topology or control transmission line properties [5].

● Decreases distribution operation and maintenance by anticipating asset failure, reduce ware and tear and remote monitoring of consumption and electric service level management [5].

● Increases resilience of transmission and distribution network to load growth by integrating renewable energy sources [5].

● Creates savings to consumers by regulating their energy consumption and by creating energy efficiency programs and peak reduction incentives. Thus, this eventually leads to reduced prices [6,7].

● Reduces the production of Green House Gases by harvesting renewable energy sources and by increasing the efficiency of existing power generation sources [7].

● Renders the grid more efficient, self healing and resilient [7,9].

● Enables Service and Energy Providers to control the consumption in order to reduce the need to peek energy production [7].

● Enables Service and Energy Providers to connect supply and demand with pricing in a dynamic manner [7].

● Enhance national security by improving safety, security and reliability [7].

● Enhance control by monitoring the voltage, current and frequency [7].

● Enables load side and distributed resources to become responsive to system needs and thus realize their significant potential to decrease costs [5].

● Realizes system-integration-based value adding opportunities that exist already or will avail themselves in the near future as distributed generation, electric vehicles, and other distributed resources [5].

Many formal definitions have been given to the Smart Grid such those that follows:

"A Smart Grid is an advanced system that incorporates widely distributed intelligent sensors and employs real-time communications to automatically sense and correct inefficiencies and disturbances on the electrical distribution system. Smart Grid's products help transform the existing grid into a self-healing, self-optimizing 21st century power system capable of supporting the wide spread use of renewables." [1]

"A Smart Grid is an electrical grid that communicates information to automatically act on supply and demand thus improving the efficiency, reliability, economics, and sustainability of the production and distribution of electricity." [2]

2.2 Smart Grid drivers, goals and evolution

The main drivers behind the realisation of the Smart Grid are limiting the emission of Green House Gases (*GHG*) and preventing climate change, providing energy security and prevention of mass blackouts, protection of the global lifestyle that depends on electricity and the creation of jobs. Thus, the goals that are set are reducing overall energy use and increase grid efficiency, increase use of renewable energy sources and reduce dependence on petroleum, support shift from oil to electric transportation and generally enhance reliability and security of the power grid network [3].

The evolution of the Smart Grid will lead towards:

- High use of renewable energy sources (20% by 2020, 20-20-20 Goal [1]) [3].
- Distributed power generation and creation of microgrids [3].
- Selling locally generated power into the grid [3].
- Distributed storage [3].
- Use of smart meters that provide near-real time usage data [3].
- Dynamic pricing [3].

● Use of smart appliances communicating with the grid [3].

● Use of energy management systems in homes, commercial and industrial facilities linked to the grid [3].

● Use of plug-in electric vehicles [3].

● Use of networked sensors and automated controls throughout the grid [3].

● Increase security infused into all Smart Grid functions [3].

The evolution of the Smart Grid is a multistage process that requires creative thinking and reengineering of system operations and energy markets with the use of ICT. First the existing electricity systems needs to be inspected and then the potential of ICT needs to be analysed. Afterwards, the technical and market processes will be redefined by implementing ICT. Thus, this process leads to smart systems [6]. Figure 2.4 shows the evolution of the Smart Grid.

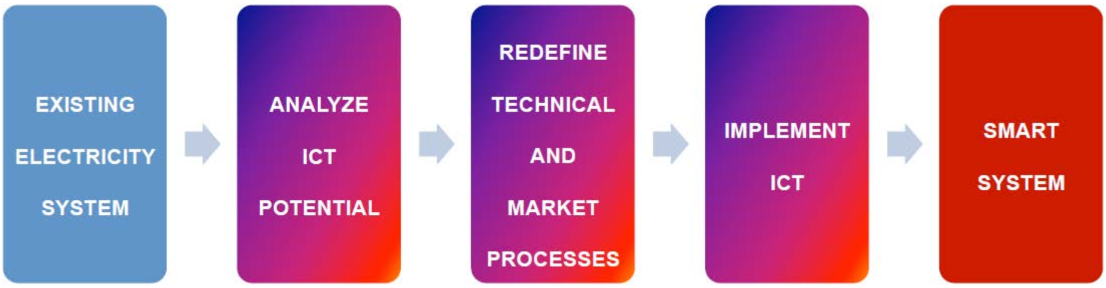

Figure 2.4: *The evolution of the Smart Grid [6]*

2.3 The role of ICT, Smart Grid architecture, entities and cost

The role of ICT in the Smart Grid is to act as an add on layer to the existing power network and provide intelligence and automation. Intelligence is provided by collecting data from deployed sensor networks that consist an Advanced Metering

Infrastructure (*AMI*) enabled by Machine to Machine (*M2M*) communication. The harvested data is processed and analyzed with Data Mining and Big Data techniques. Automation is provided via a Demand Response scheme that provides optimization, scheduling and control [1].

The general description of the architecture of the Smart Grid is as follows. A Smart Building has a communication infrastructure that enables an Agent (CPU unit) to collect data from sensors and interconnected appliances that consist an Advanced Metering Infrastructure enabled by Machine to Machine communication and sends them to an Aggregator. The Aggregator acts as a supervisor of a subset of Agents and is governed by the Service and Energy Provider that does the management. The Aggregator is connected to the Service and Energy Provider via a cloud service [1]. Figure 2.5 shows the control elements of the Smart Grid.

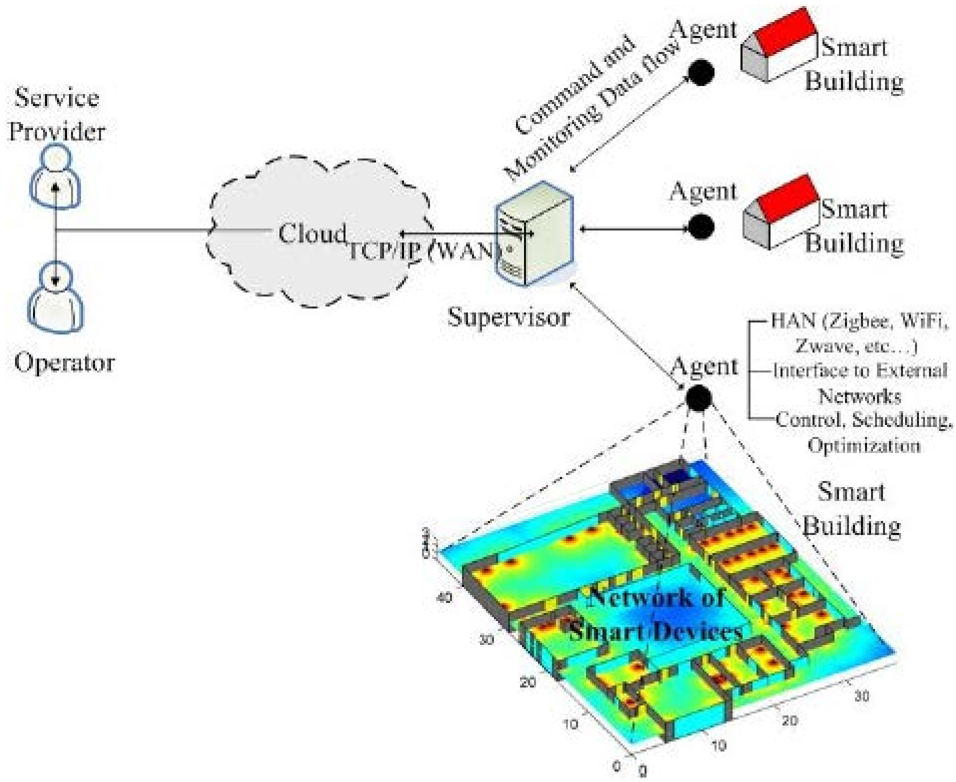

Figure 2.5: *The Smart Grid Control Elements [1]*

The Agent is composed of a CPU unit that enables data processing and of a reactive and cognitive software that handles the communication with other Agents and the User Interface (*UI*). The Agent also acts as a gateway that bridges the Home Area Network (*HAN*) with the Aggregator and receives commands and pricing signals and forwards commands to appliances and actuators [1].

The Aggregator is composed of a CPU or Virtual CPU unit that enables data processing. The Aggregator acts as a first stage of Decision Making and pre-processing of commands and Demand Response signals. The Aggregator also communicates with Agents and forwards commands to them sent by the Service and Energy Provider [1].

Other Smart Grid Entities are:

● Prosumer: A home user of the power grid network that simultaneously consumes and produces energy. In most cases the produced energy is by harvesting renewable energy sources. Thus, the consumption and the production of the energy are time variant. There are two modes of operation for the Prosumer, the Producer mode when production is greater than consumption and the Consumer mode when the consumption is greater than the production [1].

● Distributed Generation: Small generation of power at medium or low voltages that usually feeds a microgrid by harvesting renewable energy sources. Distributed Generation units can be located closer to consumers, minimizing transmission losses [1, 10].

● MicroGrid: Is composed of Distributed Generation units, storage units and consumers and is an automatically coordinated and self optimized subset of the network that incorporates all the necessary ICT infrastructure in order to be independent of the power grid network. The MicroGrid is in Island Mode when the production of the Distributed Generation units is greater than the total consumption of the consumers [1].

● Virtual Power Plant: A group of Distributed Generation units without a specific formation or even geographical proximity that are controlled by a single entity.

A Virtual Power Plant can be a group of Prosumers if and only if the total production is greater than the total consumption of energy [1].

● Open Energy Market: Real time pricing with dynamic electricity price according to production and consumption [1].

● Smart Building, Smart House: An intelligent building or house that provides end user interface to the Smart Grid via a bi-directional communication channel [1]. Also a Smart Building or Smart House provides some degrees of freedom for energy management and interconnects smart devices within the Home Energy Management Network. The Smart Building and Smart House is composed by many Smart Devices like smart metering devices, utility HAN devices, energy gateways, air-conditioning and hot water, white appliances, consumer electronics, home controls, energy displays, pools and spas and electric vehicles [4]. Figure 2.6 shows a Smart House.

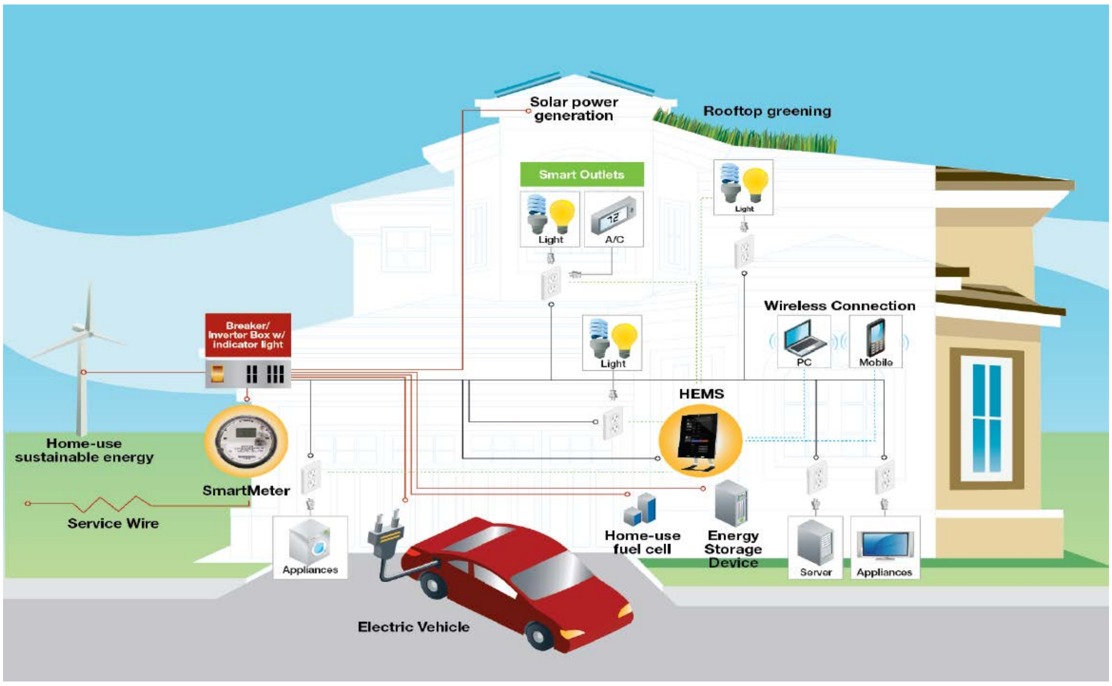

Figure 2.6: *A Smart House [9]*

● Advanced Metering Infrastructure: Is a sub domain of Machine to Machine communications specialized for the Smart Grid, Smart Building and Smart House. An

Advanced Metering Infrastructure interconnects all the entities of the Smart Grid, coordinates the Agents, provides bi-directional data flow from the Service and Energy Provider to the consumer and enhances the interconnected machines with Self Optimized Networks capabilities. An Advanced Metering Infrastructure has two layers, a Home Layer for command and data flow within the Home Area Network and a Grid Layer for command and data flow within the Grid Network [1]. Figure 2.7 shows the conceptual model of the Smart Grid.

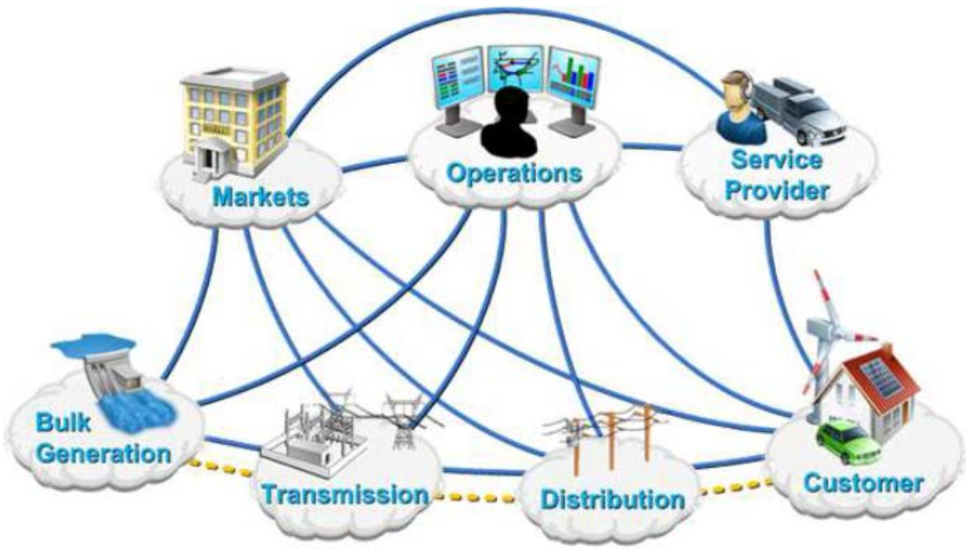

Figure 2.7: *The Smart Grid conceptual model [17]*

By inserting all those entities and complexity to the existing system, the Smart Grid in order to be implemented requires a considerable capital investment. An obvious question is who pays for the Smart Grid. The extra ICT cost that is required for the realization of the Smart Grid is provided by the avoidable cost that after the implementation of the Smart Grid is avoided [6].

3 Demand Response

Part of the evolution of the electric industry is Demand Response. Historically demand side management was centrally administered by the supply side but new decentralized demand side participation will become the new paradigm for power systems [13, 21]. Demand side participation came as the industry's response to increased pressure to increase competition, reduce market power, improve reliability and enable the use of cleaner renewable energy sources [13]. Demand Response has been envisioned to deal with unexpected supply limit events and increase reliability by shifting system load and balancing supply and demand [14, 17]. Developments in Smart Grid and Advanced Metering Infrastructure are eliminating the final technological barriers that prevent demand from participating in power markets and load management [21]. Thus, under the implementation of the Smart Grid, Demand Response will enhance demand side participation [13].

In this chapter we will attempt to provide a thorough explanation of the reasons that lead to the development of Demand Response and an overview and definition of Demand Response. Additionally, we will attempt to provide a deep analysis of the types of load that Demand Response can control as well as the Demand Response mechanisms, architecture and programs. Finally, we will focus on Demand Response benefits, limitations and security issues.

3.1 The need for Demand Response, an overview and definition

Demand Response is a functionality within the context of the Smart Grid in which the load can be managed by the demand side. With Demand Response, the demand side can manage the load requirements in ways that an upper threshold is never exceeded and the load profile across time is smooth [11]. Demand Response is a mechanism for achieving energy efficiency through managing customer consumption of electricity in response to supply conditions [12].

Demand Response can take advantage of the intelligence, automation and advanced communication equipment of the Smart Grid so that control and scheduling of the load can be implemented. Within the Smart Grid and with Demand Response, part of the demand can be shut down, non-emergency tasks can be rescheduled to off-peak times while taking advantage of stored energy [11, 12]. Figure 3.1 shows the functionality of Demand Response.

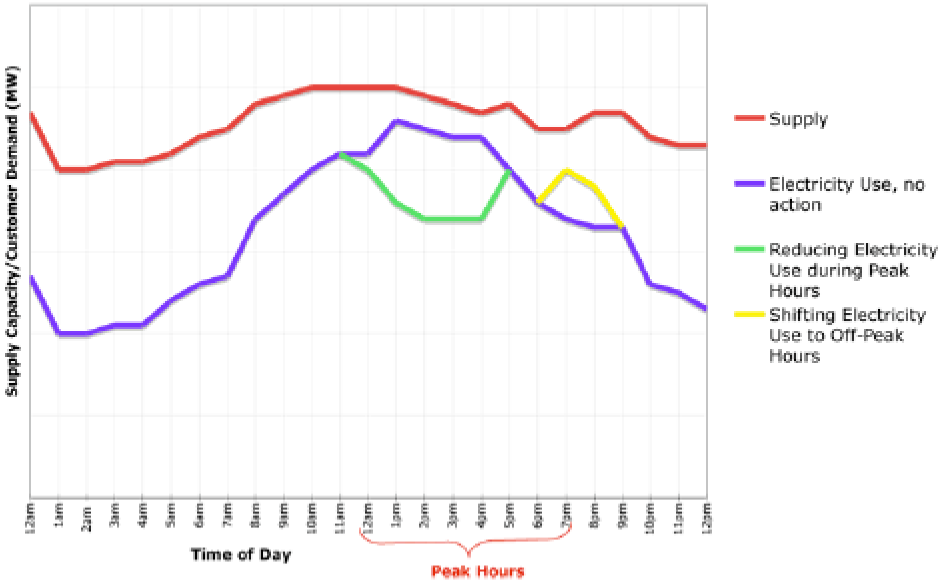

Figure 3.1: *Demand Response functionality [11]*

The need for Demand Response derives from the fact that the cost for electricity is a convex function of the load and that system capacity is a time varying function. Thus, there is the need to adapt consumption to production. With Demand Response, electricity cost by dynamic pricing can be minimum and island mode with renewable energy sources can be achieved [11]. Also, imbalances in the demand may cause grid reliability issues and energy price fluctuations. Thus, there is the need for actively balancing the demand through Demand Response [17].

Many formal definitions have been given to Demand Response such as those that follows:

"*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.*" [11, 15]

"*Demand Response: relates to any program which encourages shift of (demand) of energy by end consumers. The participation of the end customers is a response to factors such as incentive pricing, new tariff schemes, greater awareness and an increased sense of responsibility. The end consumers agree to involvement, but their participation may involve either active behavioural changes or passive responses, through the use of automation.*" [23]

All participants of Demand Response events agree under contracts with utilities to execute various Demand Response control commands so as to shape the demand and thus gain financial benefits through incentives or reductions in electrical cost [16].

Demand Response can potentially improve operational efficiency and capital efficiency, reduce harmful emissions and risk of outages and better match energy demand with changes in energy generation [22].

3.2 Types of loads and Comfort as a metric

Three types of load can be identified within the Smart House and the Smart Building. The Standard Loads, the Flexible Loads and the Elastic Loads.

The Standard Loads are loads that are vital for the operation of a house or building and for the quality of living and entertainment. All necessary and entertainment appliances, like lights and home cinema units can be characterized as Standards

Loads. Standard Loads usually create a constant background on power load or flat additional power levels [11].

The Flexible Loads are loads that are usually related to thermostatic appliances like air condition units, ovens and hotplates. Flexible Loads are usually related to the high power needs of the house and characterize cooling, heating and food preparation and conservation uses. Thus, their operations is correlated to a comfort index. The comfort index can be regarded to be the objective of operation of the appliance [11].

In the case of Flexible Loads, comfort is related to four parameters. A physical value a, a threshold value μ and two time thresholds, $T_{min}(a, \mu)$ and $T_{max}(a, \mu)$. T_{min} describes the minimum time needed to reduce a hypothetical discomfort metric, a, to the wanted value while T_{max} describes the time needed for the discomfort metric to reach a maximum threshold, μ [11].

Finally, Elastic Loads are loads that are usually related to delay tolerant power tasks. Thus, the operation of the appliances that generate Elastic Loads can be postponed for another, more suitable, time of the day. All Elastic Loads are delay tolerant and non emergency tasks. Their operation is correlated to a comfort index. The comfort index can be regarded to be the time window of the execution of the task [11].

In the case of Elastic Loads, comfort is related to a threshold time deadline value, d. The modelling of the operation of Elastic Loads is characterized by an initiation point in time, a, a duration of the power task, s, the power of the task, p, and a deadline of execution, d [11].

Flexible Loads and Elastic Loads can be controlled with Demand Response mechanisms. On the other hand, the nature of Standard Loads puts them beyond the reach of any Demand Response mechanism. The best case scenario is to control devices that consume a lot of energy and at the same time do not affect users comfort. Typically those devices are thermostatic loads [11].

With Demand Response the load requirements can be shaped dynamically. Shaping the load can lead to load shifting so as to clip the peaks and fill the valleys and generally to strategic conservation of load [18].

3.3 Demand Response mechanisms, architecture and programs

A Demand Response mechanism is the methodology for controlling Flexible and Elastic Loads in a Smart House and a Smart Building.

Direct Load Control (*DLC*) is the Demand Response mechanism in which the aggregator has direct control over a device with Flexible Load. Thus, the aggregator can directly issue an ON or OFF command to the device. The advantage of Direct Load Control is that the system's response is immediate. On the other hand, the disadvantage is that direct control over a device might decrease user comfort. Direct Load Control is usually applied upon Flexible Loads and change the state of operation (ON/OFF) of a smart appliance in real time. The objective when applying Direct Load Control on a Smart House or Smart Building is to keep the total load below a threshold. The mathematical formula for the objective of Direct Load Control is:

$$P_P(t) = \sum_{j=1}^{N<M} [P_j(t)] \leq B_P(t), \forall t$$

where P_P is the total load, B_P is the maximum capacity or maximum cost, M is the number of appliances and P_j is the load of the appliance j [11, 19].

Load Scheduling is the Demand Response mechanism in which the aggregator can transfer non emergency tasks to off peak hours. This Demand Response mechanism is usually used upon Elastic Loads and non emergency tasks. Load Scheduling is a scheduling problem where there is the need for defining the point in time to initiate or disrupt the operation of a smart appliance with Elastic Load. The advantage of Load Scheduling is that it decreases user comfort less than Direct Load Control. On the other hand, the disadvantage is that this scheme provides limited flexibility for potential load reduction due to limited variety of appliances to control. The objective of Load Scheduling is to reduce the cost of electricity to minimum. The mathematical formula for the objective of Load Scheduling is [11]:

$$min \int_{0}^{T} \sum_{j=1}^{M} [P_j(t)]^n \, dt$$

Furthermore, there are two modes of operations for Load Scheduling. On the Pre-emptive Operation a task can be scheduled with interruptions as long as it is completed on time. On the Non Pre-emptive Operation once a task is started it remains active without the ability to interrupt it. Also, there are two types of algorithms for Load Scheduling, the Off Line Algorithm and the Online Algorithm. The Off Line Algorithm is defined for a specific time horizon and all power demand related parameters are deterministically known a priori. The Online Algorithm is defined in the long-run and control commands are continually generated by a stochastic model [11].

Critical elements for enabling demand side Load Scheduling are the dynamic pricing scenarios that are evolved through Demand Response needs. The most accepted of them are the Time of Use (*TOU*), the Critical Peak Pricing (*CPP*), the Peak Time Rebate (*PTR*) and the Real Time Pricing (*RTP*) dynamic pricing scenarios.

Time of Use dynamic pricing scenarios are Demand Response Load Scheduling scenarios that segment each billing month into smaller windows each with a different pricing level related to production cost. Also, participants are provided with price signals to reduce load during high cost hours [19, 20].

Critical Peak Pricing dynamic pricing scenarios are Demand Response Load Scheduling scenarios with a floating time frame which may or may not be in effect on any given day. Advanced notification signals for a Critical Peak Pricing dynamic pricing scenario is usually dispatched 24 hours before it takes place [19, 20].

Real Time Rebate dynamic pricing scenarios are Demand Response Load Scheduling scenarios that offer rebates to customers who use less electricity during critical peak events. Advanced notification signals for a Real Time Rebate dynamic pricing scenario is usually dispatched 24 hours before it takes place [19].

Real Time Pricing dynamic pricing scenarios are Demand Response Load Scheduling scenarios that use an hourly market based pricing without specific demand response events. The driving force behind the Real Time Pricing dynamic pricing scenarios is that price will determine usage and that the price elasticity within the market will drive customer behaviour to reduce load. Advanced notification signals for a Real Time Pricing dynamic pricing scenario is usually dispatched 24 hours (day ahead pricing) or 1 hour (hour ahead pricing) before it takes place [19]. Figure 3.2 shows the Time of Use and Critical Peak Pricing dynamic pricing scenarios.

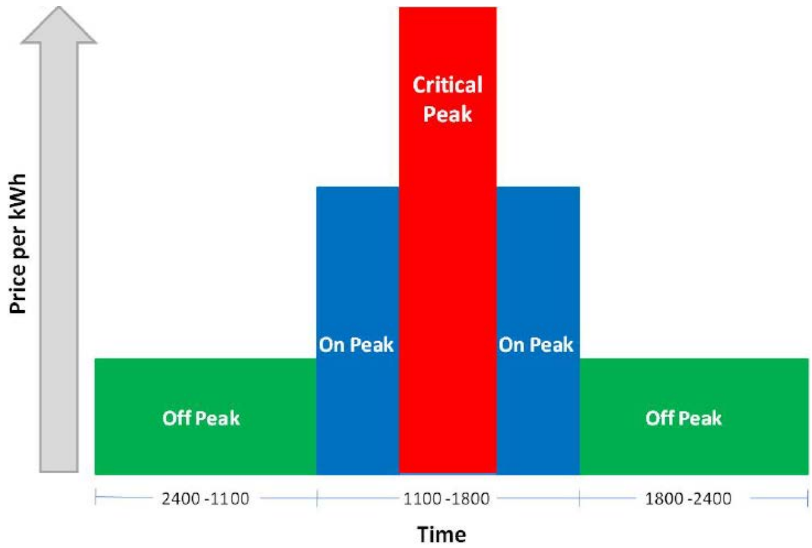

Figure 3.2: *Time of Use and the Critical Peak Pricing dynamic pricing scenarios [19]*

Finally, Incentives and Gamification can be regarded as a form of Demand Response mechanism. With this mechanism, Demand Response is achieved via appropriate incentives to the user and via providing a feeling that the user participates in a game. The advantage of Incentives and Gamification is that it does not decrease user comfort because users are participating on the control mechanism. On the other hand, the disadvantage is that this scheme cannot provide real time management due to the fact that it takes time for the user to obey to incentives. A successful incentive mechanism will keep the utility for the user positive in the long run and provide profits to the service provider [11].

The Demand Response architecture is the sequence of steps that needs to be fulfilled in order to provide a Demand Response command on a device in a Smart Home or Smart Building. The Demand Response architecture is a bidirectional flow of data and control signals. On the uplink, data with user-centric information like comfort parameters and appliance status flows from the smart device to the aggregator. On the downlink, real time pricing signals that are generated at the Utility are broadcasted to the aggregators. The Demand Response control command is decided upon specific appliances at the aggregator or at the agent [11]. Figure 3.3 shows the Demand Response interaction model.

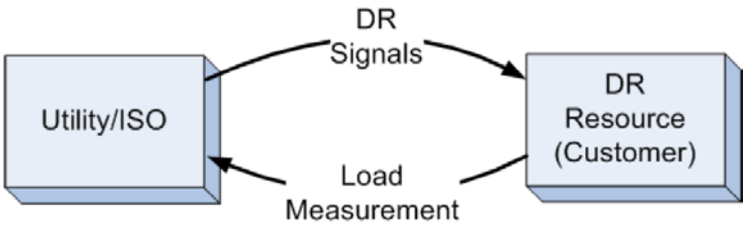

Figure 3.3: *Demand Response interaction model [17]*

Demand Response programs are programs that enable the customers to decide when and to what extent they will shape the demand [19]. Various programs have been evolved that meet different requirements, satisfy different needs and target different groups.

Individual Based Programs are programs that give to the customer the power to make the ultimate and final decisions on when, and to what amount, they will shape the usage [19].

Mass Market Programs are programs that are offered with the same incentive characteristics to a large set of relatively homogeneous customers [19].

Event Based Programs are programs that have the capability to respond to emergency reliability events [19].

Non Event Based Programs are programs that are economic based or provide load reductions that are not necessarily able to respond to emergency reliability events [19].

3.4 Demand Response benefits

A successful Demand Response implementation within the context of the Smart Grid will be beneficiary for all stakeholders, from the individual consumers to the major electricity providers. The benefits from implementing Demand Response are:

● Facilitate and increase consumer choices by providing a wide set of products, services and other incentives like the chance to be rewarded for actively participating [23].

● Manage consumer loads by facilitating adoption of Demand Response. Thus, new types of loads, like electric vehicles, that will increase the demand can be serviced without changing the peak energy production of the electricity provider [23].

● Optimize investments by optimising the use of existing network and promote optimised investment programs. Thus, investments are more secure and beneficiary and the cost to be passed through to the consumer is minimized [23].

● Avoid regional and national network congestion and provide enhanced fault tolerance. Networks of all scales can benefit from the improved efficiency through Demand Response in order to optimise operation and thus relieve congestion [23].

● Balance network operation in regional and national level through reserve or response services. Demand Response can effectively use renewable energy sources and stored energy in order to shape demand according to production and at the same time reduce carbon emissions [23].

● Change the trading position of energy suppliers by mitigating risk and allow better trading. Subsequently, this benefits the consumers because it allows suppliers to operate more efficiently and compete on their customer pricing [23].

● Promote renewable energy sources by helping energy demands to be satisfied by a mixture of traditional and renewable energy sources. Thus, countries will meet their renewable and carbon emissions targets and be more friendly to the environment [23].

● Provide future flexibility for unknown future generation or loads in the years to come. Providing Demand Response within the Smart Grid requires a large technological infrastructure. This will lead to fast future adoption of new implementations of power generation or consumption [23].

3.5 Limitations of Demand Response and security issues

Although Demand Response can be beneficiary for all stakeholders, from end-users to electricity providers, a series of barriers limit its effectiveness.

The limited number of retail customers on time based rates is the first impairment towards an effective Demand Response implementation. Annual reports highlighted the low number of retail customers who purchase electricity based on time rates. Without an expanded implementation the development of new technologies and programs and the fulfilment of Demand Response potential may be slowed [15].

The lack of consistency in the measurement and verification of demand reductions and the lack of demand responsive specific and cost effectiveness tools remains a barrier. Thus, there is the need for Demand Response measurement and verification standards and high consistency [15].

Although significant progress has been made, for an effective Demand Response implementation there is the need for commonly accepted uniform standards and increased interoperability. Until now, communications to and interactions with Demand Response resources are technology specific proprietary protocols and techniques. This results to duplicating systems and inefficient transfer of pricing and usage of information between parties [15].

The lack of customer engagement is another drawback against a successful Demand Response implementation. The general public need to be educated and informed about Demand Response and Smart Grid in order to be active and to participate in a Incentives or Gamification based Demand Response deployment. Thus, the communities need to be reached by communication, advertisements and explanatory actions in order to understand and adopt the Demand Response deployment. Otherwise, communities may not respond or respond negatively to actions by utility providers [15].

Also, the lack of Demand Response forecasting and estimation tools is the last barrier towards a successful Demand Response implementation. Currently, planning and forecasting tools are not sufficiently robust to model adequately the capability of Demand Response and to adjust consumption in near real time. Thus, new tools and methods should be developed to directly incorporate Demand Response into dispatch algorithms and resource planning models and to forecast and model the capability of Demand Response to adjust consumption in near real time [15].

Finally, since an implementation of Demand Response within the context of Smart Grid requires extended and advanced communication capabilities, various security issues rise that will change the landscape of the industry. The challenge for security rises from the fact that there is a large number of diverse stakeholders across the electricity sector with different security expectations, objectives and understanding of security. All these different stakeholders have no common approach to security in the electricity sector [23].

Thus, there is a need to establish a consistent, repeatable and adaptable process for risk management across the entire electricity sector. This process needs to be adaptable in order to meet individual organizational requirements. Also, it needs to recognize organizational constraints, to allow resource allocation based on risk management principles and to identify ownership of risk within the organization [24].

Security and privacy is a major issue and concern as countries and industries are heading towards creating Smart Cities and every appliance will be part of the Internet of Things (IOT). A successful security penetration may lead to stolen security tokens, compromised metering systems, loss of control, data theft, and production station jamming [25].

Thus, there is the need to reduce asset risk to an utility acceptable level and measure the system's ability to resist security violations while still providing service to authorized users [25].

4 Demand Response standards

A standard is a set of rules that are agreed and must be followed. A Demand Response standard is a set of rules created by a credible academic research community and is adopted by the energy industry. The goal of a Demand Response standard is to standardize all Demand Response interactions and maximize the effectiveness of Demand Response. The two most significant and effective Demand Response standards are the OpenADR and SEP standards [40].

In this chapter we will attempt to provide an exhausting analysis of the OpenADR standard that is the standard adopted by this master thesis and a quick overview of the SEP standard for conceptual completeness. Finally, we will try to identify the differences between these two standards.

4.1 The OpenADR standard

The OpenADR standard is one of the most wildly used and accepted by the academic community as well as the energy industry Demand Response standard. The OpenADR standard aims to standardise the communications and interface between energy providers and consumers, to automate the demand side to dynamic prices and grid needs and to simplify the control of the energy grid while at the same time to maximize the Demand Response effectiveness [26].

In this subchapter we will attempt to provide a deep and very thorough understanding of the OpenADR standard. All aspects of the OpenADR standard, from its' history and evolution to the provided security and certification, will be examined. Furthermore, residential and commercial implementations will be studied and analysed. Finally, the OpenADR 2.0 A and B profile will be thoroughly examined.

4.1.1 The OpenADR standard, history, definition and deployment status

The Open Automated Demand Response (*OpenADR*) standard was initiated by Lawrence Berkeley National Laboratory (*LBNL*) and the California Energy Commission (*CEC*) in the year 2002. After a five year long development phase with pilots and field trials, in the year 2007, the first commercial version, OpenADR 1.0, was launched. The first official specifications for OpenADR 1.0 were published in the year 2009. A second phase of development with pilots and field trials was initiated afterwards that lasted four years. The second development phase included all end users and sectors like wholesale markets, ancillary services, dynamic pricing, renewable energy sources and electric vehicles. Thus, in the year 2013, the second version of OpenADR, OpenADR 2.0, was launched. Also, in the year 2013, the specifications of OpenADR 2.0 were published as well as OpenADR 2.0 products [26]. Figure 4.1 shows the OpenADR logo.

Figure 4.1: *The OpenADR logo [26]*

Many formal definitions of the OpenADR standard exits such as the two that follows:

"*The OpenADR standard provides a non-proprietary, open, standardized Demand Response interface that allows electricity providers to communicate Demand Response signals directly to existing customers using a common language and existing language and existing communications such as the Internet.*" [26]

"*A communications data model designed to facilitate sending and receiving DR signals from a utility or independent system operator to electric customers. The intention of the data model is to interact with building and industrial control systems* *that are pre-programmed to take action based on a DR signal, enabling a demand response event to be fully automated, with no manual intervention. The OpenADR specification is a highly flexible infrastructure design to facilitate common information exchange between a utility or Independent System Operator (ISO) and their end-use participants. The concept of an open specification is intended to allow anyone to implement the signaling systems, providing the automation server or the automation clients.*" [29]

Figure 4.2 shows the OpenADR schema.

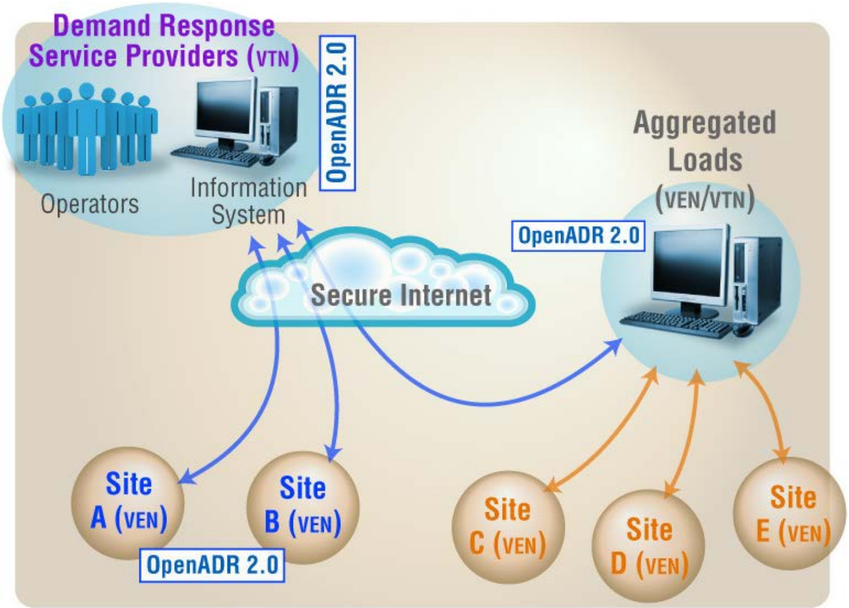

Figure 4.2: *The OpenADR schema [26]*

The OpenADR standard enjoys worldwide acceptance with deployments around the world while at the same time provides significant results for average peak load reduction [26]. Figure 4.3 shows the deployments of the OpenADR. Figure 4.4 shows the average peak load reduction of the OpenADR implementations.

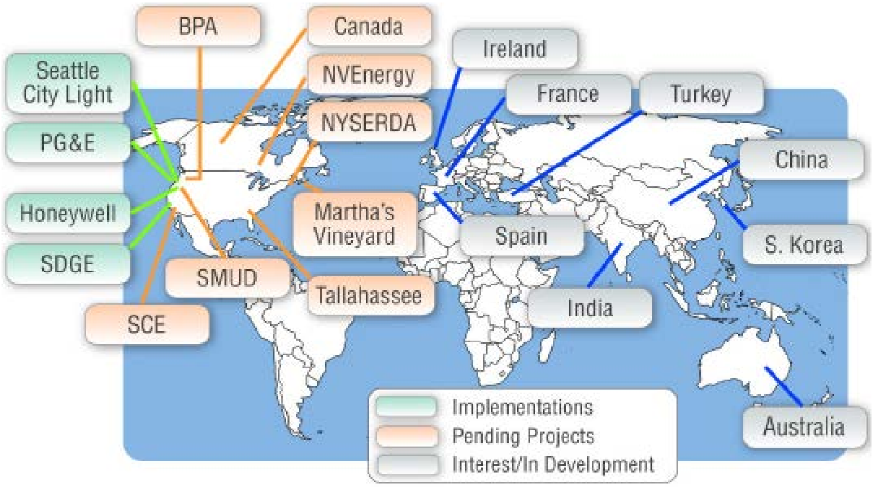

Figure 4.3: *The OpenADR deployments [26]*

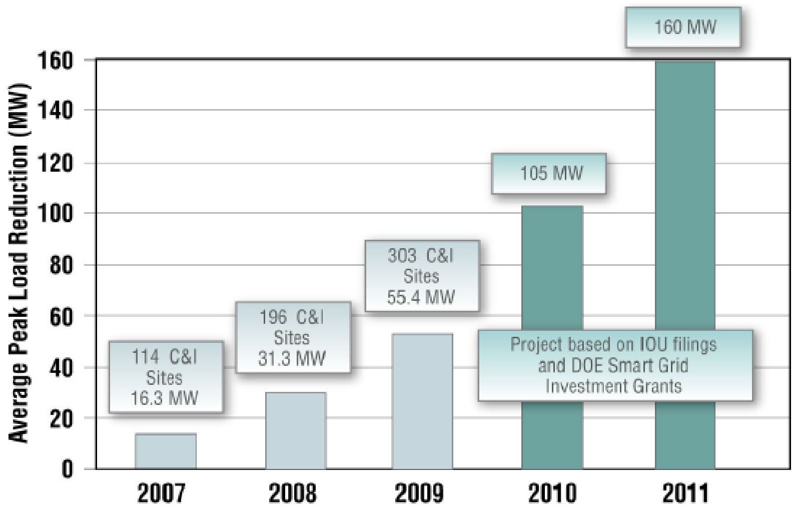

Figure 4.4: *The OpenADR average peak load reduction [26]*

At the same time, during the year 2009, the National Institute of Standards and Technology (*NIST*) Smart Grid initiative started a harmonization project with priorities to work on common standards for price models, schedule representations and Demand Response signals. The OpenADR 2.0 standards uses the standardized output from the National Institute of Standards and Technology Smart Grid initiative harmonization project while at the same time adds feedback and other price related features [26]. Figure 4.5 shows the flow diagram of the OpenADR.

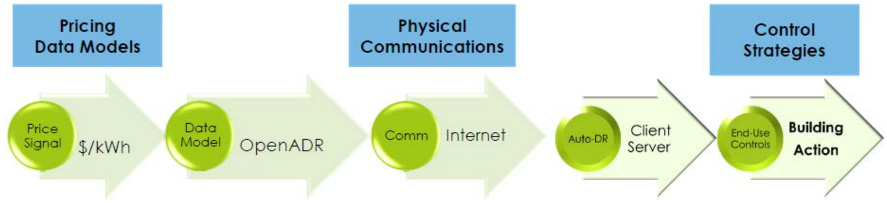

Figure 4.5: *The OpenADR flow diagram [26]*

4.1.2 The OpenADR standard evolution, actors and services

The transition from OpenADR 1.0 to 2.0 brings a large number of improvements. OpenADR 1.0 could support a limited number of vendors while OpenADR 2.0 can support a large ecosystem of vendors. OpenADR 1.0 had no certification program while OpenADR 2.0 provides a testing tool, a testing plan and certification. OpenADR 1.0 was oriented towards local Demand Response programs while OpenADR 2.0 is flexible enough and can adjust to most Demand Response programs. OpenADR 1.0 wasn't nor a national neither an international standard while OpenADR 2.0 is an international standard based on the Organization for the Advancement of Structured Information Standards (OASIS) standard. Finally, OpenADR 1.0 was limited to basic Demand Response applications while OpenADR 2.0 has an expanded architecture and includes pricing, telemetry and other services. Also, the OpenADR 2.0 standard includes architectural models for data models for information exchange, information exchange patterns and distributed energy recourses [26].

The OpenADR 2.0 offers a continuous, secure and reliable two way communication infrastructure with acknowledgment support by the end points towards the service provider. Also, the OpenADR 2.0 uses the existing internet communications infrastructure to transmit Demand Response signals while at the same time be compatible with existing control systems by using open and well established standards based on Internet Protocol (*IP*) and web technologies. Finally, the OpenADR 2.0 provides automation to Demand Response events through predesigned programs and strategies within a scalable architecture able to support different forms of Demand Response programs [27, 17].

Within the OpenADR 2.0 standard, all participants can be a Virtual Top Node (*VTN*) or a Virtual End Node (*VEN*) or a combination of them. For example, a server is now a Virtual Top Node and a client is now a Virtual End Node [26, 27].

The OpenADR 2.0 standard uses a web service like a logical request - response service where each service has a single common endpoint and an Extensible Mark-up Language (*XML*) payload where the root element defines the service and the operation [26].

The services supported by OpenADR 2.0 are [26, 27, 28, 32]:

● EiEvent service: Sends and acknowledges Demand Response events

● EiOpt service: Defines temporary availability schedules

● EiReport service: Requests and delivers reports

● EiRegisterParty service: Virtual End Node registration and device information exchange

● EiEnroll service: Enrolls a resource for participation in a Demand Response program

● EiMarketContext service: Discovers program rules and standard reports

● EiQuote service: Distributing of complex dynamic prices

● EiAvail service: Defines constraints on the availability of resources

Figure 4.6 shows a services usage scenario.

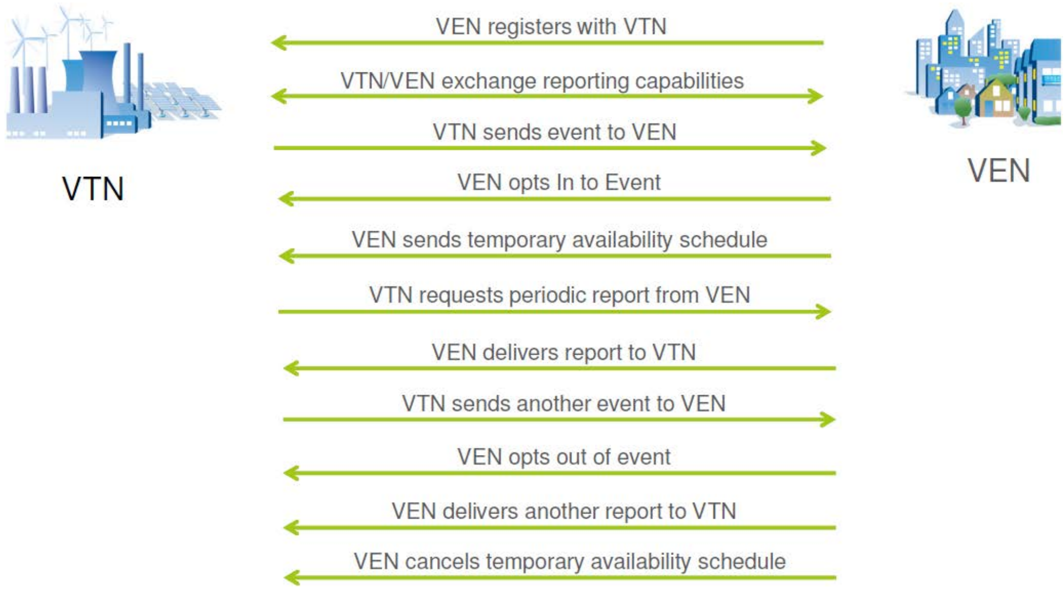

Figure 4.6: *A services usage scenario [26]*

Some examples of services payloads are [26]:

● EiEvent service payload:

1. oadrRequestEvent: Virtual End Node requests Demand Response events

2. oadrDistributeEvent: Virtual Top Node sends Demand Response events

3. oadrCreatedEvent: Virtual End Node opts in/out of events

4. oadrResponse: Virtual Top Node acknowledges Virtual End Node opt in/out

● EiOpt service payload:

1. oadrCreateOpt: Virtual End Node sends opt schedule

2. oadrCreatedOpt: Virtual Top Node acknowledges receipt of schedule

3. oadrCancelOpt: Virtual End Node cancels opt schedule

4. oadrCanceledOpt: Virtual Top Node acknowledges cancellation

● EiReport service payload:

1. oadrRegisterReport: Declares available reports

2. oadrRegisteredReport: Acknowledges receipt of available reports

3. oadrCreateReport: Requests specific report

4. oadrCreatedReport: Acknowledges receipt of request

5. oadrUpdateReport: Delivers requested report

6. oadrUpdatedReport: Acknowledges receipt of report

7. oadrCancelReport: Cancels requested report

8. oadrCanceledReport: Acknowledges cancellation request

● EiRegisterParty service payload:

1. oadrCreatePartyRegistration: Virtual End Node registration request

2. oadrCreatedPartyRegistration: Virtual Top Node registration acknowledgement

3. oadrCancelPartyRegistration: Requests cancel registration

4. oadrCanceledPartyRegistration: Acknowledges Cancelation

5. oadrRequestReregistration: Requests re-registration

The information contained within an OpenADR event is [27]:

● EventID: An unique ID for the event

● ModificationNumber: A sequence that starts at zero and is incremented by 1 each time the Virtual Top Node modifies the event

● Priority: An indication of the event priority

● MarketContext: Identifies a particular program or application

● CreatedDateTime: The time when the payload containing the event was created

● EventStatus: The status of the event, indicating if the event is near, far, active, or cancelled

● TestEvent: If not false, indicates this is a test event

● VtnComment: Arbitrary comment provided by the Virtual Top Node

4.1.3 The OpenADR standard profiles and events

The OpenADR 2.0 standard has two profiles, profile A and profile B, with different orientation and applications. OpenADR 2.0 A profile is targeted at devices with limited resources and simple Demand Response applications. OpenADR 2.0 B profile is targeted at robust devices and sophisticated Demand Response applications [26].

An OpenADR event contains information within the event object separated into five groups. The Event Descriptor group holds general metadata about the event. The Active Period group holds information about the event start time and overall duration. The Event Signal group holds information about interval data for the event. The Event Baseline group holds information about interval data for the baseline. Finally, the Target group holds information about resources targeted by the event [26].

An OpenADR event can by dissected into three states in respect to time. The first state is the Pending State, where the notification about the event and the transition into the event takes place. The Pending State can be separated into the Far State and the Near State. The Far State is the time period until the Ramp Time takes place. The Ramp Time is the transition time that the preparation of the device occurs until the event takes place. The Ramp Time takes place during the Near State. The second state is the Active State, where the event takes place. The Active State lasts for the entire event duration. Finally, the Completed State is the state after the event. Part of the Completed State is the Recovery Time where the device recovers from the event and no other events can take place [26, 27].

During the Active State of an OpenADR event various signals that carry the actionable information are transmitted to the device. Those signals carry information about the duration of the interval and the electricity price on that interval respectively [26, 27]. Figure 4.7 shows the characteristics of an OpenADR event.

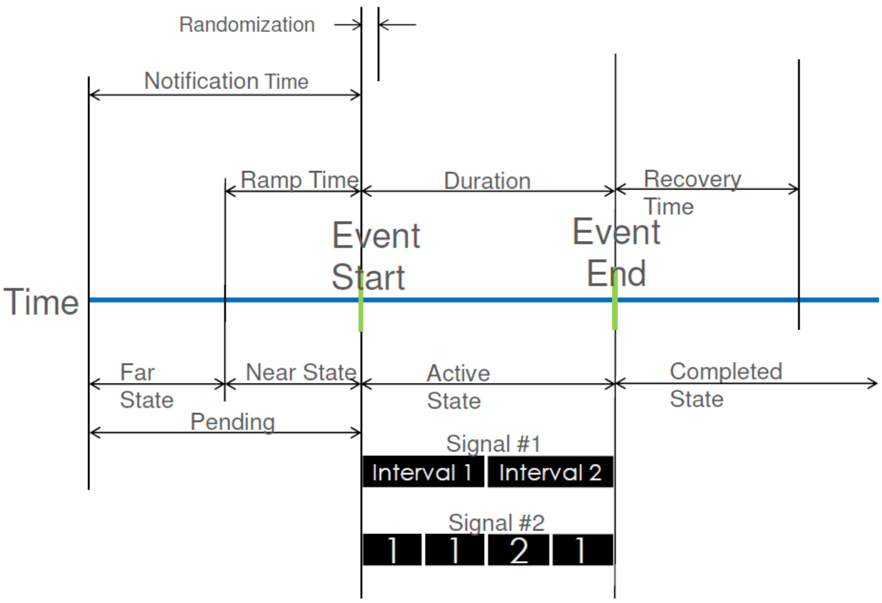

Figure 4.7: *OpenADR event characteristics [26]*

An OpenADR event can be targeted into different targets. For example, an OpenADR event can target a Virtual End Node, a group of devices, a class of devices, a service area or a specific resource. The marketContext specified in the event holds the information about the event targeting [26].

4.1.4 The OpenADR standard security and certification

As far as security is concerned, OpenADR offers two security levels, standard and high. Upon every implementation the appropriate security level must be selected [26].

Every manufacturer that needs to certify an OpenADR ready product needs to undertake a certification process that leads, if successful, to the product been certified. Four suites based on the Eclipse open source Integrated Development Environment (*IDE*) are available and provide coverage for the two device types, Virtual Top Node and Virtual End Node, and the two exchange patterns, push and pull. Each test suite consists of approximately 260 test cases for OpenADR A and B profile that cover positive, negative and functional test scenarios. Also, test coverage includes schema and conformance rules validation and validation of each test's scenarios intent. Finally, test reporting includes exchange logs, XML payloads, conformance checks completed and detailed failure information [26].

4.1.5 The OpenADR standard communication architecture and implementation configuration

The OpenADR communication architecture can be separated into three implementation classes. For low end devices, simple Demand Response events and price information the suitable class is the OpenADR 2.0 A profile. For complex high end implementations with complex events and price process as well as feedback and additional services the suitable class is the OpenADR 2.0 B profile. Finally, there is the Aggregator class that includes Independent System Operator (*ISO*) to aggregator information exchange also called OpenADR C profile [26].

The OpenADR standard uses Hypertext Transfer Protocol (*HTTP*) as a transport protocol and standard Hypertext Transfer Protocol commands. Hypertext Transfer Protocol is ideal for pull clients and possible for push if security issues are handled. Also, the OpenADR standard uses Extensible Messaging and Presence Protocol (*XMPP*) as a transport protocol. This protocol is ideal for push applications and fast Demand Response while pull is also possible. Virtual End Nodes can utilize Hypertext Transfer Protocol or Extensible Messaging and Presence Protocol but both of the protocols are mandatory for a Virtual Top Nodes [26]. Thus, the OpenADR standard event communication process is a push pull action between the Virtual Top Node and the Virtual End Node. The Virtual Top Node push an event to an Hypertext Transfer

Protocol Universal Recourse Identifier (*URI*) exposed by the Virtual End Node. On the other hand, a Virtual End Node periodically pull upgrades from the Virtual Top Node [27].

The OpenADR standard utilise the Transport Layer Security (*TLS*) cryptographic protocol with additional server and client side certificates in the Standard Security implementation. In a High Security implementation the OpenADR standard utilise the Transport Layer Security cryptographic protocol with additional server and client side certificates and Extensible Mark-up Language signatures to increase non repudiation. The Standard Security setup is mandatory while the High Security is optional [26].

The OpenADR standard can be implemented in various implementation configurations. In a Direct Connect implementation configuration the Virtual Top Node is directly connected to the Virtual End Node via the internet. For example, an OpenADR 2.0 A enabled thermostat that is retail or operator provided and needs to be configured on device level is a Virtual End Node and is controlled via the internet by the operator that is the Virtual Top Node. The drawback of this implementation is the recourse constraint and the lack of feedback [26]. Figure 4.8 shows the Direct Connect implementation configuration.

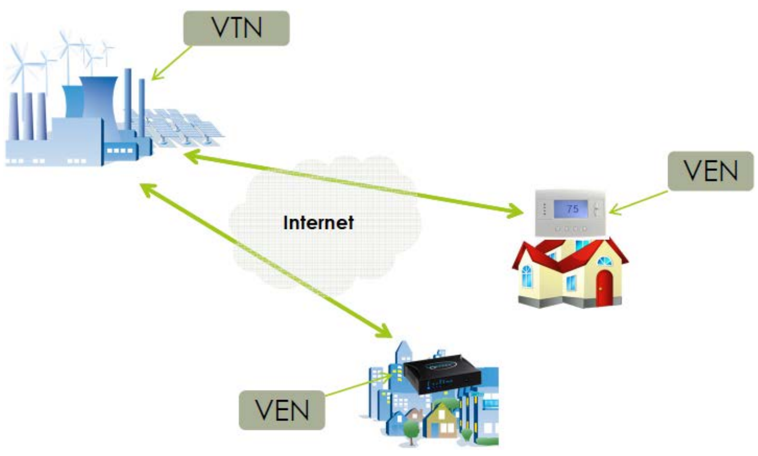

Figure 4.8: *Direct Connect implementation configuration [26]*

The Direct Connect with Cloud Interface implementation configuration enhances the Direct Connect implementation configuration with a Graphical User Interface (*GUI*) in the cloud. This alteration brings significant improvements. Within a Direct Connect with Cloud Interface implementation the devices that are controlled can be OpenADR 2.0 A or B enabled and can be configured in the cloud interface. Also there are no resource constraints and feedback is possible. The Direct Connect with Cloud Interface implementation can additionally be improved with an Energy Management System (*EMS*) in cloud or in a Local Area Network (*LAN*) [26]. Figure 4.9 shows the Direct Connect with Cloud Interface implementation configuration.

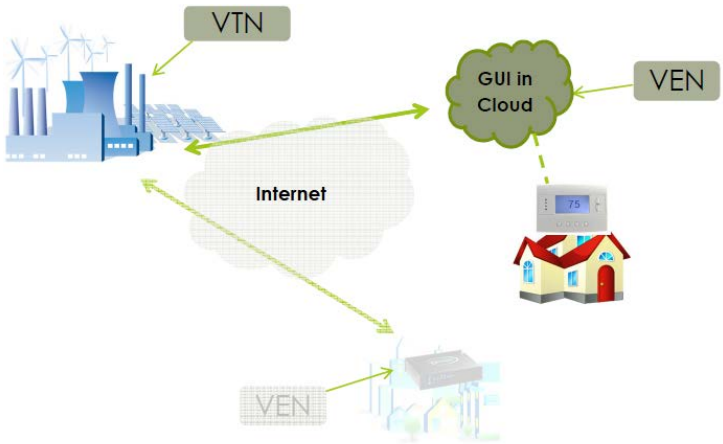

Figure 4.9: *Direct Connect with Cloud Interface implementation configuration [26]*

The Commercial and Industrial implementation configuration uses a Virtual End Node to control commercial and industrial buildings. The Virtual End Node communicates with the building via a Wide Area Network (*WAN*) [26]. Figure 4.10 shows the Commercial and Industrial implementation configuration.

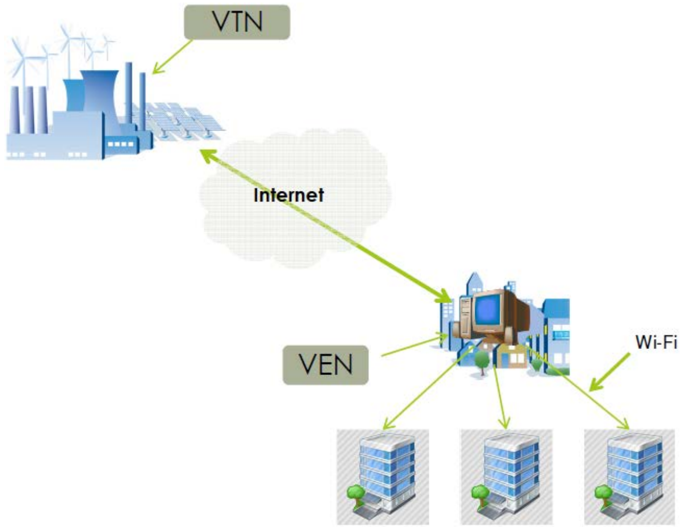

Figure 4.10: *The Commercial and Industrial implementation configuration [26]*

Finally, the Aggregator implementation configuration utilizes an aggregator with double role. The aggregator is a Virtual End Node in respect to the operator that is a Virtual Top Node and at the same time is a Virtual Top Node in respect to the Smart House or Smart Building that is a Virtual End Node [26]. Figure 4.11 shows the Aggregator implementation configuration.

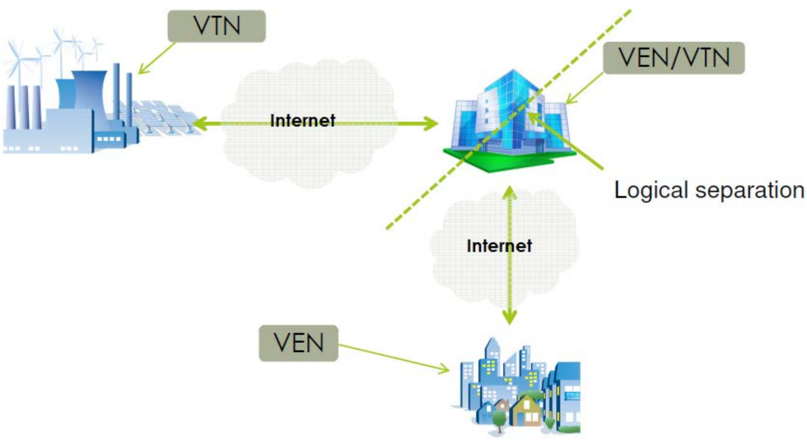

Figure 4.11: *The Aggregator implementation configuration [26]*

4.1.6 The OpenADR standard Demand Response programs

The OpenADR standard supports a wide variety of Incentive Based and Time Based Demand Response programs. A description of Demand Response programs follows.

- Incentive Based programs:
  - Direct Load Control:

Direct Load Control Demand Response program is covered by OpenADR 2.0 A profile. There is no support for feedback and the availability of OpenADR 2.0 A load control devices is limited. Direct Load Control is implemented with a Direct Connect implementation configuration directly to the device or to a Home Energy Management (*HEM*) system. Also, devices are likely to be pre configured and only a basic Demand Response event can be executed [26].

  - Interruptible Load:

Interruptible Load Demand Response program is covered by OpenADR 2.0 A profile and there is no feedback support. Interruptible Load is implemented with a Direct Connect implementation configuration directly to the device or to a Home Energy Management system. Interruptible Load Demand Response program can support deployment architectures. Instead of OpenADR 2.0 A profile, OpenADR 2.0 B profile can be used. OpenADR 2.0 B profile provides real time feedback and additional granularity to Demand Response events management like dynamic control mechanisms [26].

  - Critical Peak Pricing:

Critical Peak Pricing Demand Response program is covered by OpenADR 2.0 A profile and there is no feedback support. Critical Peak Pricing is implemented with a Direct Connect implementation configuration directly to the device or to a Home Energy Management system. Critical Peak Pricing Demand Response program can support deployment architectures. Instead of OpenADR 2.0 A profile, OpenADR 2.0 B profile can be used. OpenADR 2.0 B profile provides real time feedback and additional granularity to Demand Response events management like dynamic control mechanisms and pricing information [26].

● Load as a Capacity Resource:

Load as a Capacity Resource Demand Response program is covered by OpenADR 2.0 B profile with real time feedback and enhanced Demand Response event management [26].

● Spinning/Responsive Reserves:

Spinning/Responsive Reserves Demand Response program is a fast Demand Response program designed to respond in seconds with push scenarios. OpenADR 2.0 A profile in push mode with no feedback or OpenADR 2.0 B profile in push mode with feedback can be utilized [26].

● Non Spinning Reserves:

Non Spinning Reserves Demand Response program is covered by OpenADR 2.0 A profile and there is no feedback support. Non Spinning Reserves is implemented with a Direct Connect implementation configuration directly to the device or to a Home Energy Management system. Non Spinning Reserves Demand Response program can support deployment architectures. Instead of OpenADR 2.0 A profile, OpenADR 2.0 B profile can be used. OpenADR 2.0 B profile provides real time feedback and additional granularity to Demand Response events management like dynamic control mechanisms [26].

● Time Based programs:

OpenADR standard supports Dynamic Pricing Demand Response programs, like Real Time Pricing and Peak Pricing, as well as Time of Use Pricing programs. In a Dynamic Pricing program the customer knows the electricity price a day in advance while in Time of Use Pricing program the customer knows the electricity price more than a day in advance although the price can change over time [26].

The pricing elements are Market Based Real Time Prices that are static and entered manually by the operator or by real time feed by the energy market via the internet. Also, pricing elements can by Retail Peak Pricing with Time of Use Rates. There are two scenarios for Dynamic Pricing Mapping Strategies. The first is the absolute mapping of price ranges according to operation modes and the second is the relative mapping of prices according to operation modes. In the relative mapping of prices scenario there are customized dynamic price response strategies to varying prices [26].

4.1.7 The OpenADR standard implementation, residential case

Although the energy consumption from residential customers is about 20% of a nations' total energy consumption, residential Demand Response implementations are very limited. Nevertheless, utilities and regulators are motivated from the success of various commercial implementations to expand Demand Response programs to residential customers [30].

A residential environment presents various technical issues towards an implementation of automated Demand Response. Thus, a Home Automation Network (*HAN*) is examined in order to unveil its potential in implementing wide scale Demand Response systems that communicate directly to individual residences [30].

The role of OpenADR standard is to define and codify the messages exchanged between an energy supplier and the participating consumer during a Demand Response event. Thus, building on well established information technology (*IT*) and communications standard, the process of transforming a proposed standard to an operational one is fast [30].

The OpenADR standard offers [30]:

● A continuous, secure and reliable two way information exchange infrastructure for Virtual End Nodes, Virtual Top Nodes and Demand Response Automation Servers (*DRAS*).

● Translation between Demand Response price and event information to internet signals in order to facilitate automation.

● Automation upon receiving of external Demand Response signals through the use of Demand Response programs.

● Opt-out or override to consumers if a Demand Response event is not desired by the end user.

● A complete data model describing the architecture to communicate price, reliability and all other Demand Response activation signals.

● Scalability in order to engulf different forms of Demand Response programs.

● Standardized message formats by using Extensible Mark-up Language tags that allows future semantic and content extensions to messages.

● Open standards based technology such as Simple Object Access Protocol (*SOAP*) and web services from the basis of the OpenADR standard.

Currently, a home network is composed by a wide variety of inexpensive communications and networking technologies in order to support each product's particular needs. Both low and high bandwidth applications exist such as on/off and volume control and audio and video delivery. Unfortunately most of the network technologies used have not evolved from a common set of root standards. Thus, most of the network protocols used in a home network operate outside of any standards sanctioning body [30].

On the other hand, there is another category of home networks, which were not designed specifically for residential use but are found in many homes. Those are the networks that are based upon Internet Protocol and are the heart of every Personal Computer (*PC*), laptop, printer and router found in every home. In every home network, Internet Protocol forms the backbone of the network and is able to serve all the needs found in home automation applications. Unfortunately, for cost and complexity reasons, Internet Protocol based networks remain beyond the reach of most home control applications [30].

While wired Internet Protocol standards are too costly for low-end home networking applications, a new generation of wireless low-power networking devices is available. These devices are fashioned within a standard process and comply to publicly available specifications like the IEEE 802.15.4. These devices will find acceptance in residential control applications. Thus, a new breed of modern, inexpensive and interoperable low-end devices could form the basis of a home automation network [30].

Various standardized technologies that can pass signals into a residence are Radio Broadcast Data System (*RBDS*), Power Line Communications (*PLC*) and low power wireless protocols [30]. Figure 4.12 shows a Home Automation Network.

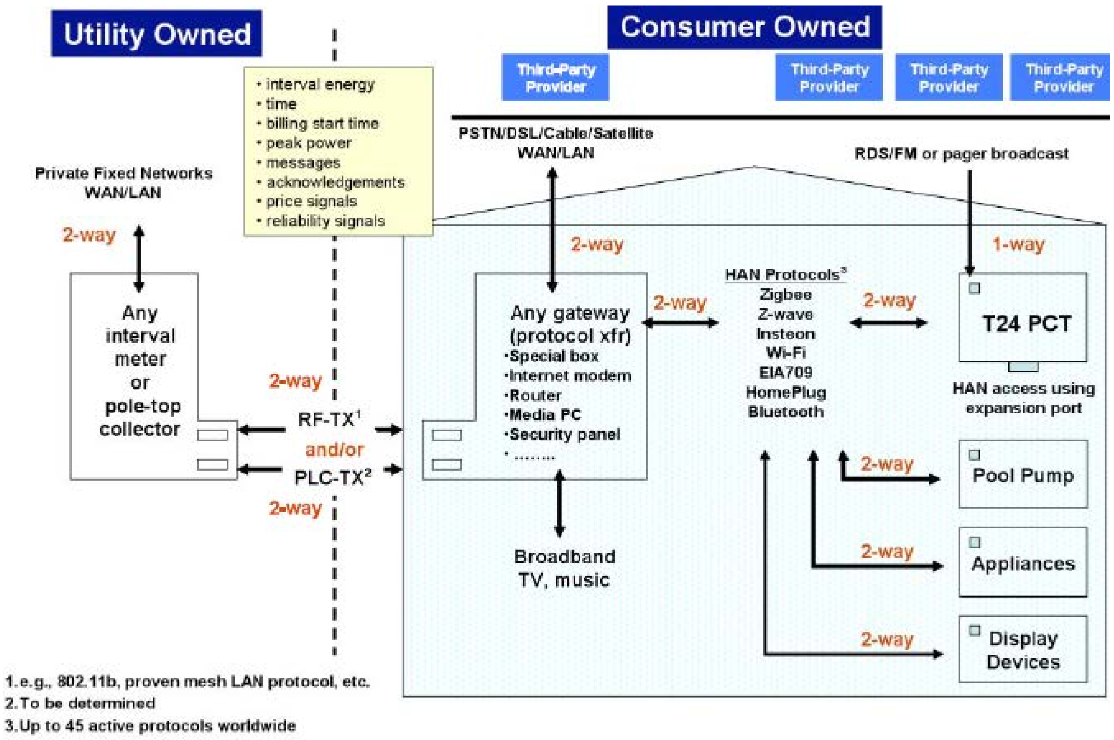

Figure 4.12: *A Home Automation Network [30]*

Since a plethora of protocols, devices and other components exist that can form and be integrated into a Home Automation Network capable of receiving, executing and reporting Demand Response events, the evaluation process is of paramount importance. Various aspects of every implementation needs to be evaluated in respect to many performance metrics. The most important metrics are [30]:

● Bandwidth: Bandwidth quantifies how much data can be transferred over a communications channel in a given time interval. Most residential load control applications require relatively little communications bandwidth.

● One-way, two-way: Attempting to minimize cost and complexity, some devices have implemented a way of data transmission in one direction only, from the controller to the device. This way of transmitting data is capable of accepting a Demand Response event but cannot give any kind of feedback.

● Security: Giving security to the wide variety of functions within home networks is a complex issue. Three major security related tasks are identified. First, every node must verify that the received data packet is the same as the data packet that was transmitted. Second, every node must verify that the exact number of packets that are transmitted are received. Third, every node must verify that the received message originate from a trusted source. Many of these characteristics are missing from simple home network protocols. On the other hand, high level Internet Protocol based protocols are capable of implementing all these security tasks. Security risks are more critical considering the wireless nature of many communications within a Home Automation Network.

● Openness: Openness is a metric related to the public or private nature of the network protocol specification. Public protocols are more likely to be adopted for use in devices.

● Protocol Specification Status and Completeness: Protocol Specification Status and Completeness is a metric that reveals if the protocol is mature enough to support commercial products.

● Developer Protocol Interface: Developer Protocol Interface is a metric that characterise how a system designer gains access to the protocol. The availability of an interface specialised for creating protocol messages is a state that enables easy programming of the protocol. On the other hand, if application level commands are passed to a chip that formats and sends the appropriate message results into a state that discourage protocol programming.

● Deployment and Diagnostic Tools: Deployment and Diagnostic Tools is a metric that characterizes the availability of specialized tools that enables a diagnosis to be made for the deployment prior to the actual deployment or even fine-tune the actual deployment by using feedback from the diagnostic tool. Especially since most Home Automation Network will be wireless, they must be robust, simple and efficient.

● Application Programmers Interface (*API*): Application Programmers Interface is a metric that characterises the existence of software interfaces at several key locations in the logical program flow. These software interfaces define the capabilities and behaviour of the system. The lack of an Application Programmers Interface is considered to be a bad design.

Table 4.1, 4.2, 4.3 and 4.4 shows the characteristics of high level protocols.

Protocol	Bandwidth	One/Two Way Communication	Security -Data Integrity -Protection from Data Loss -Confidentiality	Openness
ZigBee	250Kbps max.	Two way	-predicted to be very robust -security and data loss status unknown - application templates	Private, available to license holders
6lowpan	250Kbps max.	Two way	-Full IP security capablities	IETF standard
Bluetooth	1-3 mbps	Two way	-Full IP security capabilities	IEEE standard
Wi-Fi	2 to 248 mbps	Two way	-Full IP security capabilities	IEEE standard
Z-wave	9.6 or 40Kbps	Two way	-Minimal security addins in progress	Private, available to license holders

Table 4.1: *High level protocol characteristics (1) [30]*

Protocol	Protocol Specification Status	Protocol Status/ Completeness	Developer Protocol Interface	Deployment / Diagnostic Tools
ZigBee	Written spec. available to licensees	Early chip level implementations shipping	802.15.4 plus assembled modules, multiple vendors	Yes
6lowpan	Written spec.	Complete with routing protocols in progress	802.15.4 plus assembled preliminary modules, multiple vendors	Yes
Bluetooth	Written spec.	Complete	MAC/PHY chips and modules from multiple vendors.	Yes
Wi-Fi	Written spec.	Complete	MAC/PHY chips and modules from multiple vendors.	Yes
Z-Wave	Written spec. available to licensees	Complete, security additions in progress	Microcontroller interface available under license	Yes

Table 4.2: *High level protocol characteristics (2) [30]*

Protocol	Bandwidth	One/Two Way Communication	Security -Data Integrity -Protection from Data Loss -Confidentiality	Openness
HomePlug 1.0 (Intellon)	14 mbps/ 85 mbps (unofficial)	Two way	-considered very robust -security and data loss status unknown	Private, available to license holders
HomePlug CC	7.5kbps	Two way	-predicted to be very robust -security and data loss status unknown	Private, available to license holders
HomePlug AV (IEEE P1901)	>100mbps	Two way	Security characteristics expected similar to current IP protocols	Proposed open IEEE standard
Insteon	2.99kbps	One way	-marginal data integrity -no security or data loss protection	Private, at least one additional license holder
UPB	240bps	One way with ACK	-marginal data integrity -no security or data loss protection	Private, licensed to multiple manufacturers
X-10	120 bps	One way and One way with ACK	-marginal data integrity -no security or data loss protection	Private

Table 4.3: *High level protocol characteristics (3) [30]*

Protocol	Protocol Specification Status	Protocol Status/ Completeness	Developer Protocol Interface	Deployment / Diagnostic Tools
HomePlug 1.0 (Intellon)	Written spec. available to licensees	Implementations shipping since 2002	MAC/PHY chip from Intellon	Yes
HomePlug CC	Written spec. available to licensees	Early chip level implementations shipping	MAC/PHY chip from Yitran	No
HomePlug AV (IEEE P1901)	Written submission to IEEE	Still in proposal stage of development	None at present, expect MAC/PHY chip level access	No
Insteon	Written spec.	Implementations shipping since 2004	TBD	Private, at least one additional license holder
UPB	Written spec. available to licensees	Implementations shipping since 2002, multiple manufacturers	Command interface to microcontroller	Yes, embedded diagnostic tools
X-10	Written spec.	One way and One way with ACK	-marginal data integrity -no security or data loss protection	Yes

Table 4.4: *High level protocol characteristics (4) [30]*

The vision for home use is a suitable control and data network that controls the residence in order to optimize convenience, comfort and security while at the same time minimizes energy use and provides meaningful response to utility load requests. Given the large variety of systems and protocols that must be integrated and controlled by a single entity, the major problem for a successful Demand Response implementation for residential use with OpenADR is the problem of integration [30].

Two approaches have been proposed for attacking the problem of integration. The first approach involves a gateway that is capable of translating messages from each existing system into the OpenADR language and format. This approach is labour intensive and often fragile because it requires a central system to understand the commands and the communications protocols for all systems connected to the gateway and to control them in an effective way. Thus, this solution may be non realistic, mainly due to the software effort required. The second approach also involves a gateway that controls the system but this time it only integrates the infrastructure that is already compatible. Thus, the gateway do not have the burden of integrating the complete environment of devices, just those that are already designed to accept and make good use of Demand Response events by the gateway [30].

To conclude, the current missing element of the residential Demand Response puzzle is a standardized residential computing and communications environment in which a Home Automation Network interacts with a Demand Response standard like the OpenADR. Furthermore, there are other pressing research questions about the OpenADR implementation into a residential scenario. Is there a need for the rich semantics of OpenADR to be mapped into simpler set of signals more suitable for residential use? Reducing the semantics of OpenADR for residential use will require less computational power and dedicated hardware but part of its capabilities will be lost. What protocol provides a minimum level of reliability for time-shifting operations? A deep understanding of control requirements will drive the acceptance of various home networking technologies [30].

When the barriers that exist now are removed, field testing of potential implementations of Demand Responds strategies via the OpenADR standard will commence. Also, it is very likely that the residential case will be very similar to small commercial implementations due to the fact the small commercial buildings very often are operated like residential buildings [31].

4.1.8 The OpenADR standard implementation, commercial case

The commercial implementation of the OpenADR standard is far more studied than the residential implementation. With respect to the applicability of the OpenADR standard, commercial cases can be separated into small commercial buildings and large commercial buildings.

The difference between small and large buildings is that small buildings are generally not equipped with centralized Energy Management and Control Systems (*EMCS*) and they lack personnel and metering infrastructure to measure demand and shape strategies for Demand Response. Also, small buildings are operated like residential buildings. The owner, with limited information mainly derived from the electricity bill, has to decide upon the Demand Response strategy. Finally, the internet availability is limited. On the other hand, large buildings are equipped with Energy

Management and Control Systems, there is ever present personnel and wildly deployed metering infrastructure to measure demand and shape strategies for Demand Response. Also, large buildings provide a plethora of information upon which the Demand Response strategy can be based [31].

Additionally, years of research on Demand Response strategies for large commercial buildings resulted in an understanding of systems and strategies that are applicable to those systems. The same understanding for small buildings does not exist [31].

A typical deployment for a commercial OpenADR implementation consist of the following entities [31]:

● Control Panel: A Control Panel that can accept over a user interface (*UI*) the Demand Response strategy and can communicate with the other devices in the facility wirelessly.

● Control equipment: The control equipment is a device that accept signals from a Control Panel and implements the corresponding Demand Response command. A typical example of control equipment is a Programmable Communicating Thermostat (*PCT*) that communicates with the Control Panel via Z-Wave wireless communications.

● Smart Meter: A Smart Meter records the Demand Response information for the whole building in real time and at given time intervals. The owner of the building owns the data and allows a third party to have access to it.

● Advanced Telemetry Meter Data and Device Server: An Advanced Telemetry Meter Data and Device Server collects meter and device status information and data from each facility. An Advanced Telemetry Meter Data and Device Server can act as an Aggregator for a group of facilities.

● Demand Response Automation Server: A Demand Response Automation Server is responsible for managing Demand Response events and for providing Demand Response signals to the facilities. All Demand Response signals from the Demand Response Automation Server are following the OpenADR standard. All communications between the various servers and the facility is made via the Internet by a broadband connection in the facility. Also, a Demand Response Automation Server is designed and developed to manage Demand Response events for a group of facilities, for each facility independently and as one entity. Figure 4.13 shows a typical OpenADR Demand Response commercial implementation.

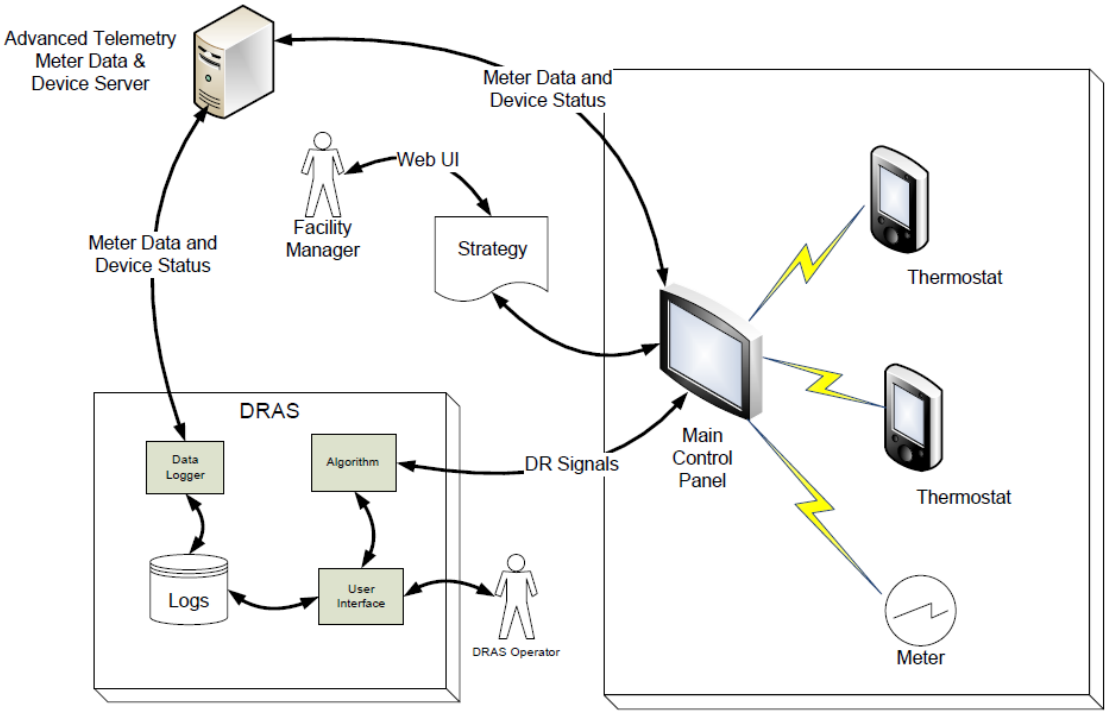

Figure 4.13: *A typical OpenADR Demand Response commercial implementation [31]*

To initiate a Demand Response event, an operator inputs general Demand Response event parameter such as event date, start and end time Demand Response strategy. Figure 4.14 shows the user interface of a Demand Response Automation Server.

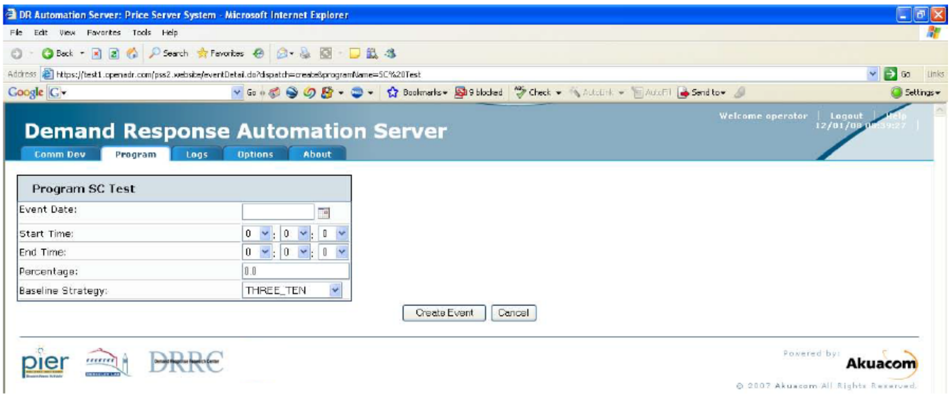

Figure 4.14: *A Demand Response Automation Server user interface [31]*

Various Demand Response control programs and strategies can be programmed. The operator, via the Demand Response Automation Server user interface and monitor the facility, observe the facility's response during the event and dynamically change the control parameters. Figure 4.15 shows the user interface of a Demand Response Automation Server with multiple Demand Response events.

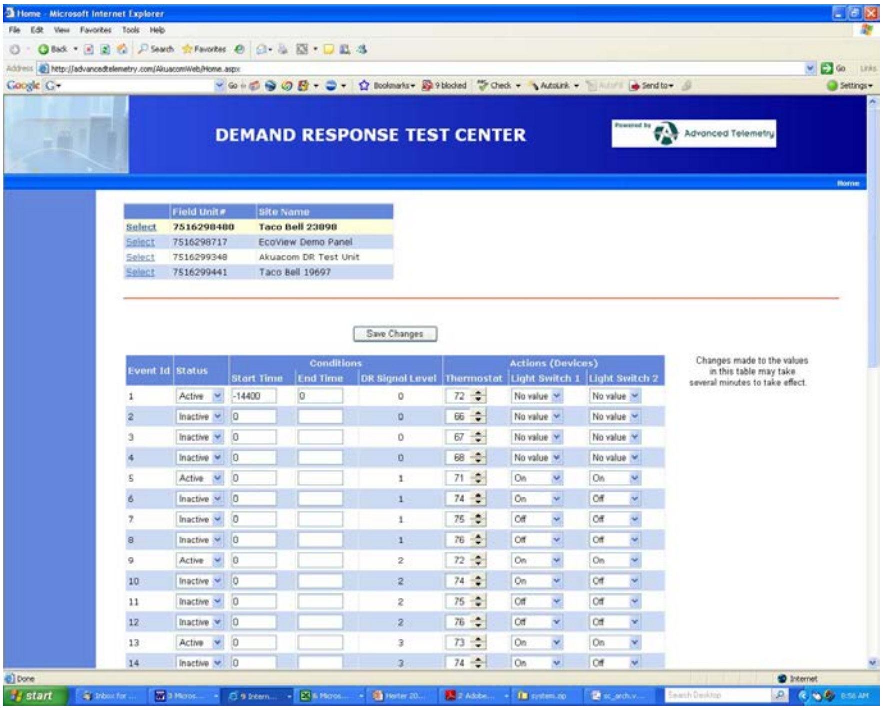

Figure 4.15: *A Demand Response Automation Server user interface with multiple Demand Response events [31]*

Table 4.5 shows examples of Demand Response event and strategies and table 4.6 shows examples of Demand Response events monitoring during the events.

Site	Test Date	Precooling	Start time	End Time	DR strategy
San Juan Capistrano	10/16/2008	1-2pm	2pm	3:15pm	Pre-cool at 72Deg starting at 1 pm and set up temp . 76F from 2-2:50, 77F from 2:50-3, 78F from 3-3:15, event canceled at 3:15
Hesperia	10/16/2008	No	2pm	3:15pm	76F from 2-2:50, 77F from 2:50-3, 78F from 3-3:15, event canceld at 3:15
San Juan Capistrano	10/17/2008	1-2 pm	2pm	4pm	Pre-cool starting at 1 pm and set up temp at 70Deg. 1-74, 2-75, 3-76, 4-77
Hesperia	10/17/2008	1-2 pm	2pm	4pm	precool starting at 1 pm and set up temp at 70Deg 1-76, 2-76, 3-77, 4 77
San Juan Capistrano	10/22/2008	2-3pm	3pm	5pm	Pre-cool at 70F starting at 2pm. 3-5pm 10% Shed from 3/10 MA baseline
Hesperia	10/22/2008	2-3pm	3pm	5pm	Pre-cool at 70F starting at 2pm. 3-5pm 10% Shed from 3/10 MA baseline
San Juan Capistrano	10/24/2008	11am-noon	noon	2pm	Pre-cool at 70F starting at 11am. Noon to 2pm, 10% Shed from 3/10 baseline
Hesperia	10/24/2008	11am-noon	noon	2pm	Pre-cool at 70F starting at 11am. Noon to 2pm, 10% Shed from 3/10 baseline
San Juan Capistrano	11/7/2008	8am-noon	noon	2pm	Pre-cool at 70F starting at 8am. Noon to 2pm, 10% Shed from 3/10 baseline
Hesperia	11/7/2008	8am-noon	noon	2pm	Pre-cool at 70F starting at 8am. Noon to 2pm, 10% Shed from 3/10 baseline
Hesperia	11/14/2008	8am-noon	noon	2pm	Pre-cool at 70F starting at 8am. Noon to 2pm, 19% Shed from 3/10 baseline

Table 4.5: *Example of Demand Response events and strategies [31]*

Site	Date	Baseline	Shed requested	Shed achieved	Pre-cool Kitchen Temp (DegF)	Precool Dining Temp (DegF)	Max Kitchen Temp (DegF)	Max Dining Temp (DegF)	Max OAT (DegF)
SJC	22-Oct	3/10_MA	10%	8%	75	74	80	78	78
SJC	24-Oct	3/10	10%	7%	74	74	79	78	100
SJC	7-Nov	3/10	10%	7%	76	72	79	78	79
Hesperia	22-Oct	3/10_MA	10%	13%	73	70	78	78	78
Hesperia	24-Oct	3/10	10%	11%	72	71	78	77	82
Hesperia	7-Nov	3/10	10%	17%	75	70	80	77	66
Hesperia	14-Nov	3/10	19%	18%	74	71	77	78	70

Table 4.6: *Example of Demand Response events monitoring during the events [31]*

After a Demand Response event is programmed and before it takes place, the Demand Response Automation Server notifies the facility about the pending event. Then, the event begins and the Demand Response Automation Server sends an initial control signal to the facility. The Demand Response Automation Server calculates the demand since the start of the event and compares it against a baseline. If the facility is not responding enough in respect to the baseline that is set then the Demand Response Automation Server sends a more intrusive Demand Response control signal to the facility. If the facility is responding more that it should then the Demand Response Automation Server relaxes the control strategy by sending the appropriate control signal. This process of continuous checking of the demand of the facility by the Demand Response Automation Server continues for the complete duration of the Demand Response event. Finally, when the event is scheduled to finish, the Demand Response Automation Server send a signal to the facility to end the event. Thus, the facility returns to normal operation [31].

For commercial buildings, three basic models for implementing Demand Response are identified [31]:

● Demand Response strategy is implemented completely within the load controllers. Figure 4.16 shows the implementation of the Demand Response strategy within the load controllers.

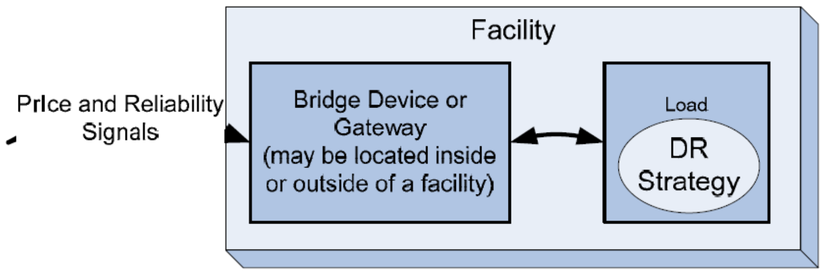

Figure 4.16: *Demand Response strategy within the load controllers [31]*

● Use of centralized controller within the facility to program and control the Demand Response strategies for the entire facility. Figure 4.17 shows the use of centralized controller to program and control the Demand Response strategies.

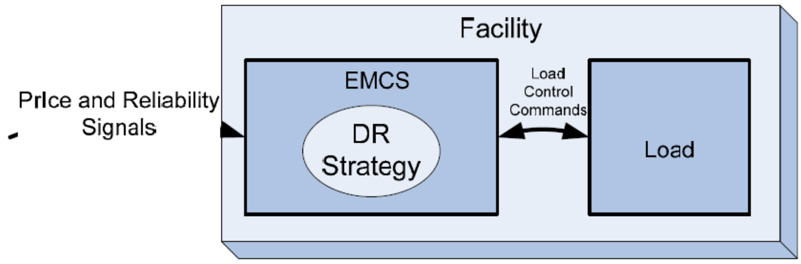

Figure 4.17: *Use of centralized controller to program and control the Demand Response strategies [31]*

● The Demand Response control strategy is implemented completely outside the facility. Figure 4.18 shows the implementation model that the Demand Response strategy is implemented outside the facility.

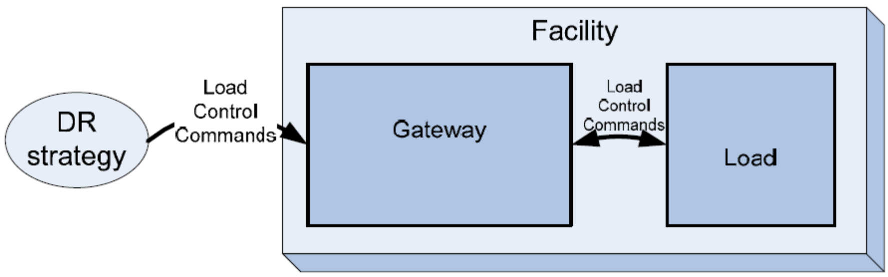

Figure 4.18: *Demand Response strategy is implemented outside the facility [31]*

The difference between these three implementations is where the Demand Response signals are converted into control signals. Which implementation will be materialized is heavily influenced by the existing technological infrastructure of the given scenario.

The first implementation methodology is based upon standalone communicating load controllers that are able to receive OpenADR signals and that contain the Demand Response strategy. Thus, this implementation methodology may not require a bridge client that requires some level of configuration to distribute messages. Thus, the stand alone controllers must have enough intelligence to accept Demand Response event information from the Demand Response Automation Server. Also, each load controller must be pre-programmed. With this implementation methodology is very difficult to implement large Demand Response due to the lack of centralized control. Finally, when the loads to be controlled are away from each other, the installation may require more than one client for each facility thus increasing the implementation cost [31].

The second implementation methodology uses an Energy Management Control System that provides centralized control. An Energy Management Control System by definition is easy to program and not necessarily rely on a computer to display the user interface and the control strategies. Also, an Energy Management Control System is able to receive standard Demand Response event information such as OpenADR signals and gives to the operator the ability to make decisions about the control strategy. More advanced Energy Management Control Systems can be found in large commercial buildings and are able to host a smart client to pull Demand Response signals [31].

Finally, in the third implementation methodology the Demand Response strategy is implemented outside the building at an external server. Thus, the facility does not receive any price or reliability signals, it receives only control signals. But, the external server needs to have access to all site specific device models and description of inputs and outputs for each load controller. Also, usually Demand Response control strategies require feedback from the facility to the server with information about system status. This approach requires minimum on site installations but all decisions are made in an external server. Therefore, controlling the Demand Response strategy can be problematic [31].

In terms of communications infrastructure needed for an implementation of Demand Response with OpenADR, all technologies that support Internet Protocol and can utilize the internet as mean of communication are supported and can be used [31]. Table 4.7 shows a summary of communication means.

Type	Must be dedicated to DR devices	Two Way	Installation costs	Monthly costs	Costs to implement in devices
T-carrier	No	Yes	High if dedicated	High if dedicated	Low (Ethernet)
DSL	No	Yes	Medium if dedicated	Medium if dedicated	Low (Ethernet)
Cable	No	Yes	Medium if dedicated	Medium if dedicated	Low (Ethernet)
ISDN	No	Yes	Medium if dedicated	Medium if dedicated	Low (Ethernet)
Fiber	No	Yes	High if dedicated	High if dedicated	Low (Ethernet)
Satellite	No	Maybe	High if dedicated	Medium if dedicated	Low (Ethernet)
WiMax	No	Yes	Medium if dedicated	Medium if dedicated	Low (Ethernet)
Mobile	Yes	Yes	Low	Medium	High
POTS	No	Yes	Low	Medium	Low if not dedicated
BPL	No	Yes	Medium if dedicated	Medium if dedicated	Low (Ethernet)

Table 4.7: *Summary of communication means [31]*

To conclude, a commercial implementation of Demand Response with OpenADR is heavily enabled by control technologies and communication means. Thus, lack of technology is not an obstacle. The lack of awareness of options and the benefits of Demand Response are obstacles. Also, especially for small commercial implementations, the cost is a significant barrier [31]. Table 4.8 and 4.9 shows example results from commercial implementations of Demand Response.

Business Type	Program	Average Monthly kWh		2007-2008 Difference		2007-2008 Difference Corrected for Non-Participant Change *(%)*
		Summer 2007	Summer 2008	(kWh)	(%)	
Office	None (control)	1025	976	49*	-5%	
	4° ACC	934	631	303*	-32%	-27%
	CPP	1061	668	393*	-37%	-32%
Restaurant	None (control)	3340	3252	88*	-3%	
	4° ACC	3249	2907	342	-11%	-8%
	CPP	3377	2944	432*	-13%	-10%
Retail	None (control)	1754	1716	38*	-2%	
	4° ACC	1663	1370	292	-18%	-15%
	CPP	1790	1408	383*	-21%	-19%
Average	4° ACC and CPP participants	1606	1238	369*	-23%	-20%

Table 4.8: *Example of commercial implementation Demand Response results (1) [41]*

Business Type	Program	Average Hourly Usage (kWh/h)		Impact	
		Normal	Event	(kWh/h)	(%)
Office	4° ACC	2.1	1.3	0.8	-38%
	CPP	2.4	1.8	0.6	-24%
Restaurant	4° ACC*	8.9	8.8	0.1	-1%
	CPP	13.4	13.1	0.4	-3%
Retail	4° ACC	3.5	2.7	0.8	-22%
	CPP	5.8	5.0	0.8	-14%

Table 4.9: *Example of a commercial Demand Response implementation results (2) [41]*

4.1.9 The OpenADR 2.0 A profile specification

An OpenADR 2.0 A device needs to support limited EiEvent services, Hypertext Transfer Protocol as a transport mechanism and a standard security level. Extensible Messaging and Presence Protocol as a transport mechanism and high security level are optional [32].

- EiEvent service:

Events are created by a Virtual Top Node and sent to the Virtual End Node. The oadrDistributeEvent payload contains one or more events described by the oadrEvent element. The oadrResponseRequired element contains information about which events require a response and which events do not require a response. If a response is required, the Virtual End Node responds with an oadrCreatedEvent paload containing eventResponse elements for each oadrEvent [32]. Figure 4.19 shows the EiEvent push pattern.

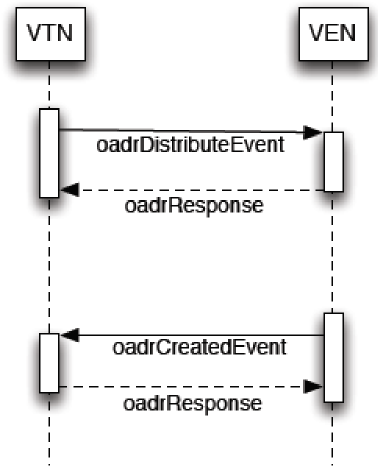

Figure 4.19: *EiEvent push pattern [32]*

The pull case is initiated by a Virtual End Node that requests events by sending an oadrRequestEvent to a Virtual Top Node. Then, the Virtual Top Node responds with an oadrDistributeEvent. After the pull case is completed the response from the Virtual End Node is the same as in the push case [32]. Figure 4.20 shows the EiEvent pull pattern.

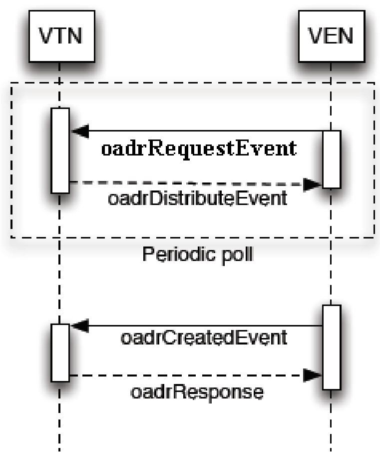

Figure 4.20: *EiEvent pull pattern [32]*

When a response is required for an event, an initial oadrCreatedEvent is always sent from the Virtual End Node to the Virtual Top Node. If a Virtual End Node wants to change its opt-state during an event it sends a subsequent oadrCreatedEvent message containing the new state for a given event [32]. Code snippets 4.1, 4.2, 4.3 and 4.4 shows various payloads.

```
<!--  ******* oadrResponse ******** -->
<xs:element name="oadrResponse">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="ei:eiResponse"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Code snippet 4.1: *The oadrResponse payload [32]*

```
<!--  ******* oadrDistributeEvent ******** -->
<xs:element name="oadrDistributeEvent">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="ei:eiResponse" minOccurs="0"/>
         <xs:element ref="pyld:requestID"/>
         <xs:element ref="ei:vtnID"/>
         <xs:element name="oadrEvent" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
               <xs:sequence>
                  <xs:element ref="ei:eiEvent"/>
                  <xs:element name="oadrResponseRequired"
type="oadr:ResponseRequiredType">
                     <xs:annotation>
                        <xs:documentation>oadr: This defines when repsonses are re-
quired Can be always or never</xs:documentation>
                     </xs:annotation>
                  </xs:element>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Code snippet 4.2: *The oadrDistributeEvent payload [32]*

```
<!--  ******* oadrCreatedEvent ******** -->
<xs:element name="oadrCreatedEvent">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="pyld:eiCreatedEvent"/>
      </xs:sequence>
  </xs:complexType>
</xs:element>
```

Code snippet 4.3: *The oadrCreatedEvent payload [32]*

```
<!--  ******* oadrRequestEvent ******** -->
<xs:element name="oadrRequestEvent">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="pyld:eiRequestEvent"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Code snippet 4.4: *The oadrRequestEvent payload* [33]

An oadrDistributeEvent will always contain on its' payload all events that apply to the Virtual End Node which are pushed or pulled. Also, an oadrDistributeEvent has the following components [32]:

● A requestID is set by the Virtual Top Node in order to uniquely identify this request and any contained events and is used by the Virtual End Node for the oadrCreatedEvent event responses.

● A vtnID identifying the Virtual Top Node that sends the request.

● oadrEvent elements.

An oadrEvent element describes the event, the signal values and the time periods that apply to the signals. Also, every oadrEvent has an EiEvent element that contains detailed information about the event and an oadrResposeRequired element that indicates if a Virtual End Node must respond with an oadrCreatedEvent. An oadrResposeRequired element can take the values "always" and "never" [32].

An EiEvent eventDescriptor contains the following fields [32]:

● eventID: a unique ID for this event within the context of a Virtual Top Node.

● modificationNumber: A sequence that starts at zero and is incremented by 1 each time the Virtual Top Node modifies the event.

● Priority: An indication of the event priority with 0 being no priority.

● marketContext: Identifies a particular program or application defined grouping that is related to an event.

● createdDateTime: The time the payload containing the event was created.

● eventStatus: The status of the event, indicating if the event is “near”, “far”, “active” or “cancelled”.

● testEvent: If not false, indicates this is a test event.

● vtnComment: Arbitrary comment provided by the VTN.

An EiActivePeriod defines the start time and the duration of an event. The event signals that are applied during the active period are defiened in an EiEventSignals element. An EiEventSignals element contains EiEventSignal elements, each one of them contains a sequence of durations and the sum of them must be equal to the full duration of the active period of the event. Also, every EiEventSignal element contains a signalType with information about level or price. The signalPayload contains relative values for the duration that can take the values "normal", "low", "moderate" and "special" [32].

An EiTarget element can be used to explicitly specify the entities targeted by an event and it may contain one or more venIDs, groupIDs, resourceIDs, or partyIDs. When a Virtual Top Node is used as an aggregator and wants to target multiple Virtual End Nodes then an EiTarget can be used. If an EiTarget is not present then the Virtual End Node is the only resource targeted by the Virtual Top Node [32]. Code snippets 4.5, 4.6, 4.7 and 4.8 shows various payloads.

```
<!--  ******* eiEvent ******** -->
<xs:element name="eiEvent" type="ei:eiEventType"/>
<xs:complexType name="eiEventType">
     <xs:sequence>
          <xs:element ref="ei:eventDescriptor"/>
          <xs:element ref="ei:eiActivePeriod"/>
          <xs:element ref="ei:eiEventSignals"/>
          <xs:element ref="ei:eiTarget"/>
     </xs:sequence>
</xs:complexType>
```

Code snippet 4.5: *The EiEvent payload [32]*

```
<!-- ***** eventDescriptor ***** -->
<xs:element name="eventDescriptor" type="ei:eventDescriptorType"/>
<xs:complexType name="eventDescriptorType">
     <xs:sequence>
          <xs:element ref="ei:eventID"/>
          <xs:element ref="ei:modificationNumber"/>
          <xs:element name="priority" type="xs:unsignedInt" minOccurs="0"/>
          <xs:element name="eiMarketContext">
               <xs:complexType>
                    <xs:sequence>
```

```
                        <xs:element ref="emix:marketContext"/>
                   </xs:sequence>
              </xs:complexType>
          </xs:element>
          <xs:element name="createdDateTime" type="xcal:DateTimeType"/>
          <xs:element ref="ei:eventStatus"/>
          <xs:element name="testEvent" type="xs:string" minOccurs="0"/>
          <xs:element name="vtnComment" type="xs:string" minOccurs="0"/>
     </xs:sequence>
</xs:complexType>
```

Code snippet 4.6: *The eventDescriptor payload [32]*

```
<!-- ***** eiActivePeriod *****-->
<xs:element name="eiActivePeriod" type="ei:eiActivePeriodType"/>
<xs:complexType name="eiActivePeriodType">
     <xs:sequence>
          <xs:element ref="xcal:properties"/>
          <xs:element ref="xcal:components"/>
     </xs:sequence>
</xs:complexType>
<!-- ***** eiEventSignals ***** -->
<xs:element name="eiEventSignals" type="ei:eiEventSignalsType"/>
<xs:complexType name="eiEventSignalsType">
     <xs:sequence>
          <xs:element ref="ei:eiEventSignal" maxOccurs="unbounded"/>
     </xs:sequence>
</xs:complexType>
```

Code snippet 4.7: *The eiActivePeriod payload [32]*

```
<!-- ***** eiTarget *****-->
<xs:element name="eiTarget" type="ei:eiTargetType"/>
<xs:complexType name="eiTargetType">
     <xs:sequence>
          <xs:element ref="ei:groupID" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="ei:resourceID" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="ei:venID" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="ei:partyID" minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
</xs:complexType>
```

Code snippet 4.8: *The eiTarget payload [32]*

An oadrCreatedEvent contains oadrResponse elements that contain a responseCode, a responseDescription and a requestID for every event that gets acknowledged. The eventResponse elements are paired to events using the qualifiedEventID which contains an eventID and a modificationNumber. The optType may have a value of “optIn” or “optOut” to indicate the Virtual End Node action for a given event [32]. Code snippets 4.9, 4.10, 4.11 and 4.12 shows various payloads.

```
<!-- ******* resourceID ******* -->
<xs:element name="resourceID" type="xs:string"/>
<!-- ******* groupID *******-->
<xs:element name="groupID" type="xs:string"/>
<!-- *******  partyID *******  -->
<xs:element name="partyID" type="xs:string"/>
<xs:simpleType name="EiExtensionTokenType">
     <xs:annotation>
          <xs:documentation>Pattern used for extending string enumeration, where al-
lowed</xs:documentation>
     </xs:annotation>
     <xs:restriction base="xs:token">
          <xs:pattern value="x-\S.*"/>
     </xs:restriction>
</xs:simpleType>
<!--  ******* venID ******** -->
<xs:element name="venID" type="xs:string"/>
<!--  ******* vtnID ******** -->
<xs:element name="vtnID" type="xs:string"/>
<!--  ******* eventID ******** -->
<xs:element name="eventID" type="xs:string"/>
<!--  ******* modificationNumber ******** -->
<xs:element name="modificationNumber" type="xs:unsignedInt"/>
<!--  ******* qualifiedEventID ******** -->
<xs:element name="qualifiedEventID" type="ei:QualifiedEventIDType"/>
<!--  ******* QualifiedEventIDType ******** -->
<xs:complexType name="QualifiedEventIDType">
     <xs:annotation>
          <xs:documentation>Fully Qualified Event ID includes the eventID and the Mod-
ification Number</xs:documentation>
     </xs:annotation>
     <xs:sequence>
          <xs:element ref="ei:eventID"/>
          <xs:element ref="ei:modificationNumber"/>
     </xs:sequence>
</xs:complexType>
```

Code snippet 4.9: *Various IDs payloads [32]*

```
<!--  ******* responseDescription ******** -->
<xs:element name="responseDescription" type="xs:string"/>
<!--  ******* optType ******** -->
<xs:element name="optType" type="ei:OptTypeType"/>
<xs:simpleType name="OptTypeType">
     <xs:restriction base="xs:token">
          <xs:enumeration value="optIn"/>
          <xs:enumeration value="optOut"/>
     </xs:restriction>
</xs:simpleType>
```

Code snippet 4.10: *The responseDescirption payload [32]*

```
<!--  ******* eiResponse ******** -->
<xs:element name="eiResponse">
     <xs:complexType>
          <xs:sequence>
               <xs:element ref="ei:responseCode"/>
               <xs:element ref="ei:responseDescription" minOccurs="0"/>
               <xs:element ref="pyld:requestID"/>
          </xs:sequence>
     </xs:complexType>
</xs:element>
```

Code snippet 4.11: *The eiResponse payload [32]*

```
<!--  ******* eventResponses ******** -->
<xs:element name="eventResponses">
     <xs:complexType>
          <xs:sequence>
               <xs:element name="eventResponse" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                         <xs:sequence>
                              <xs:element ref="ei:responseCode"/>
                              <xs:element ref="ei:responseDescription" minOccurs="0"/>
                              <xs:element ref="pyld:requestID"/>
                              <xs:element ref="ei:qualifiedEventID"/>
                              <xs:element ref="ei:optType"/>
                         </xs:sequence>
                    </xs:complexType>
               </xs:element>
          </xs:sequence>
     </xs:complexType>
</xs:element>
```

Code snippet 4.12: *The eventResponse payload [32]*

An initial oadrCreatedEvent response must be sent for each event that requires acknowledgment. Afterwards, EiCreatedEvents may also be sent to change the opt-state of a Virtual End Node. The grouping of events in an oadrCreatedEvent is completely up to the Virtual End Node and does not necessarily correspond to the grouping of events in an oadrDistributeEvent. The Virtual End Node is free to send one event per payload or group multiple pending events into a single oadrCreatedEvent payload [32]. Code snippet 4.13 shows the eiCreateEvent payloads.

```
<!--  ******* eiCreatedEvent ******** -->
<xs:element name="eiCreatedEvent">
     <xs:complexType>
          <xs:sequence>
               <xs:element ref="ei:eiResponse"/>
               <xs:element ref="ei:eventResponses" minOccurs="0"/>
               <xs:element ref="ei:venID"/>
          </xs:sequence>
     </xs:complexType>
</xs:element>
```

Code snippet 4.13: *The eiCreateEvent payload [32]*

4.1.10 The OpenADR 2.0 B profile specification

The only supported service by the OpenADR 2.0 A profile is the EiEvent that is simplified by the following ways [33]:

- Only one signal per event is allowed.

● The event targeting is limited by venID, groupID, resourceID and partID.

● Targeting at the signal level with device classes is not supported.

● Baselines are not supported.

● Modification with modificationDateTime and modificationReason are not supported.

OpenADR 2.0 B profile Virtual Top Nodes must be compatible with OpenADR 2.0 A profile Virtual End Nodes while OpenADR 2.0 B profile Virtual End Nodes can optionally choose to be compatible with OpenADR 2.0 A profile Virtual End Nodes [33].

● EiEvent service:

Events are created by a Virtual Top Node and sent to a Virtual End Node. The oadrDistributeEvent payload contains one or more events described by the oadrEvent element. The oadrResponseRequired element contains information about which events require a response and which event do not require a response. If a response is required, the Virtual End Node responds with an oadrCreatedEvent paload containing eventResponse elements for each oadrEvent [33].

For the push pattern interaction, a Virtual Top Node will send events to a Virtual End Node via an oadrDistributeEnent payload. The response, if required, for an oadrDistributeEnent event is a transport level acknowledgement, in Hypertext Transfer Protocol a 200 response code. For the pull pattern interaction, an oadrDistributeEnent will be sent from a Virtual Top Node to the Virtual End Nose as response to an oadrPoll. Also, a Virtual End Node can send an oadrRequestEvent to a Virtual Top Node in order to pull events from it. If a response is required, then, the Virtual End Node sends an oadrCreatedEvent in second message to the Virtual Top Node [33]. Figure 4.21 shows the EiEvent push pattern.

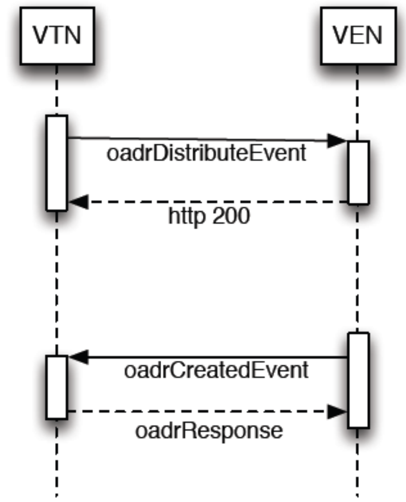

Figure 4.21: *EiEvent push pattern [33]*

As far as the pull case is concerned, the Virtual End Node requests events by sending an oadrPoll to the Virtual Top Node and the Virtual Top Node responds with an oadrDistriduteEvent. From this point on, the interaction is the same as in the push pattern interaction [33]. Figure 4.22 shows the EiEvent pull pattern.

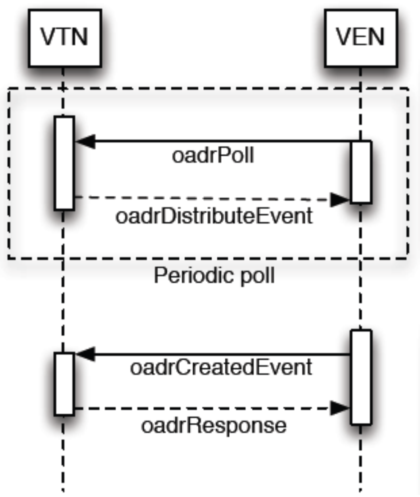

Figure 4.22: *EiEvent pull pattern [33]*

When a response is required for an event, an initial oadrCreatedEvent is always sent from the Virtual End Node to the Virtual Top Node. If the Virtual End Node wants to change its opt-state during an event it sends a subsequent oadrCreatedEvent message containing the new state for the given event [33].

An oadrDistributeEvent will always contain on its' payload all events that apply to the Virtual End Node that are pushed or pulled. Also, an oadrDistributeEvent has the following components [33]:

● A requestID that is set by the Virtual Top Node in order to uniquely identify this request and any contained events and is used by the Virtual End Node for the oadrCreatedEvent event responses.

● A vtnID identifying the Virtual Top Node that sends the request.

● oadrEvent elements.

An oadrEvent element describes an event, the signal values and time periods that apply to the signals. Also, every oadrEvent has an EiEvent element that contains detailed information about the event and an oadrResposeRequired element that indicates if a Virtual End Node must respond with an oadrCreatedEvent. An oadrResposeRequired element can take the values "always" and "never" [33].

An EiEvent eventDescriptor contains the following fields [33]:

● eventID: a unique ID for this event within the context of a Virtual Top Node.

● modificationNumber: A sequence that starts at zero and is incremented by 1 each time the Virtual Top Node modifies the event.

● priority: An indication of the event priority with 0 being no priority.

● marketContext: Identifies a particular program or application defined grouping that is related to an event.

● createdDateTime: The time the payload containing the event was created.

● eventStatus: The status of the event, indicating if the event is “near”, “far”, “active” or “cancelled”.

● testEvent: If not false, indicates this is a test event.

● vtnComment: Arbitrary comment provided by the VTN.

An EiActivePeriod defines the start time and the duration of an event. The event signals that are applied during the active period are defined in an EiEventSignals element. An EiEventSignals element contains EiEventSignal elements that each one of them contains a sequence of durations and the sum of them must be equal to the full duration of the active period of the event. Also, each EiEventSignal element contains a signalType with information about level or price. The signalPayload contains relative values for the duration that can take the values "normal", "low", "moderate" and "special"[33].

An EiTarget element can be used to explicitly specify the entities that the event targets and it may contain one or more venIDs, groupIDs, resourceIDs, or partyIDs. When a Virtual Top Node is used as an aggregator and wants to target multiple Virtual End Nodes then an EiTarget can be used. If an EiTarget is not present then the Virtual End Node is the only resourse targeted by the Virtual Top Node [33].

An oadrCreatedEvent contains oadrResponse elements that contain a responseCode, a responseDescription and a requestID for each event that gets acknowledged. The eventResponse elements are paired to events using thequalifiedEventID which contains an eventID and a modificationNumber. The optType may have a value of “optIn” or “optOut” to indicate the Virtual End Node action for a given event [33].

An initial oadrCreatedEvent response must be sent for each event that requires acknowledgment. Afterwards, EiCreatedEvents may also be sent to change the opt-state of a Virtual End Node. The grouping of events in an oadrCreatedEvent is completely up to the VEN and does not necessarily correspond to the grouping of events in an oadrDistributeEvent. The Virtual End Node is free to send one event per payload or group multiple pending events into a single oadrCreatedEvent payload [33].

The differences between the OpenADR 2.0 A profile event mechanism and the OpenADR 2.0 B profile event mechanism are [33]:

● Event signal type can be different than SIMPLE type.

● Events can contain multiple event elements.

● Event signals can contain EiTarget elements, different from the event-level EiTarget. Also, the target types must be constrained to endDeviceAsset with the enumerated values for device type.

● Event signals can contain baseline elements.

● The event description can contain modificationDateTime and modificationReason elements.

● Apart from the oadrCreatedEvent, the EiOpt service can be used to send schedules from the Virtual End Node to the Virtual Top Node.

● The oadrPoll element is used in the pull exchange pattern.

The OpenADR 2.0 B profile EiTarget element includes several target types and device class can also be applied. If a Virtual End Node receives an event target that is not targeted for this Virtual End Node it should reject it and respond with the appropriate error code [33].

Also, within the OpenADR 2.0 B profile signals multiple attributes can be used. The attributes that are used are the eiEventSignal:signalName, eiEventSignal:signalType, and eiEventSignal:itemBase [33].

The report types supported by the OpenADR 2.0 B profile are intended to provide a specific type of report functionality that is supported by a Virtual Top Node or a Virtual End Node. Reports can be in the form of metadata or data reports and can report history usage and Green Button or telemetry usage and status [33].

● Metadata is used to specify reporting capabilities and is exchanged as part of the report registration process. This report type can report specifications for all report types. Every report within the metadata report has a reportSpecifierID that is used to provide identification for the report specification for all subsequent interactions. Also, each report profile has a reportName attribute that will be used in a metadata report for further references [33].

● Data reports report the actual data that is measured or calculated. The value that the data reports report is called "data point" and has attributes such as units. A data report can contain several data points while each one of them is represented in the schema with the rID attribute [33].

The core reporting operations are [33]:

● Registering Reporting Capabilities (Figure 4.23)

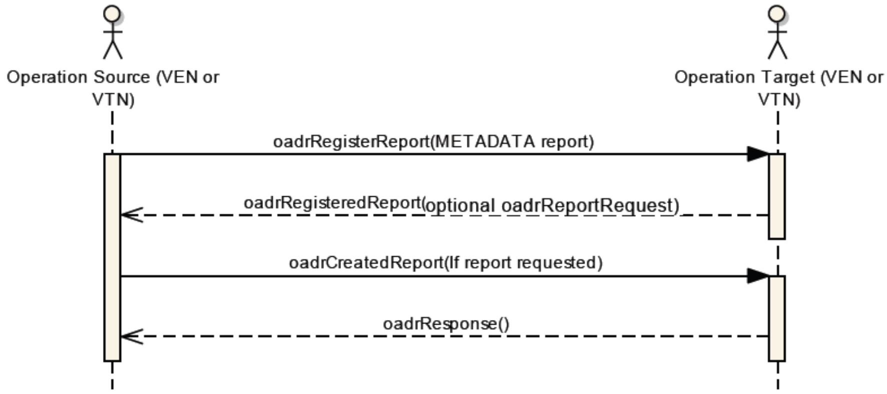

Figure 4.23: *Registering reporting capabilities use case [33]*

● Requesting Reports (Figure 4.24)

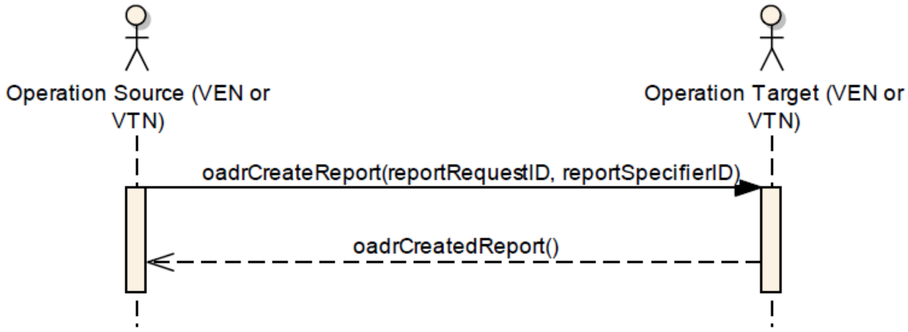

Figure 4.24: *Requesting reports use case [33]*

● Sending Reports (Figure 4.25)

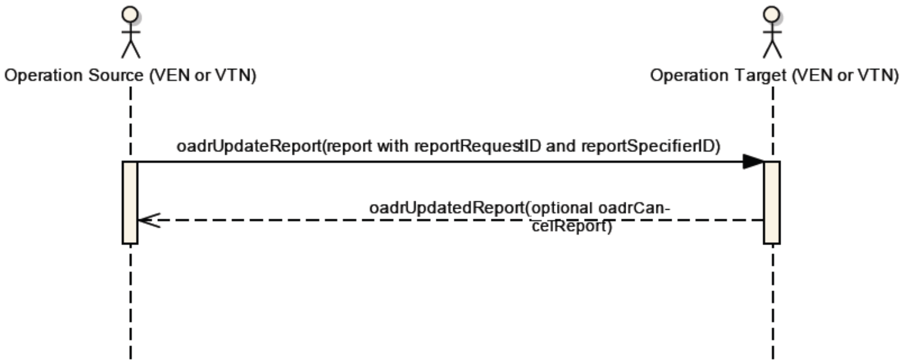

Figure 4.25: *Sending reports use case [33]*

● Cancelling Reports (Figure 4.26)

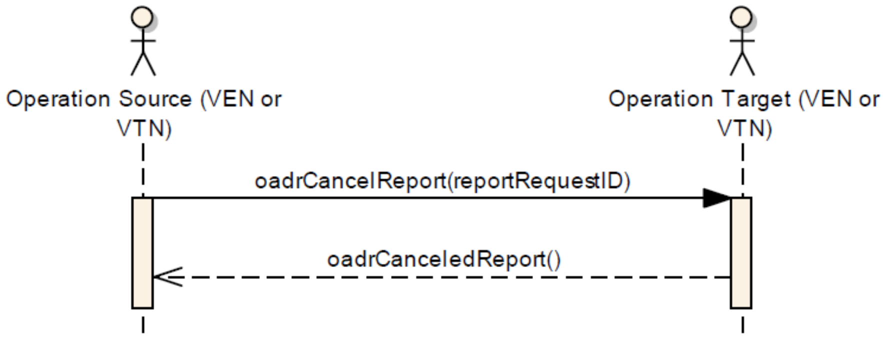

Figure 4.26: *Cancelling reports use case [33]*

The OpenADR 2.0 B profile reports registration of Virtual End Nodes to Virtual Top Nodes via the EiRegisterParty service. The supported operations are the oadrQueryRegistration, oadrCreatePartyRegistration, oadrCancelPartyRegistration and the oadrRequest Reregestration operation [33]. Table 4.10 shows the EiRegisterParty service request and response payloads.

Request Payload	Response Payload	Re-ques tor	Responder
oadrQueryRegistration	oadrCreatedPartyRegistration	VEN	VTN
oadrCreatePartyRegistration	oadrCreatedPartyRegistration	VEN	VTN
oadrCancelPartyRegistration	oadrCanceledPartyRegistration	VEN VTN	VTN VEN
oadrRequestReregestration	oadrResponse	VTN	VEN

Table 4.10: *The EiRegisterParty service request and response payloads [33]*

- Query registration (Figure 4.27)

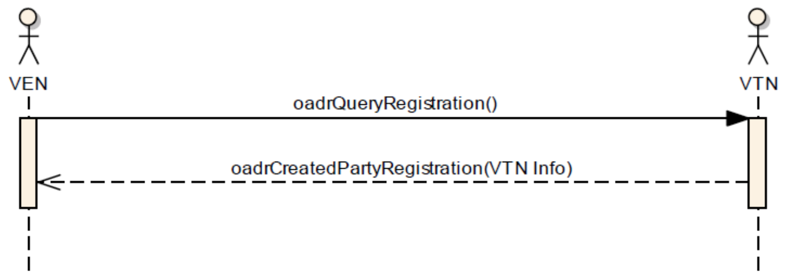

Figure 4.27: *Query registration use case [33]*

- Create registration (Figure 4.28)

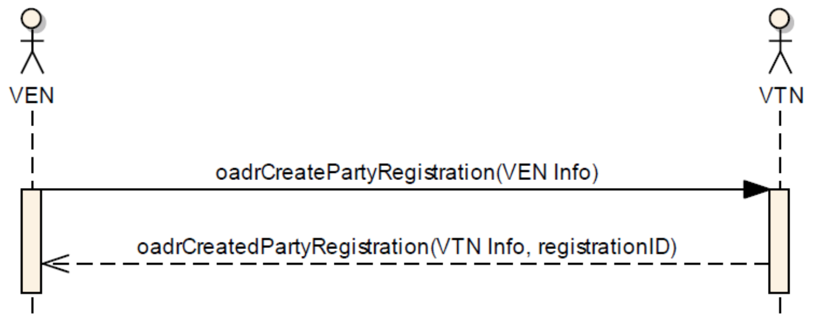

Figure 4.28: *Create registration use case [33]*

● Request registration (Figure 4.29)

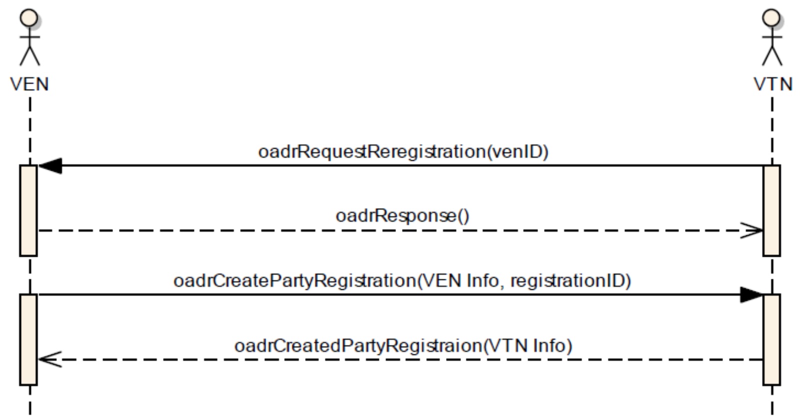

Figure 4.29: *Request registration use case [33]*

● Cancel registration (Figure 4.30)

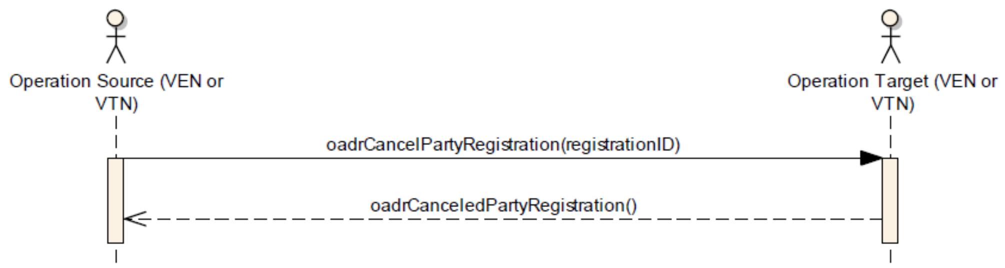

Figure 4.30: *Cancel registration use case [33]*

The OpenADR 2.0 B profile uses the EiOpt service in order to create and deliver Opt-In and Opt-Out schedules from the Virtual End Nodes to the Virtual Top Nodes. The supported service operations are the oadrCreateOpt and the oadrCancelOpt [33]. Table 4.11 shows The EiOpt service request and response payloads.

Request Payload	Response Payload	Requestor	Responder
oadrCreateOpt	oadrCreatedOpt	VEN	VTN
oadrCancelOpt	oadrCanceledOpt	VEN	VTN

Table 4.11: *The EiOpt service request and response payloads [33]*

- Create Opt (Figure 4.31)

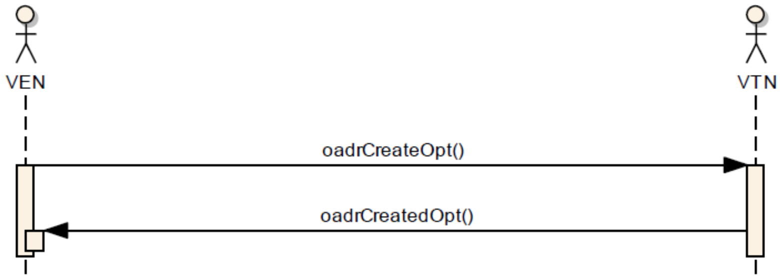

Figure 4.31: *Create Opt use case [33]*

- Cancel Opt (Figure 4.32)

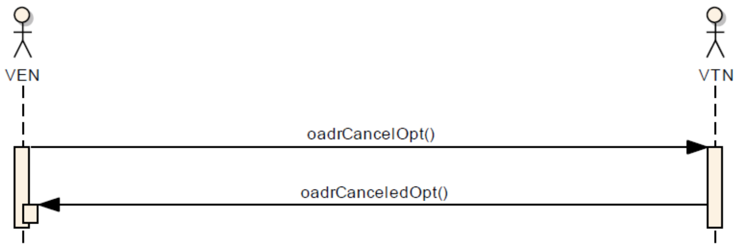

Figure 4.32: *Cancel Opt use case [33]*

Finally, pull patterns are periodically used by the Virtual End Nodes to periodically pull information from Virtual Top Nodes. The oadrPoll service provide this functionality [33]. Table 4.12 shows the oadrPoll service request and response payloads.

Request Payload	Response Payload	Requestor	Responder
oadrPoll	One of the following: • oadrResponse • oadrDistributeEvent • oadrCreateReport • oadrRegisterReport • oadrCancelReport • oadrUpdateReport • oadrCancelPartyRegistration • oadrRequestReregistration	VEN	VTN

Table 4.12: *The oadrPoll service request and response payloads [33]*

The oadrPoll request has as single sub-element the ID of the Virtual End Node that initiates the polling [33]. Code snippet 4.14 shows the oadrPoll payload. Figures 4.33, 4.34, 4.35 and 4.36 shows the various oadrPoll use cases.

```
<oadr:oadrPoll ei:schemaVersion="2.0b">
     <ei:venID>VEN_123</ei:venID>
</oadr:oadrPoll>
```

Code snippet 4.14: *The oadrPoll payload* [33]

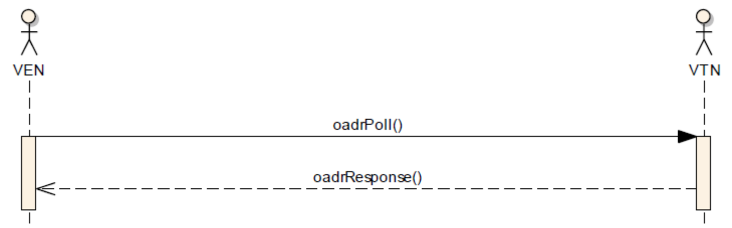

Figure 4.33: *The oadrPoll use case with nothing in queue [33]*

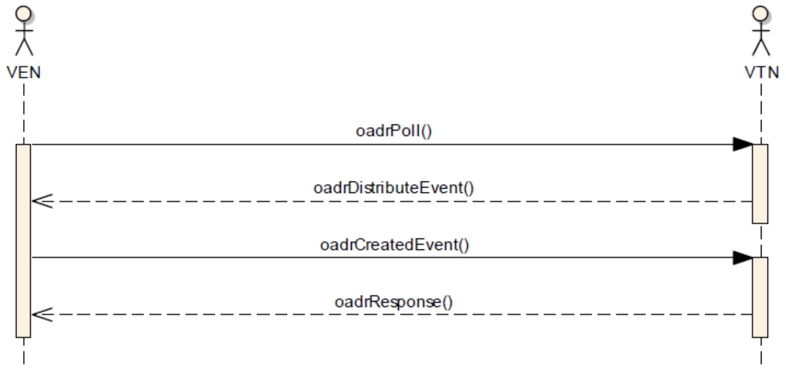

Figure 4.34: *The oadrPoll use case with an oadrDistributeEvent reply [33]*

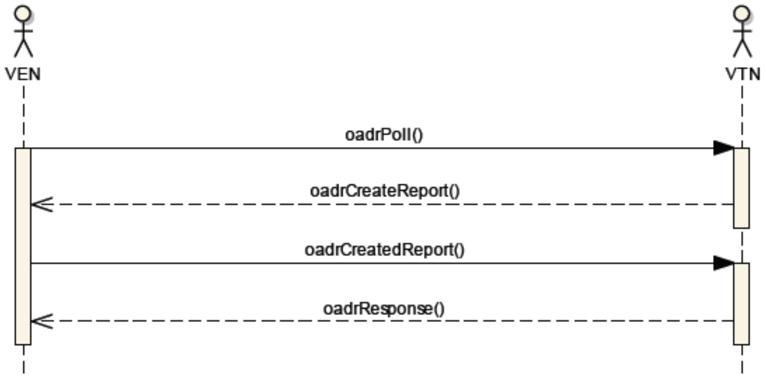

Figure 4.35: *The oadrPoll use case with an oadrCreateReport reply [33]*

Figure 4.36: *The oadrPoll use case with an oadrRequestReregistration reply [33]*

4.2 Other Demand Response Standards

The SEP standard is another Demand Response standard that is in many ways parallel to the OpenADR standard. The SEP standard is wildly used and accepted by the academic community and energy industry. However, there are many notable differences between the two before mentioned standards.

In this subchapter we will try to provide a quick overview of the SEP standard and identify all major differences between the SEP standard and the OpenADR standard.

4.2.1 The SEP Standard

The Smart Energy Profile (*SEP*) standard is a Demand Response standard that is oriented in providing software applications and code in support of pricing, Demand Response and other energy related applications. It is designed to operate within the smart building and provide device registration, monitoring and control [34].

The SEP standard is built upon the ZigBee low power wireless radio communication standard that is build upon the IEEE 802.15.4 standard and the HomePlug power line communication standard that is build upon the IEEE P1901 standard [34].

The SEP standard will provide Demand Response for residential and commercial buildings while clustering common functionalities between the two cases. Thus, a cluster library called ZigBee cluster library is created that cluster small common functional chunks [35].

Vendors and device manufacturers have the ability to test their devices and acquire a certification that the device is SEP approved and compatible [35]. The organization that coordinates all interoperability issues is the Consortium for SEP 2.0 (*CSEP*) [37].

The SEP standard has two versions, the firsts is SEP 1.0 and the latest is SEP 2.0. The SEP 2.0 standard has the following attributes [36]:

● All blocks used by SEP 2.0 standard are standard compliant.

● The SEP 2.0 standard will run on any Internet Protocol enabled physical layer like Wireless Fidelity (*WiFi*), HomePlug and 802.15.4 using ZigBee Internet Protocol.

● The SEP 2.0 standard supports Transmission Control Protocol (*TCP*), User Datagram Protocol (*UDP*) and Internet Protocol version 6 (*IPv6*).

● At the SEP 2.0 standard all layers in the Transmission Control Protocol / Internet Protocol must be certified.

● The SEP 2.0 standard supports the Advanced Encryption Standard at 128 bits (*AES128*) as a minimum security requirement for the Media Access Control (*MAC*) layer. Also it supports the Extensible Authentication Protocol (*EAP*) and the Transport Layer Security protocol for the Network layer security and for the Application layer security supports Secure Hypertext Transfer Protocol (*HTTPS*).

● The SEP 2.0 standard supports Secure Hypertext Transfer Protocol for data exchange, Extended Multicast Dynamic Name System (*xmDNS*) and Dynamic Name System Service Discovery (*DNS-SD*) for service discovery. It also supports Extensible Mark-up Language and Efficient Extensible Mark-up Language Interface (*EXI*) for payload.

● The SEP 2.0 standard commands are packages as Extensible Mark-up Language packets and they are delivered over Secure Hypertext Transfer Protocol.

The transition from SEP 1.0 to SEP 2.0 brought many changes to the security architecture, the network and application layers. Thus, there is no backwards compatibility from SEP 2.0 to SEP 1.0 [36].

Some of the SEP 2.0 features are [37]:

- Price communication
- Demand Response and load control
- Energy usage information and metering data
- Text messaging
- Prepayment metering
- Plug in electric vehicles
- Distributed energy resources
- Billing communication
- File download and update

4.2.2 Differences between the SEP and the OpenADR standards

The SEP 2.0 standard overlaps with the OpenADR 2.0 in terms of features due to the common goal that both share. The common goal is providing Demand Response. The main differences between SEP 2.0 and OpenADR standards can be identified in the functionality and the roles that each standard has [38].

The SEP 2.0 standard is focus in communications between devices within a Home Area Network while OpenADR 2.0 is focus in communications between an operator and a customer. Also, SEP 2.0's features make it more suitable for a residential case while OpenADR 2.0 provides a variety of Demand Response signals and market rules that makes it more suitable for a commercial case [38, 39]. Figure 4.37 shows the differences between the scope of the SEP 2.0 and OpenADR standards.

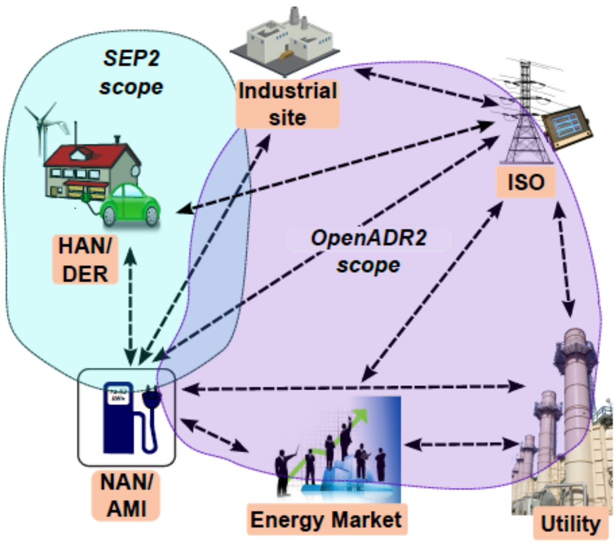

Figure 4.37: *Differences between the scope of the SEP 2.0 and OpenADR standards [39]*

An other difference between the SEP 2.0 standard and the OpenADR 2.0 standard is that SEP 2.0 supports Direct Load Control while OpenADR 2.0 provides limited Direct Load Control because the signals from the operator have to be translated into Direct Load Control Commands and the customer side. Also, the SEP 2.0 standard is focused on an Internet of Things (*IoT*) implementation, it supports Extensible Mark-up Language compression with the Efficient Extensible Mark-up Language Interchange [38, 39].

Also, the SEP 2.0 standard provides a modular profile structure while the OpenADR 2.0 standard provides a tiered profile structure. As far as reporting capabilities are concerned, the SEP 2.0 standard does not support full reporting services while the OpenADR 2.0 standard supports full reporting services. As a transport layer protocol, the SEP 2.0 standard uses a Hypertext Transfer Protocol that is REST styled with create, read update and delete (*CRUD*) functionalities while the OpenADR 2.0 standard uses Hypertext Transfer Protocol and Extensible Messaging and Presence Protocol. Finally, the SEP 2.0 standard has not specified non repudiation while OpenADR 2.0 standard has optional non repudiation [38].]. Table 4.13 shows the differences between the SEP 2.0 and OpenADR standards.

Functionalities	OpenADR 2.0	SEP 2.0
Direct Load Control	Limited	Yes
Profile Structure	Tiered profile	Modular profile
Full Reporting Services	Yes	No
Supported Transports	Simple HTTP, XMPP	HTTP (REST-styled)
Non Repudiation	Optional	Not Specified

Table 4.13: *Differences between the SEP 2.0 and OpenADR 2.0 standards [38]*

5 Admission Control

In a heterogeneous environment with many different entities that consist a network, control is critical for efficiency and for providing certain services. Admission Control is used in order to control the admission of subscribers into a network and for meeting certain Quality of Service requirements for all subscribers of the network.

In this chapter we try to provide a thorough description of Admission Control, how it function, under what constrains and where it aims. Also, we provide an understanding of Admission Control as a problem, how its evaluated and what types of solutions can exist.

5.1 Admission Control

Admission Control is an algorithm that controls the admission of a subscriber into a network. A subscriber is admitted into the network if and only if all Quality of Service (*QoS*) constraints can be satisfied and guaranteed without jeopardizing the Quality of Service of existing subscribers in the network [42, 43, 46]. Admission Control is a very effective method for optimal resource management [44].

The control of the admission of subscribers into a network is made using a network usage descriptor that specifies the characteristics and limitations of the network and the Quality of Service requirements. An Admission Control algorithm aims to make efficient use of the network, guarantee Quality of Service and fairness for all subscribers into the network [42, 43].

Various types of network subscribers have different types of Quality of Service demands and different Service Level Agreements (*SLA*). Customers with identical or very similar Quality of Service demands and Service Level Agreements belong to the same Service Class. For each Service Class there is a different Admission Control algorithm that manage the admissions of new subscribers into the network according to the Quality of Service demands of this particular Service Class [43]. Figure 5.1 shows the Admission Control algorithm flowchart.

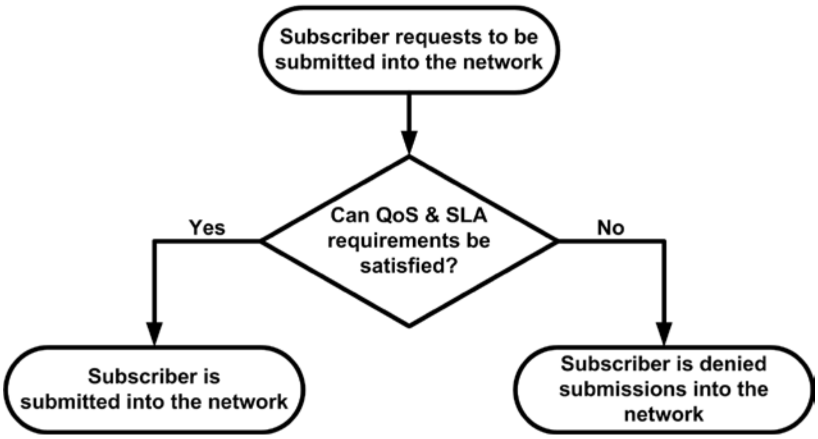

Figure 5.1: *Admission Control algorithm flowchart*

Admission Control schemes control the flow of subscribers with different Service Class into the network with an efficient mechanism to cope with the different Quality of Service requirements. Also, Admission Control schemes are classified by different characteristics that characterize the admission policy under different network parameters. Every Admission Control scheme is evaluated for it's performance by various metrics that assess the provided Quality of Service to the subscribers. Moreover, since Admission Control operates in real time an admission control algorithm should be executed very fast [43].

An Admission Control problem is a multi objective optimization problem that is tailored for providing maximum efficiency and utility to the network while at the same complying to the Quality of Service requirements of the subscribers. The design and implementation of a Admission Control algorithm should be very careful in order to avoid false decisions that jeopardise the Quality of Service of the subscribers [43].

There are two types of Admission Control algorithms, the Global Admission Control and the Local Admission Control. A Global Admission Control has knowledge and considers the complete state of the network for every admission decision. A Local Admission Control has knowledge and considers the state of a small portion of the network for every admission decision. Usually, Global Admission Control algorithms are slower, more adaptable, more optimized and more complex than Local Admission Control algorithms [45].

6 A Demand Response algorithm for grid reliability using Admission Control for a fair scheme

One of the main incentives that lead to the creation of the Smart Grid and Demand Response is to provide grid reliability. By the term grid reliability we mean that the demand for power does not exceeds a certain value. This is done in order to assure that no power outages occur and that the power grid maintain its characteristics in a stable and continues fashion.

Thus, under the Smart Grid and by using Demand Response, a Utility can shape the demand in such a way that the grid remains reliable at all times.

In this chapter we propose a Demand Response algorithm for grid reliability using admission control for a fair scheme.

6.1 The actors

In the Smart Grid, Utilities and Consumers alike play a vital role. The Utility produce power and can issue Demand Response events in order to shape the demand for power. On the other hand, the Consumer consumes power and can accept Demand Response events in order to shape the consumption of power. Between the Utility and the Consumer there is a Demand Response Automation Server. The Demand Response Automation Server is a Virtual End Node in respect to the Utility and a Virtual Top Node in respect to the Consumers. The Demand Response Automation Server provides feedback to the Utility from the Consumers and forwards the Demand Response events from the Utility to the Consumers. Figure 6.1 shows the actors in the proposed algorithm.

Figure 6.1: *The actors in the proposed algorithm*

In the proposed algorithm we assume a Utility, Subscribers and a Demand Response Automation Server. Also, we assume that this is an incentive driven Demand Response program. Thus, the Utility has to provide a payback to all Subscribers that accept and implement a Demand Response event.

6.1.1 Modeling the Subscribers

A Subscriber is consuming energy, sending requests to provide load reduction and accepting Demand Response events. Every Subscriber consumes energy by activating loads. For every Subscriber we assume Standard Loads, Flexible Loads and Elastic Loads. The Standard Loads cannot be subjected to Demand Response events. Flexible Loads and Elastic Loads can be subjected to Demand Response events. We also assume that every Subscriber can provide a certain load reduction without violating the Service Level Agreement. The load reduction by the Subscriber is achieved by shaping the usage of Flexible Loads and Elastic Loads via accepting Demand Response events by the Demand Response Automation Server. The load reduction by the Subscriber is issued in percentages of the forecasted load of the Subscriber.

For the Subscriber S_i , the provided load reduction is:

$$L_{RS_i} = P_{S_i} L_{FS_i}\,, \qquad i = 1, 2, \ldots N$$

with P_{S_i} being the percentage of the load reduction of the Subscriber S_i, L_{FS_i} is the forecasted load of the Subscriber S_i and N is the total number of subscribers.

The Service Level Agreement is a metric that characterize the minimum guaranteed provided service by the Utility to the Subscriber. We assume that the Service Level Agreement is equal to the maximum percentage of load reduction that each Subscriber can provide.

$$SLA_{S_i} = P_{S_i}^{MAX}, \qquad i = 1, 2, \dots N$$

The Quality of Service is a metric that characterize the user satisfaction in respect to the comfort provided by the service. We assume the Quality of Service is a linear function and inverse proportional to the percentage of the load reduction provided by the Subscriber.

$$QoS_{S_i} = \left(\frac{SLA_{S_i} - P_{S_i}}{SLA_{S_i}} \right) 100\%, \qquad i = 1, 2, \dots N$$

Also, every Subscriber has an effective Demand Response time, t_{DRS_i}. The effective Demand Response time is the time that a Demand Response event can be in effect. After that time the Demand Response event stops being in effect. Also, no other Demand Response event can be accepted by a Subscriber that has already implemented one Demand Response event. Moreover, all Demand Response events are considered to be non-preemptive. Thus, once a Demand Response event is in effect it cannot be stopped. The only exception to this is that a Demand Response event can be stopped when the request from the Utility for load reduction is no longer in effect.

In an incentive driven Demand Response program, the utility function dictates the payback to the Subscriber from the Utility for participating in the Demand Response program.

We assume three types of utility functions, a linear and two exponentials functions. The exponential functions are a convex and a concave function.

The utility function is:

$$PB_{S_i} = \int_0^T \left(L_{FS_i} - L_{RS_i}\right)^a , \qquad i = 1, 2, \dots N$$

where $a = 2$ for a convex utility function, $a = 1$ for a linear utility function and $a = 1/2$ for a concave utility function.

The total payback by the Utility to all subscribers is:

$$PB_T = \sum_{i=1}^{N} PB_{S_i} , \qquad i = 1, 2, \dots N$$

6.1.2 Modeling the Utility

A Utility is providing energy for a group of consumers and issues Demand Response events in order to shape the energy consumption. The Utility monitors the energy usage by the Subscribers. In cases of increased load requirements the Utility can issue a request to a Demand Response Automation Server to decrease the load consumed by the Subscribers. The requests by the Utility to the Demand Response Automation Server is issued in percentages of the maximum forecasted energy load.

For the Utility U, the request for load reduction is:

$$R_U = P_{R_U} L_F^{MAX}$$

with P_{R_U} being the requested reduction percentage and L_F^{MAX} is the maximum value of the forecasted load consumed by the subscribers. The forecasted load is the summary of the forecasted load consumed by all subscribers:

$$L_F = \sum_{i=1}^{N} L_{FS_i} , \qquad i = 1, 2, \dots N$$

The total time that the request for load reduction by the Utility is in effect is called Grid Reliability time, t_{GR}:

$$t_{GR} = t_{GR}^{E} - t_{GR}^{S}$$

with t_{GR}^{S} being the time when the forecasted load exceeds the request for load reduction and t_{GR}^{E} being the time after t_{GR}^{S} that the forecasted load is lower than the request for load reduction. In other words, t_{GR} is all t that the total forecasted subscriber load exceeds the request for load reduction, $L_F(t) \geq R_U$.

The total forecasted load production cost by the Utility is a exponential convex function of the forecasted load production.

$$C_{L_F} = \int_{0}^{T} {L_F}^2$$

Thus, the total gain is:

$$G_T = C_{L_F} - C_{L_{DR}} - PB_T$$

with $C_{L_{DR}}$ being the load production cost after the implementation of the Demand Response events.

6.1.3 Modeling the Demand Response Automation Server

The Demand Response Automation Server accepts the requested load reduction from the Utility and dispatch Demand Response events to Subscribers in order to implement the requested load reduction.

The Demand Response events are issued using Admission Control in three different schemes.

Admission control schemes:

- Admissions will be implemented using a "High Reduction First" scheme. The Subscriber that can provide the highest load reduction will be admitted first.

● Admissions will be implemented using a "Low Reduction First" scheme. The Subscriber that can provide the lowest load reduction will be admitted first.

● Admissions will be implemented using a "Random" scheme. The Subscribers will be admitted randomly.

Every Admission Control scheme will be implemented using two different scenarios:

Admission control scenarios:

● "Minimum number of Subscribers" with $P_{S_i} = P_{S_i}^{MAX} = SLA_{S_i}$

When the "minimum number of Subscribers" scenario is implemented, all Demand Response events that will be imposed to Subscribers will be at the maximum percentage allowed by the Service Level Agreement of each Subscriber.

● "Maximum number of Subscribers" with $P_{S_i} < P_{S_i}^{MAX} = SLA_{S_i}$

When the "maximum number of Subscribers" scenario is implemented, all Demand Response events that will be imposed to Subscribers are equal to a given percentage each time. All Subscribers will suffer the same Demand Response event except from the Subscribers that have a lower Service Level Agreement. The Subscribers that have a Service Level Agreement lower than the given percentage will suffer the maximum Demand Response event allowed by their Service Level Agreement.

6.1.4 Other constrains

The total Utility gain must be greater than zero in order for a Demand Response request to be implemented.

$$G_T \geq 0$$

Also, the number of Subscribers that suffer a Demand Response event must be equal or less than the total number of Subscribers.

$$N_{DR} \leq N$$

Finally, for every Subscriber, the percentage of provided load reduction must be equal or less than the Service Level Agreement.

$$P_{S_i} \leq SLA_{S_i}$$

6.1.5 Providing fairness and define the optimum utility function for the Utility

Fairness to the Utility is provided by a utility function that guaranties that the total gain is greater or at least equal to zero, $G_T \geq 0$. Also, the optimum utility function for the Utility is the utility function that provides the maximum gain, thus the minimum payback to the Subscribers.

For total gain we have:

$$G_T \geq 0 =>$$

$$C_{L_F} - C_{L_{DR}} - PB_T \geq 0 =>$$

$$C_{L_F} - C_{L_{DR}} \geq PB_T$$

Let the load reduction provided by every subscriber be x_i . Also, the total load reduction provided by all subscribers is $\sum_{i=1}^{N} x_i$.

Thus,

$$C_{L_F} - C_{L_{DR}} \geq PB_T =>$$

$$\left(\sum_{i=1}^{N} x_i\right)^2 \geq \sum_{i=1}^{N} {x_i}^a$$

where $a = 2$ for a convex utility function, $a = 1$ for a linear utility function and $a = {}^1/_2$ for a concave utility function.

Proof that $(\sum_{i=1}^{N} x_i)^2 \geq \sum_{i=1}^{N} {x_i}^a$ always holds for $x_i > 1$ and $a = 2$ or $a = 1$ or $a = {}^1/_2$:

$$\left(\sum_{i=1}^{N} x_i\right)^2 = \sum_{i=1}^{N} {x_i}^2 + \sum_{i\neq j} x_i x_j > \sum_{i=1}^{N} {x_i}^2 > \sum_{i=1}^{N} {x_i}^1 > \sum_{i=1}^{N} {x_i}^{1/2}$$

Thus, the total gain is always greater than zero and the proposed utility function is always fair to the Utility. Also, the optimum utility function for the Utility is provided by the concave utility function.

6.1.6 Providing fairness and define optimum utility function for the Subscribers

Fairness to the Subscribers is provided by a utility function that provides the higher payback to the subscriber that can provide the higher load reduction. Also, the optimum utility function for the Subscribers is the utility function that provides the maximum payback to the Subscribers.

The utility function is:

$$PB = \left(L_{FS_i} - L_{RS_i}\right)^a$$

where $a = 2$ for a convex utility function, $a = 1$ for a linear utility function and $a = {}^1\!/_2$ for a concave utility function.

Thus, the proposed utility function is always fair to the Subscribers because the subscriber that can provide the higher load reduction will be provided with the higher payback.

Also, since $\left(L_{FS_i} - L_{RS_i}\right)^2 > \left(L_{FS_i} - L_{RS_i}\right)^1 > \left(L_{FS_i} - L_{RS_i}\right)^{1/2}$ for $\left(L_{FS_i} - L_{RS_i}\right) > 1$, the optimum utility function for the Subscribers is the convex utility function.

6.1.7 Providing fairness for the Subscribers and the Utility via the Admission Control scheme

The Admission Control schemes provides a uniform process under which the requests of the Subscribers are admitted. Providing a uniform process under all three admission control schemes provides fairness towards the Subscribers and the Utility.

More specifically, the "High Reduction First" Admission Control scheme admits first the user that can provide the highest load reduction. The second user that will be admitted will provide that second highest load reduction. The "Low Reduction First" Admission Control scheme admits first the user that can provide the lowest load reduction. The second user that will be admitted will provide that second lowest load reduction. Finally, the "Random" Admission Control scheme randomly admits the first user. The second user that will be randomly admitted as well. Thus, via a uniform Admission Control scheme fairness is provided towards the Subscribers and the Utility.

6.1.8 Pre-emptive and non pre-emptive Demand Response events

A pre-emptive Demand Response event is a Demand Response event that once started it can be stopped without reaching the maximum duration that it can last. Also, a pre-emptive Demand Response event can be resumed at a later time after being stopped.

A non pre-emptive Demand Response event is a Demand Response event that once started it cannot be stopped.

As stated earlier, we assume that all Demand Response events are non pre-emptive. The only exception to this is that a Demand Response event can be stopped when the request from the Utility for load reduction is no longer in effect.

6.2 Algorithm flowchart and system model

Figure 6.2 shows the flowchart of the proposed algorithm.

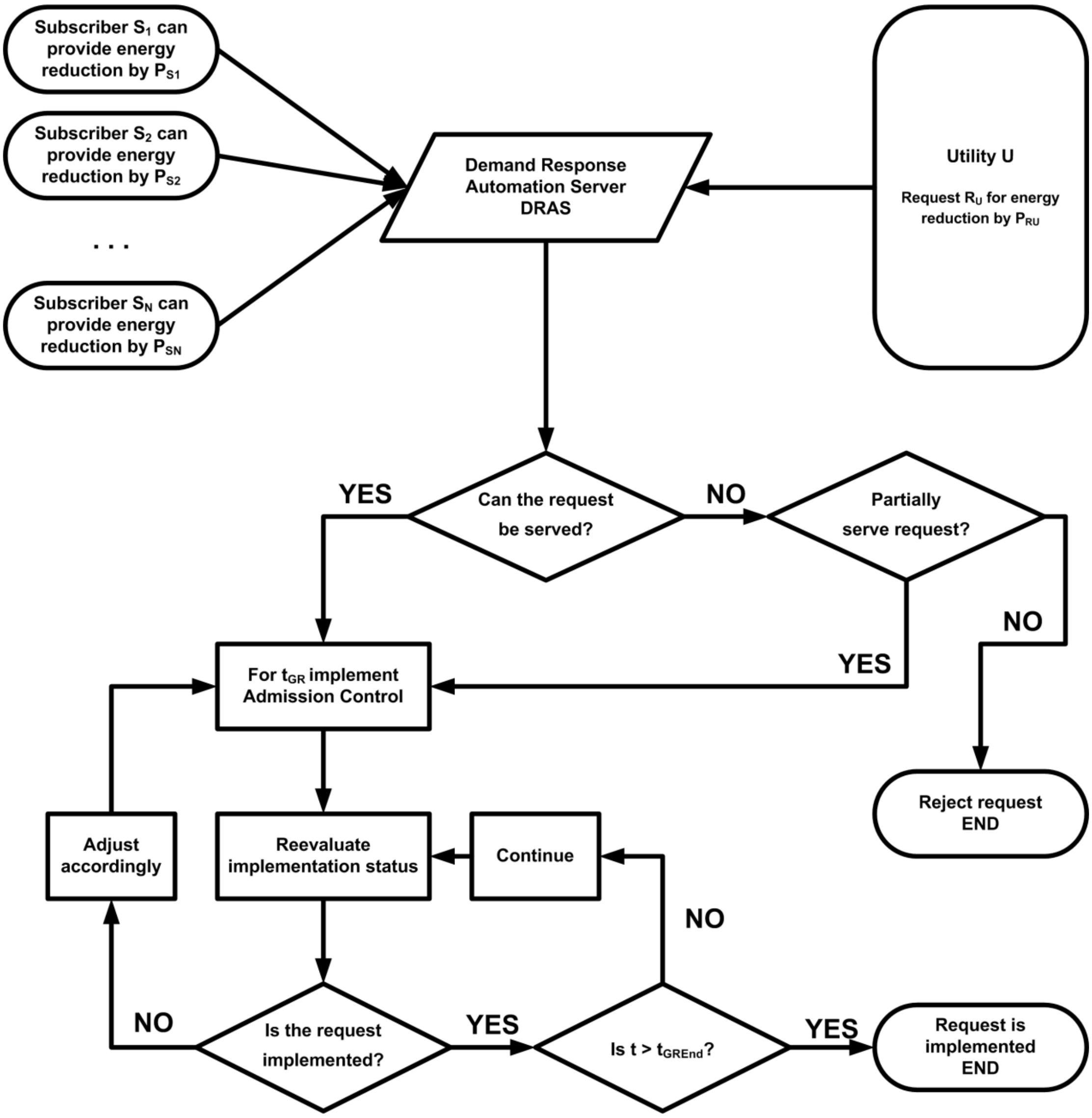

Figure 6.2: *The flowchart of the proposed algorithm*

The Demand Response Automation Server has as input the percentage of energy reduction that every subscriber can provide and the percentage of load reduction requested by the Utility. Also, the Demand Response Automation Server has complete knowledge of the environment of Subscribers.

Thus, the Demand Response Automation Server can calculate the total forecasted load from data provided by individual Subscribers. When the Utility request a load reduction, the Demand Response Automation Server can fairly accurate predict if this request can be served.

If a given request can be served then the Demand Response Automation Server implements Admission Control for the Grid Reliability time. The Demand Response Automation Server admits the Subscriber with the highest priority as dictated by the Admission Control scheme that is in effect.

When the first Subscriber is admitted the Demand Response Automation Server re-evaluate the load usage. If the load usage is higher than the requested load reduction by the Utility, then the Demand Response Automation Server admits the second in priority user as dictated by the Admission Control scheme that is in effect.

This goes on until the load usage is less or equal than the requested load reduction by the Utility. This loop is in effect until the end of the Grid Reliability time. The Demand Response events from the Demand Response Automation Server to the subscribers are effective until the end of the Grid Reliability time or until the end of the effective Demand Response time.

If a given request cannot be fully served then the Demand Response Automation Server can choose to partially implement the request or reject the request. If the Demand Response Automation Server choose to partially implement the request then the Admission Control sequence gets initiated until all users are admitted and no further load reduction can take place. The Demand Response events from the Demand Response Automation Server to the subscribers are effective until the end of the Grid Reliability time or until the end of the effective Demand Response time. Figure 6.3 shows the system model of the proposed algorithm.

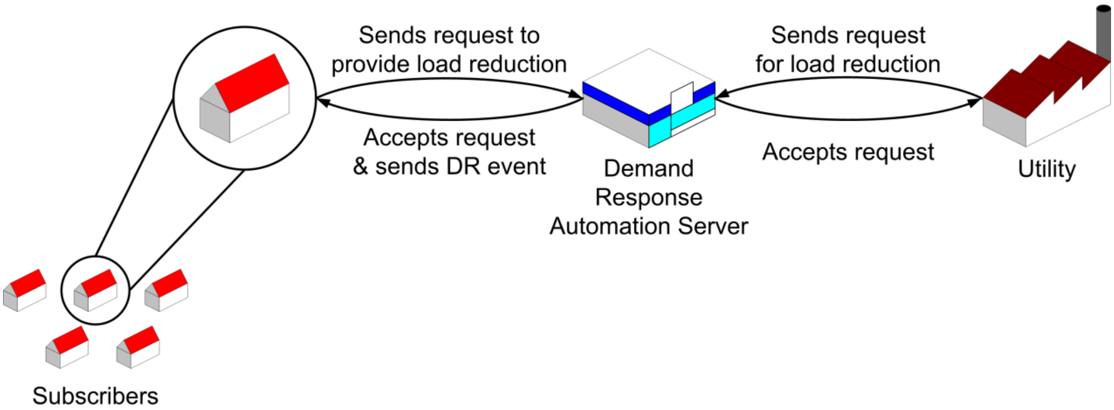

Figure 6.3: *The system model of the proposed algorithm*

6.3 Algorithm pseudocode

Step 01: Set number of Subscribers, N.

Step 02: $\forall t$ define the daily profile (forecasted Subscriber load), $L_{FS_i} = f(t)$, of all Subscribers.

Step 03: Set the t_{DRS_i}, the effective time of a Demand Response event for all Subscribers.

Step 04: Set the SLA_{S_i}, the Service Level Agreement for all Subscribers.

Step 05: $\forall t$ calculate the total forecasted load, $L_F(t) = f(t)$.

Step 06: Set requested reduction percentage, P_{R_U}.

Step 07: Define the request for load reduction, $R_U = P_{R_U} L_F^{MAX}$.

Step 08: $\forall t$ find all t_{GR} that the total forecasted subscriber load exceeds the request for load reduction, $L_F(t) \geq R_U$.

Step 09: $\forall t_{GR}$ implement Demand Response algorithm for grid reliability with admission control for a fair scheme.

Step 09a: For all users, calculate $\sum_{t=t_{GR_S}}^{t_{GR_S}+t_{DRS_i}} L_{RS_i}(t)$

Step 09b: Sort all users according to the Admission Control scheme that is in effect.

Step 09c: for t_{GR}

while $L > R_U$

Admit user with the highest priority according to the Admission Control scheme that is in effect.

Step 10: Calculate the number of used Subscribers to implement the Demand Response request, Quality of Service, and Payback for all subscribers, the Utility Gain and the Demand Response successfulness.

6.4 Complexity of the algorithm

The complexity of an algorithm is the order of growth of the running time of an algorithm and characterize the efficiency of the algorithm. Since high precision in determining the running time is not required, the asymptotic efficiency of the algorithm is used. Also, since the input of the algorithm is large due to the existence of double nested loops, the multiplicative constants and lower order terms are dominated by the effects of the input size of the double nested loops. Also, we have only an asymptotic upper bound, so we will use the O-notation to describe the complexity of the proposed algorithm [47]. Thus, we conclude that, the complexity of the proposed algorithm is $O(n^2)$.

6.5 Implementation of the proposed algorithm with the OpenADR 2.0 Demand Response standard

The proposed algorithm can be implemented by utilizing the OpenADR Demand Response standard. All required technology for a successful implementation is available and already proven under various commercial implementations. More specifically, a Smart House is equipped with a main control panel that is capable of controlling loads and sending meter data and device status to an advanced telemetry and meter data device server. Finally, a Demand Response Automation Server can log data, implement a Demand Response algorithm and accept input from an Operator.

Moreover, the OpenADR 2.0 B profile provide all required functions for a successful implementation of the proposed algorithm. More specifically, the OpenADR 2.0 B profile fully supports feedback, the Demand Response events can target a group or class of appliances, the priority of the Demand Response events can be set, the status of a Demand Response event can be determined and the Demand Response events can be scheduled.

The implementation of the proposed algorithm with the OpenADR standard is very similar to the small commercial case as described in the subchapter 4.1.8. of this work.

7 Simulation and results

In the previous chapter we propose a Demand Response algorithm for grid reliability using admission control for a fair scheme. In this chapter we will showcase the simulation of the proposed algorithm.

More specifically, we will go through the code that we write in order to simulate the proposed algorithm. Also, we will thoroughly analyse how all actors where simulated. Finally, we will demonstrate how all metrics that will be used to evaluate the proposed algorithm were extracted.

We focus on the residential case and the results are based on a comparative analysis that do not influence the general findings of the results.

7.1 Simulating the Subscribers

In order to create a complete environment of subscribers we need to start with a realistic profile for the daily load consumed by one Subscriber. Then, we implement key changes upon the daily load consumed by one Subscriber that will result in the creation of a diverse and realistic environment of Subscribers.

A realistic profile for the daily load consumed by one subscriber was provided by [46]. Figure 7.1 shows the daily load consumed by one Subscriber.

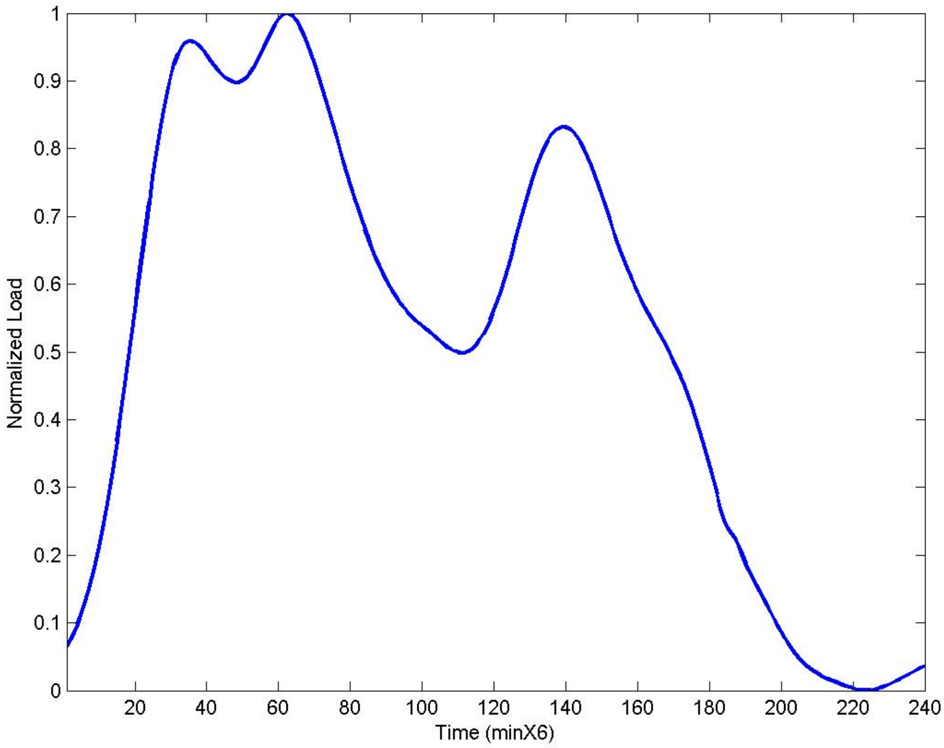

Figure 7.1: *The daily load consumed by one Subscriber*

The first alteration that we will implement is to multiply the normalized load with 1000. By doing so we avoid the changes in the behaviour of the proposed utility functions around 1. Thus, the daily load will be from 0 to 1000 Watts. Code snippet 7.1 shows the importing of the daily load profile form the source file and multiplying by 1000.

```
%Import load profile from source
L_temp = importdata('load_profile_source.mat');
L_temp = 1000*L_temp';
```

Code snippet 7.1: *Importing the daily load profile form the source file and multiply by 1000*

Also, we set that the number of subscribers equal to 1000. Code snippet 7.2 shows the setting of the number of Subscribers equal to 1000.

```
%Set number of subscribers
N=1000;
```

Code snippet 7.2: *Setting the number of Subscribers equal to 1000*

The second alteration that we will implement is to offset the initiation of the daily load with a random normal distributed time offset factor. The random normal distributed time offset factor has a minimum equal to 0 and maximum equal to 10 hours. By doing so we can simulate the randomness of an environment of Subscribers as far as the initiation of the daily activity by a single Subscriber is concerned. Code snippet 7.3 shows the implementation of random normal distributed time offset.

```
%Set minimum and maximum time offset - (10=1 hour)
t_Offset_min=0;
t_Offset_max=100;

%Create matrix with daily load profile of all subscribers
L=zeros(240,N);
for i=1:1:240
    for j=1:1:N
        L(i,j)=L_temp(i);
    end
end

%Extend load day before & day after
for j=1:1:N
    for i=1:1:240
        L(i,j)=L(i,j);
    end
    for i=240:1:480
        L(i,j)=L(i-239,j);
    end
    for i=480:1:720
        L(i,j)=L(i-239,j);
    end
end

%Extend time day before & day after
t=zeros(720,N);
for j=1:1:N
    %Implement time offset
    t_Offset=normrnd(t_Offset_min,t_Offset_max);
    t_Offset = round(t_Offset);
    t(240,j)=t_Offset;
    for i=1:1:480
        t(240+i,j)=t(240+i-1,j)+6;
    end
    for i=1:1:239
        t(240-i,j)=t(240-i+1,j)-6;
    end
end

%Trim load for one day only
for j=1:1:N
    for i=1:1:720
        if ((t(i,j)<0)||( t(i,j)> 1440));
            L(i,j)=0;
        end
    end
end

%Find place before first non zero load
L_Z_start=zeros(1,N);
```

```
for j=1:1:N
    for i=1:1:360
        if ((L(i,j)<=0));
            L_Z_start(1,j)=L_Z_start(1,j)+1;
        end
    end
end

%Find last non zero load
L_Z_end=zeros(1,N);
for j=1:1:N
    L_Z_end(1,j)=L_Z_start(1,j)+240;
end

%Adjust beginning of load
for j=1:1:N
    L_Z_start(1,j)=L_Z_start(1,j)+1;
end

%Copy load to the beginning of matrix
for j=1:1:N
   for i=1:1:720-L_Z_start(1,j)
        L(i,j)=L(i+L_Z_start(1,j)-1,j);
   end
end

%Trim matrix for one day only
for i=720:-1:241
        L(i,:)=[];
end
```

Code snippet 7.3: *Implementation of the random normal distributed time offset*

The third and final alteration that we will implement is to multiply the daily load with a random uniform distributed amplitude factor. The random uniform distributed amplitude factor has a minimum equal to 0.5 and maximum equal to 1.5. By doing so we can simulate the randomness of an environment of Subscribers as far as the load consumed by a single Subscriber is concerned. Code snippet 7.4 shows the implementation of random normal distributed amplitude factor.

```
%Set minimum and maximum amplitude factor (X10)
A_F_min=50;
A_F_max=150;

%Apply amplitude factor
for j=1:1:N
    A_F=(randi([A_F_min,A_F_max]))/100;
    for i=1:1:240
        L(i,j)=L(i,j)*A_F;
    end
end
```

Code snippet 7.4: *Implementation of the random normal distributed amplitude factor*

For every subscriber a different amplitude factor and time offset factor is assigned. Thus, the environment of the 1000 Subscribers that we create in order to test upon the proposed algorithm can be considered as a accurate representation of a real environment of Subscribers. The diversity of the generated subscribers can be depicted on the two pictures that follow. Figure 7.2 shows the daily load of four Subscribers and figure 7.3 shows the daily load of twenty Subscribers.

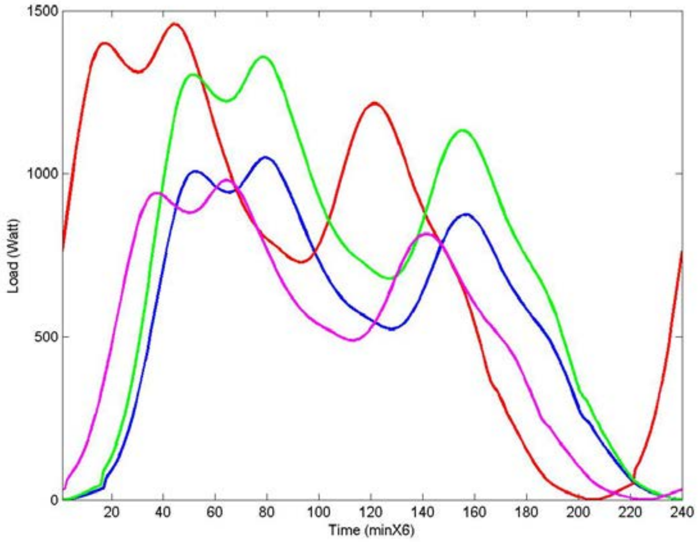

Figure 7.2: *The daily load of four subscribers*

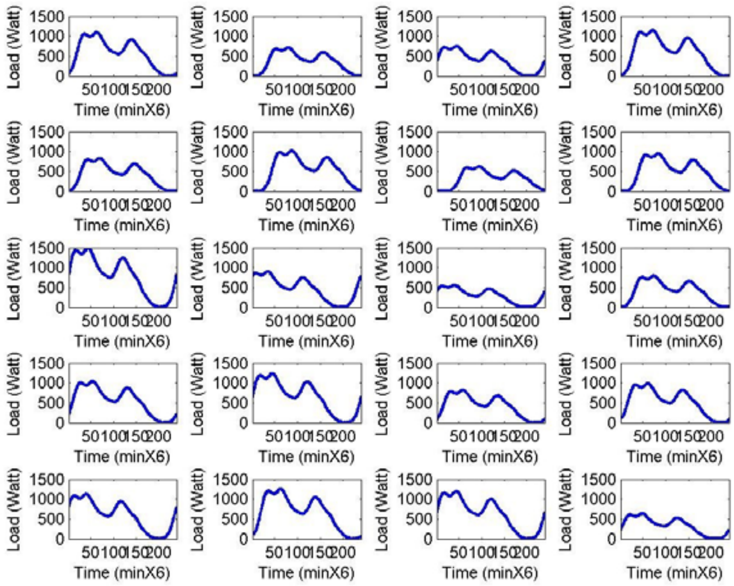

Figure 7.3: *The daily load of twenty subscribers*

Finally, every Subscriber is characterized by a Service Level Agreement and an effective Demand Response time as described in the previous chapter. Thus, we assign to every Subscriber a random uniform distributed Service Level Agreement and a random uniform distributed effective Demand Response time. The random uniform distributed Service Level Agreement has a minimum equal to 5% and maximum equal to 50%. The random uniform distributed effective Demand Response time has a minimum equal to 1 hour and maximum equal to 5 hours. Code snippet 7.5 shows the setting of the random normal distributed Service Level Agreement and effective Demand Response time.

```
%Set minimum and maximum Service Level Agreement
SLA_min=5;
SLA_max=50;

%Set minimum and maximum effective Demand Response time - (10=1 hour)
t_DR_min=10;
t_DR_max=50;

%SLA for each subscriber
SLA=zeros(N,1);
for i=1:1:N
    SLA(i,1)=round(randi([SLA_min,SLA_max]));
end

%Effective DR time for each subscriber
t_DR=zeros(N,1);
for i=1:1:N
    t_DR(i,1)=round(randi([t_DR_min,t_DR_max]));
end
```

Code snippet 7.5: *Setting the random normal distributed Service Level Agreement and effective Demand Response time*

7.2 Simulating the Utility

The Utility knows a forecast of the total load consumed by all subscribers as well as the requested reduction percentage. Thus, the request for load reduction, the Grid Reliability time and the total forecasted load production cost before can be defined. Code snippet 7.6 shows the defining of the total load forecast of all subscribers. Code snippet 7.7 shows the defining of the requested reduction percentage and the request for load reduction. Code snippet 7.8 shows the defining of the total forecasted load production cost. Figure 7.4 shows the total forecasted load of all Subscribers and figure 7.5 shows the total forecasted load of all subscribers with Grid Reliability time and the request for load reduction.

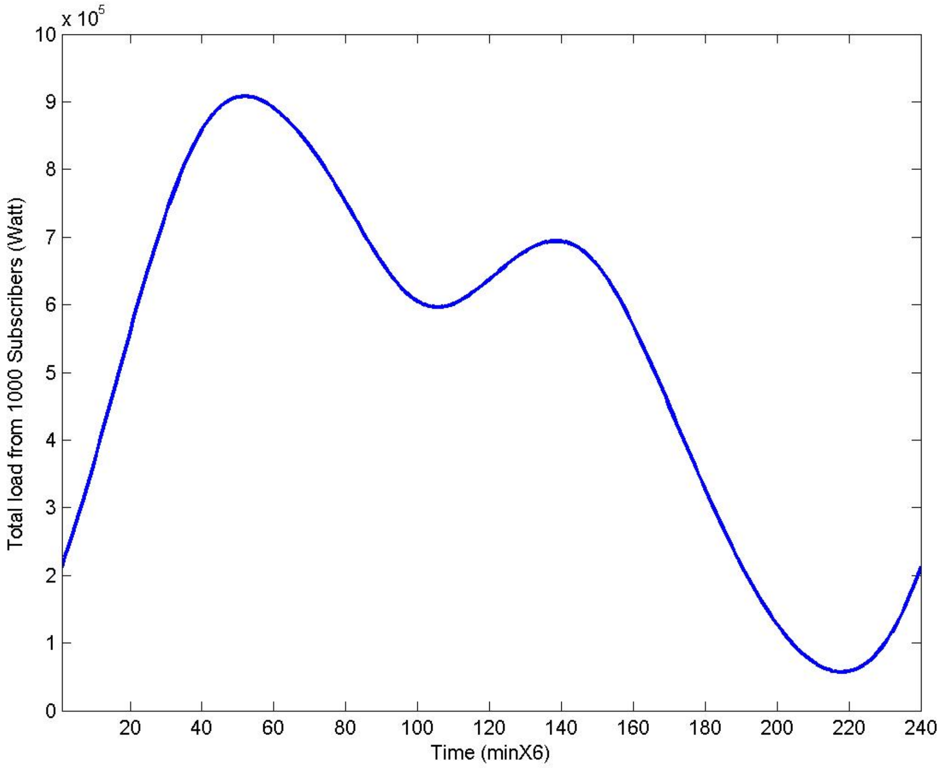

Figure 7.4: *The total forecasted load of all subscribers*

```
%Total load
L_total=zeros(240,1);
for i=1:1:240
    for j=1:1:N
        L_total(i,1)=L_total(i,1)+L(i,j);
   end
end
```

Code snippet 7.6: *Defining the total load forecast of all subscribers*

```
%Requested reduction percentage
P_GR = 80;

%Request for load reduction
L_GR_initiate = (P_GR/100)*max(L_total);
```

Code snippet 7.7: *Defining the requested reduction percentage and the request for load reduction*

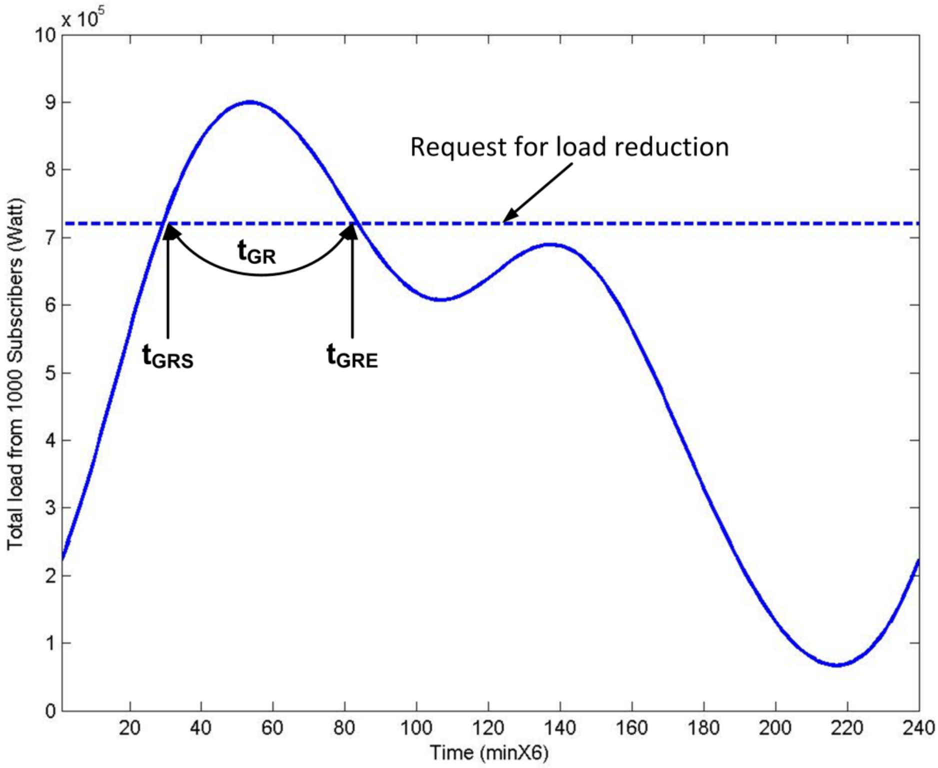

Figure 7.5: *The total forecasted load of all subscribers with Grid Reliability time and the request for load reduction*

```
%Cost before DR
Cost_before_DR=zeros(240,1);
for i=1:1:240
   Cost_before_DR(i,1) = (L_total(i,1))^2;
end

Cost_before_DR_total=0;
for i=1:1:240
    Cost_before_DR_total=Cost_before_DR_total+Cost_before_DR(i,1);
end
```

Code snippet 7.8: *Defining the total forecasted load production cost*

7.3 Simulating the Demand Response Automation Server

The Demand Response Automation Server must accept as input all data provided by the Subscribers and the Utility and implement the Demand Response algorithm.

The following code snippet handles the pre-processing of the data of the load of the Subscribers, the Admission Control logic for all three Admission Control schemes and the implementation of the Admission Control algorithm. Code snippet 7.9 shows the pre-processing of data, Admission Control logic and Admission Control algorithm implementation.

```
%Grid reliability load
L_GR=L_total;
for i=1:1:240
    if (L_total(i,1)<L_GR_initiate);
        L_GR(i,1)=0;
    end
end

%Trim subscriber load for grid reliability load
L_trim_GR_load=L;
for j=1:1:N_max
    for i=1:1:240
        if (L_total(i,1)<L_GR_initiate);
            L_trim_GR_load(i,j)=0;
        end
    end
end

%Trim subscriber load for effective Demand Response time
t_DR=t_DR';
L_trim_t_DR=L_trim_GR_load;
non_zero_element_first_L_trim_t_DR=find(L_GR, 1, 'first');
for j=1:1:N_max
    for i=non_zero_element_first_L_trim_t_DR+t_DR(j):1:240
        L_trim_t_DR(i,j)=0;
    end
end

%Calculate for each subscriber the load reduction
L_reduction=zeros(1,N_max);
for j=1:1:N_max
    for i=1:1:240
        L_reduction(1,j)=L_reduction(1,j)+L_trim_t_DR(i,j);
    end
end

SLA=SLA';

for j=1:1:N_max
    L_reduction(1,j)=L_reduction(1,j)*(SLA(1,j)/100);
end

%Admission control logic
Admission_control=zeros(2,N_max);
```

```
Admission_control(1,:)=L_reduction(1,:);
Admission_control(2,1)=1;

for i=2:1:N_max
    Admission_control(2,i)=Admission_control(2,i-1)+1;
end

Admission_control=Admission_control';
Admission_control = sortrows(Admission_control,1);
%Comment out next line to implement Low Reduction First scheme
Admission_control = flipud(Admission_control);
Admission_control=Admission_control';

%Uncomment next line to implement Random Admission Control scheme
%Admission_control(2,:)=randperm(1000,1000);

%Demand Response Algorithm event sequence
L_total_DR=L_total';
L_DR=L;
DR_map=zeros(240,N_max);

for j=1:1:N_AC
get_load_sequence=Admission_control(2,j);
    for i=1:1:240
        if ((L_total_DR(1,i)<L_GR_initiate))
            L_DR(i,get_load_sequence)=0;
        end
    end
    non_zero_element_first_L_DR=find(L_DR(:,get_load_sequence), 1, 'first');
    for
i=non_zero_element_first_L_DR:1:non_zero_element_first_L_DR+t_DR(1,get_load_sequence)
        if ((L_total_DR(1,i)>L_GR_initiate))
            L_total_DR(1,i)=L_total_DR(1,i)-
((L(i,get_load_sequence))*SLA(1,get_load_sequence)/100);
            DR_map(i,get_load_sequence)=1;
        end
    end
end
```

Code snippet 7.9: *Pre-processing of data, Admission Control logic and Admission Control algorithm implementation*

Also, in order to implement the various Admission Control scenarios the following code was written. Code snippet 7.10 shows the implementation of the alternative Admission Control scenarios. Figure 7.6 shows an implementation of the proposed Demand Response algorithm. Figure 7.7 and figure 7.8 shows a partial implementation of the proposed Demand Response algorithm. Figure 7.9 shows the forecasted load cost, the cost after Demand Response and the gain from Demand Response. Figure 7.10 and figure 7.11 shows the forecasted load of a Subscriber and the load after the Demand Response event. Figure 7.12 and figure 7.13 shows the payback towards a Subscriber.

```
%Alter SLA data for alternative scenarios - Obey original SLA
SLA_new=10;
for i=1:1:N_max
    if (SLA(i,1)<SLA_new)
        SLA(i,1)=SLA(i,1);
    end
    if (SLA(i,1)>SLA_new)
        SLA(i,1)=SLA_new;
    end
end
```
Code snippet 7.10: *Implementation of the alternative Admission Control scenarios*

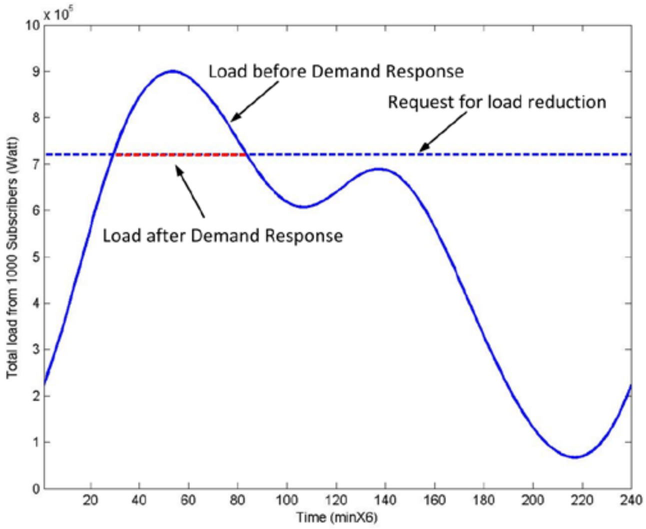

Figure 7.6: *An implementation of the proposed Demand Response algorithm*

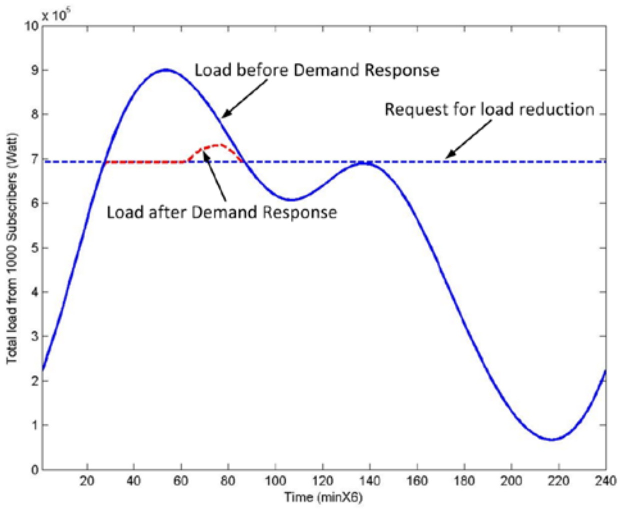

Figure 7.7: *A partial implementation of the proposed Demand Response algorithm (1)*

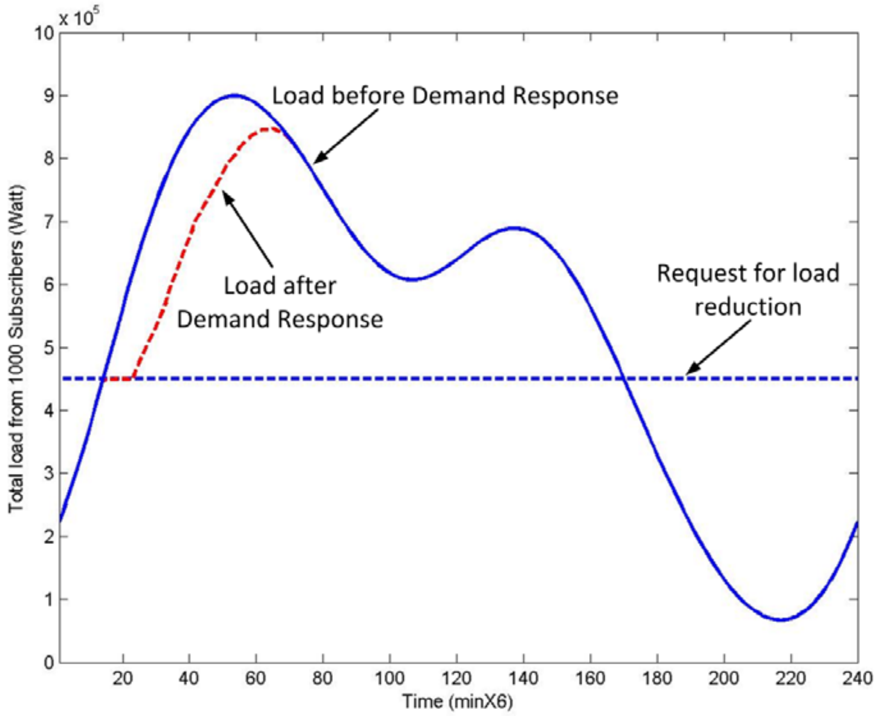

Figure 7.8: *A partial implementation of the proposed Demand Response algorithm (2)*

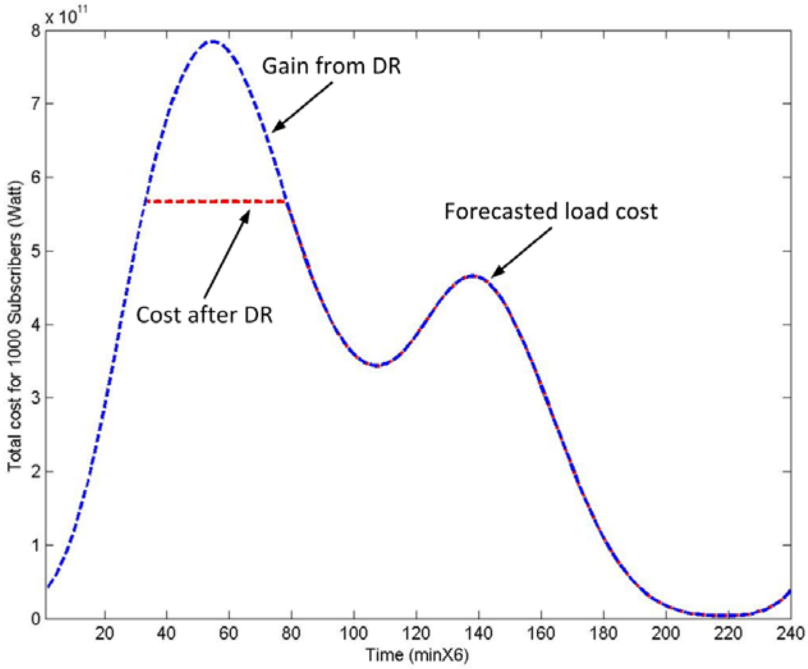

Figure 7.9: *The forecasted load cost, the cost after Demand Response and the gain from Demand Response*

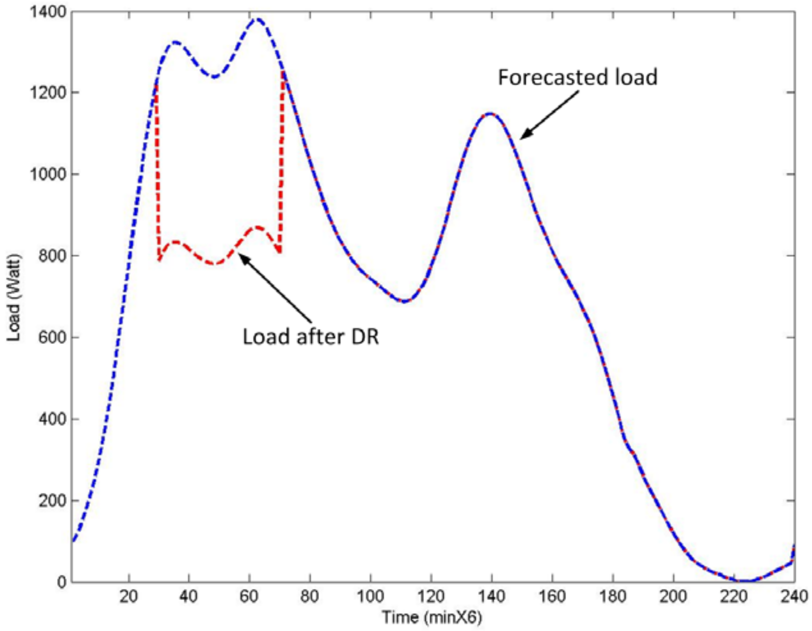

Figure 7.10: *The forecasted load of a Subscriber and the load after the Demand Response event (1)*

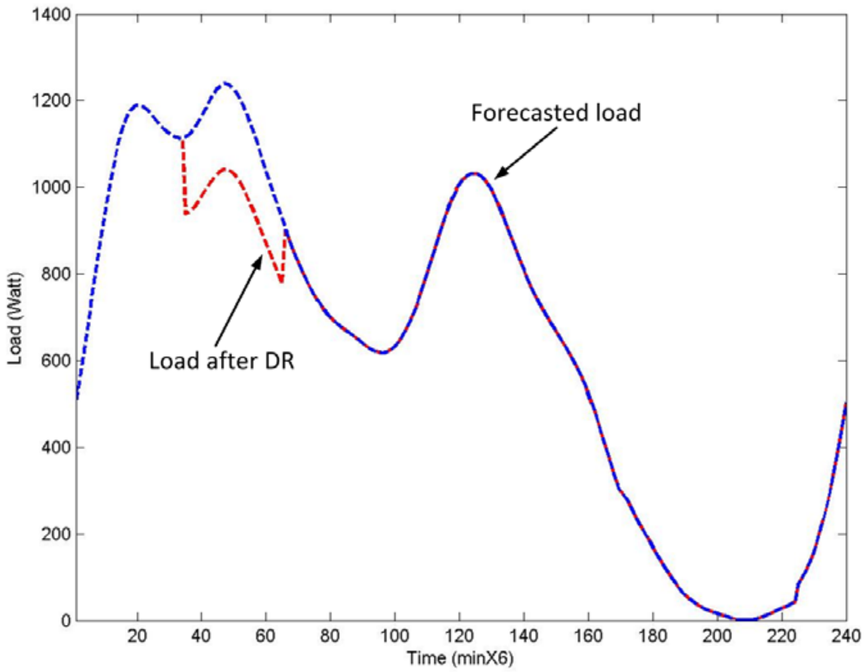

Figure 7.11: *The forecasted load of a Subscriber and the load after the Demand Response event (2)*

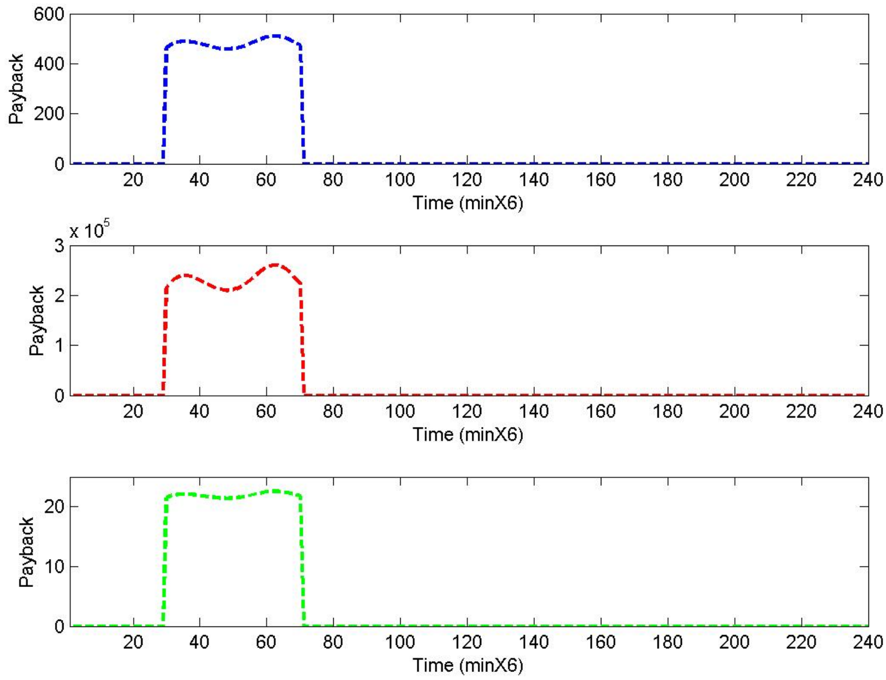

Figure 7.12: *Payback towards a Subscriber (1)*

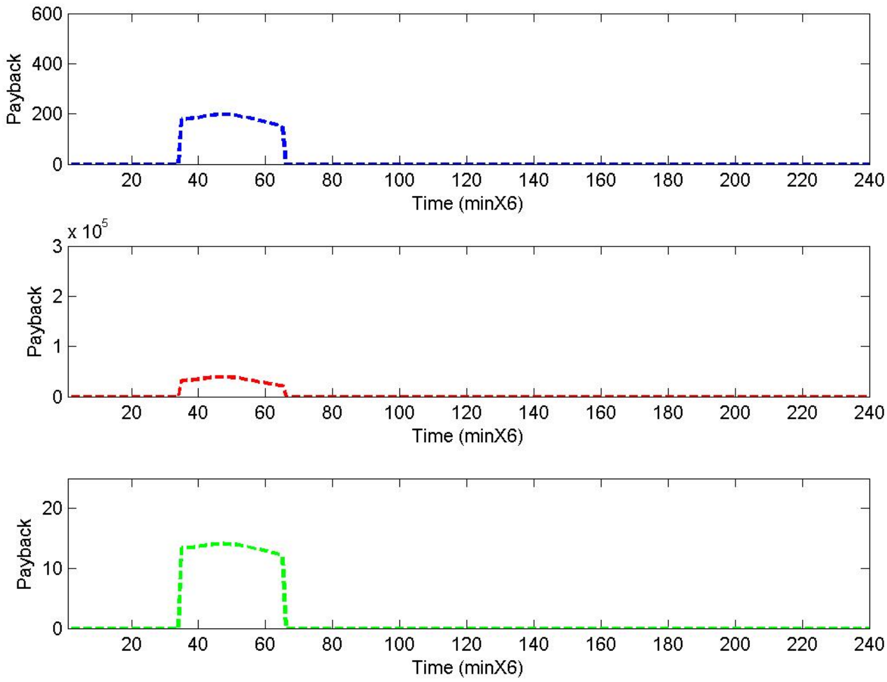

Figure 7.13: *Payback towards a Subscriber (2)*

7.4 Results

In order to conclude upon the effectiveness of the proposed algorithm and compare the three different Admission Control schemes and two different Admission control strategies the following metrics where calculated:

- Number of used Subscribers in order to implement the Demand Response request.

- Average Quality of Service of the environment of Subscribers.

- Average Quality of Service of the environment of Subscribers that suffer a Demand Response event.

- Total Payback from the Utility to the Subscribers that suffer a Demand Response event.

- Average Payback from the Utility to the Subscribers that suffer a Demand Response event.

- Utility gain.

- Demand Response successfulness.

Code snippet 7.11 shows the defining of the number of subscribers that suffer a Demand Response event. Code snippet 7.12 shows the defining of the average Quality of Service of all subscribers and the average Quality of Service of all subscribers that suffer a Demand Response event. Code snippet 7.13 shows the defining of the total and average Payback from the Utility to the Subscribers. Code snippet 7.14 shows the defining of the Utility gain. Code snippet 7.15 shows the defining of the Demand Response successfulness.

```
%Number of subscribers that suffer a DR event
Subscriber_flag=zeros(1,N_max);
for j=1:1:N_max
    for i=1:1:240
        if (DR_map(i,j)==1)
```

```
            Subscriber_flag(1,j)=1;
        end
    end
end

Used_subscribers=0;
for i=1:1:N_max
    Used_subscribers=Used_subscribers+Subscriber_flag(1,i);
end
```

Code snippet 7.11: *Defining the number of subscribers that suffer a Demand Response event*

```
%Quality of service
SLA_original=SLA_original';
QoS=zeros(1,N_max);
for j=1:1:N_max
    if (Subscriber_flag(1,j)==0)
        QoS(1,j)=100;
    end
    if (Subscriber_flag(1,j)==1)
        QoS(1,j)=((SLA_original(1,j)-SLA(1,j))/(SLA_original(1,j)))*100;
    end
end

QoS_average=0;
for j=1:1:N_max
    QoS_average=QoS_average+QoS(1,j);
end
QoS_average=QoS_average/N_max;

QoS_average_DR=0;
for j=1:1:N_max
    QoS_average_DR=QoS_average_DR+(QoS(1,j)*Subscriber_flag(1,j));
end
QoS_average_DR=QoS_average_DR/Used_subscribers;
```

Code snippet 7.12: *Defining the average Quality of Service of all subscribers and the average Quality of Service of all subscribers that suffer a Demand Response event*

```
%Payback base value
PB_Base_Value=1;

%Linear Payback
PB_Linear=zeros(240,N_max);
for j=1:1:N_max
    for i=1:1:240
    PB_Linear(i,j)=DR_map(i,j)*PB_Base_Value*((L(i,j)-(L(i,j)*(SLA(1,j)/100)))^(1));
    end
end

PB_Linear_total_per_subscriber=zeros(1,N_max);
for j=1:1:N_max
    for i=1:1:240
PB_Linear_total_per_subscriber(1,j)=PB_Linear_total_per_subscriber(1,j)+PB_Linear(i,j)
;
    end
end

PB_Linear_total=0;
for j=1:1:N_max
    PB_Linear_total=PB_Linear_total+PB_Linear_total_per_subscriber(1,j);
end

%Convex Payback
PB_Convex=zeros(240,N_max);
for j=1:1:N_max
    for i=1:1:240
    PB_Convex(i,j)=DR_map(i,j)*PB_Base_Value*((L(i,j)-(L(i,j)*(SLA(1,j)/100)))^(2));
    end
```

```
end

PB_Convex_total_per_subscriber=zeros(1,N_max);
for j=1:1:N_max
    for i=1:1:240
PB_Convex_total_per_subscriber(1,j)=PB_Convex_total_per_subscriber(1,j)+PB_Convex(i,j)
;
    end
end

PB_Convex_total=0;
for j=1:1:N_max
    PB_Convex_total=PB_Convex_total+PB_Convex_total_per_subscriber(1,j);
end

%Concave Payback
PB_Concave=zeros(240,N_max);
for j=1:1:N_max
    for i=1:1:240
    PB_Concave(i,j)=DR_map(i,j)*PB_Base_Value*((L(i,j)-
(L(i,j)*(SLA(1,j)/100)))^(1/2));
    end
end

PB_Concave_total_per_subscriber=zeros(1,N_max);
for j=1:1:N_max
    for i=1:1:240
PB_Concave_total_per_subscriber(1,j)=PB_Concave_total_per_subscriber(1,j)+PB_Concave(i
,j);
    end
end

PB_Concave_total=0;
for j=1:1:N_max
    PB_Concave_total=PB_Concave_total+PB_Concave_total_per_subscriber(1,j);
end
```

Code snippet 7.13: *Defining the total and average Payback from the Utility to the Subscribers*

```
%Cost after DR
L_total_DR=L_total_DR';
Cost_after_DR=zeros(240,1);
for i=1:1:240
   Cost_after_DR(i,1) = (L_total_DR(i,1))^2;
end

Cost_after_DR_total=0;
for i=1:1:240
    Cost_after_DR_total=Cost_after_DR_total+Cost_after_DR(i,1);
end

%Gain from DR
Gain_From_DR = Cost_before_DR_total - Cost_after_DR_total;

Gain_Linear = Gain_From_DR - PB_Linear_total;
Gain_Convex = Gain_From_DR - PB_Convex_total;
Gain_Concave = Gain_From_DR - PB_Concave_total;
```

Code snippet 7.14: *Defining the Utility gain*

```
Reliability_flag=1;
for i=1:1:240
   if (L_total_DR(i,1)>L_GR_initiate)
       Reliability_flag=0;
   end
end
```

Code snippet 7.15: *Defining the Demand Response successfulness*

Also, in order to provide a statistical result, 100 batches of 1000 Subscribers per batch were created. Upon each batch, all three Admission Control schemes were implemented for requested reduction percentages from 1% to 50% with 1% steps. Also, upon each batch and for every Admission Control scheme, both Admission Control scenarios were implemented. More specifically, the "maximum number of Subscribers" Admission Control scenario was implemented for percentages of load reduction from 5% to 50% with 5% steps.

7.4.1 "High Reduction First" - Number of used subscribers results

Figure 7.14 shows the number of used Subscribers for the minimum number of Subscribers scenario and figure 7.15 shows the number of used Subscribers for the maximum number of Subscribers scenario.

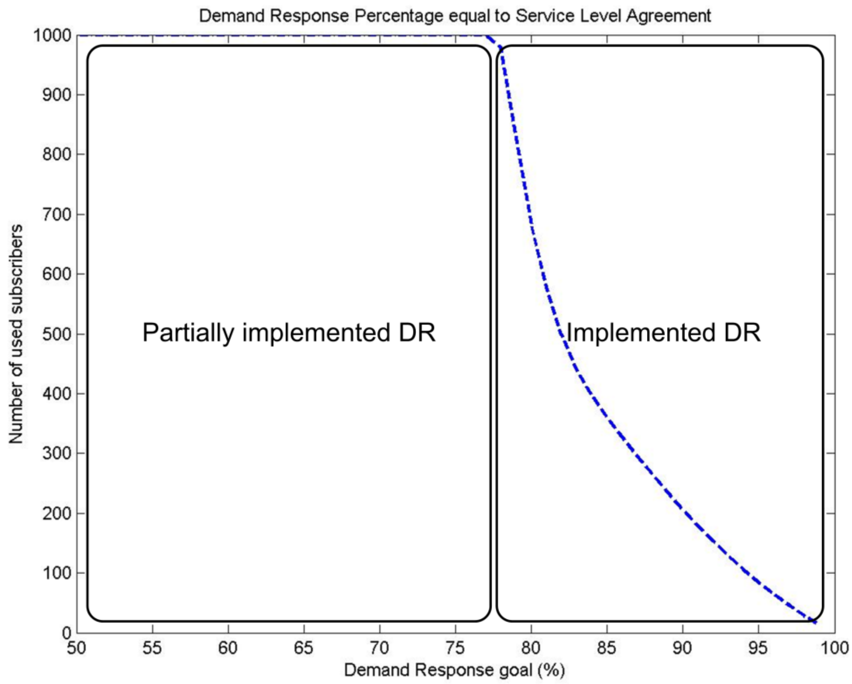

Figure 7.14: *Number of used Subscribers for the minimum number of Subscribers scenario*

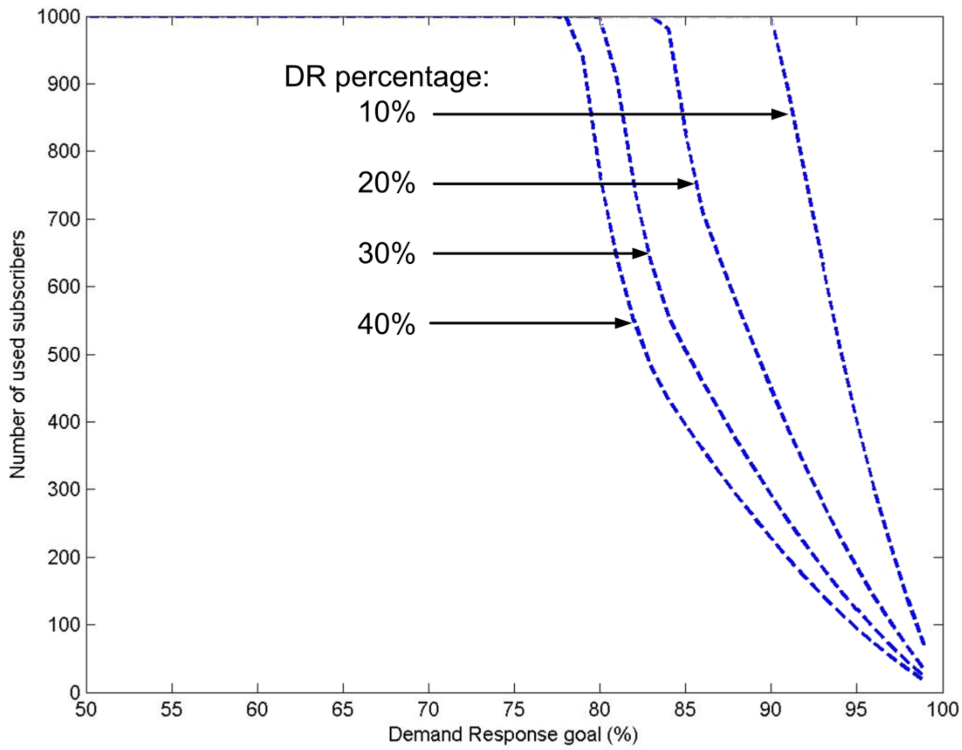

Figure 7.15: *Number of used Subscribers for the maximum numbers of Subscribers scenario*

7.4.2 "High Reduction First" - Quality of Service results

For the following two figures, the blue line is for the average Quality of Service of the environment of Subscribers and the red line is for the average Quality of Service of the environment of Subscribers that suffer a Demand Response event. Figure 7.16 shows the Quality of Service for the minimum number of Subscribers scenario and figure 7.17 shows the Quality of Service for the maximum number of Subscribers scenario.

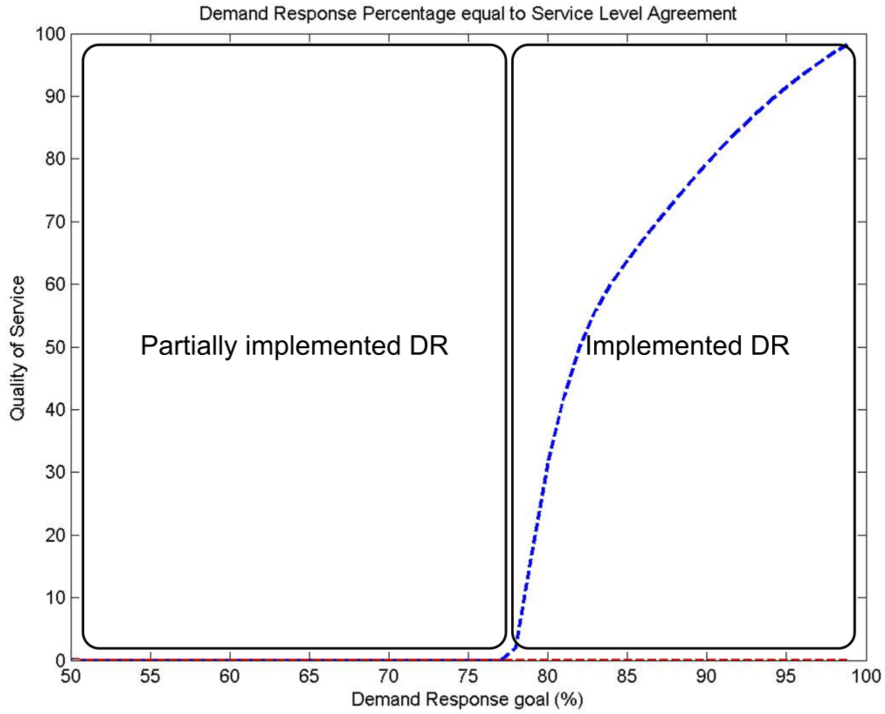

Figure 7.16: *Quality of Service for the minimum numbers of Subscribers scenario*

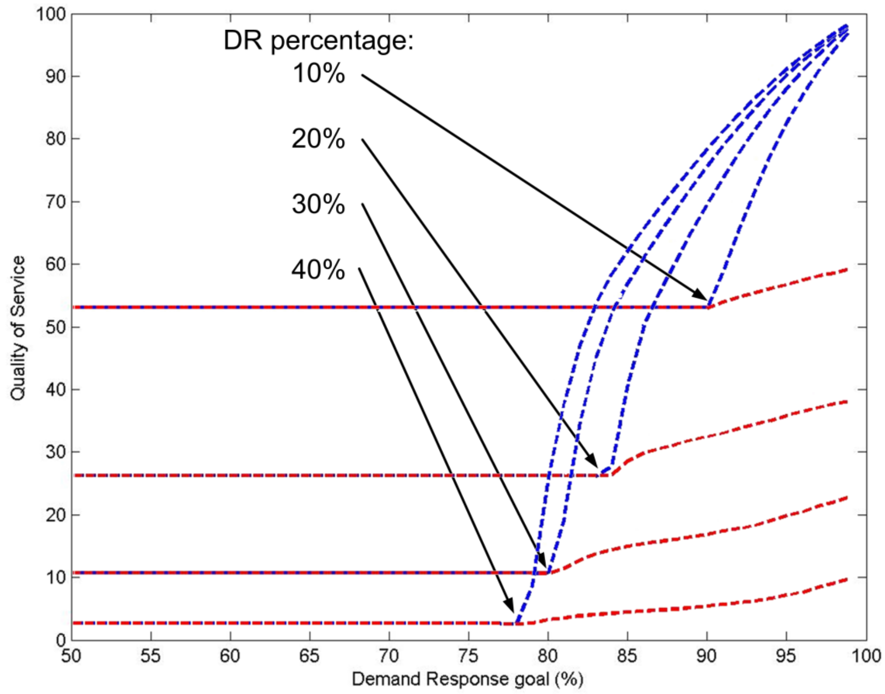

Figure 7.17: *Quality of Service for the maximum numbers of Subscribers scenario*

7.4.3 "High Reduction First" - Total payback results

For the following two figures, the blue line is for the total payback resulting from the linear utility function, the red line is for the total payback resulting from the convex utility function and the green line is for the total payback resulting from the concave utility function. Figure 7.18 shows the total payback for the minimum number of Subscribers scenario and figure 7.19 shows the total payback for the maximum number of Subscribers scenario.

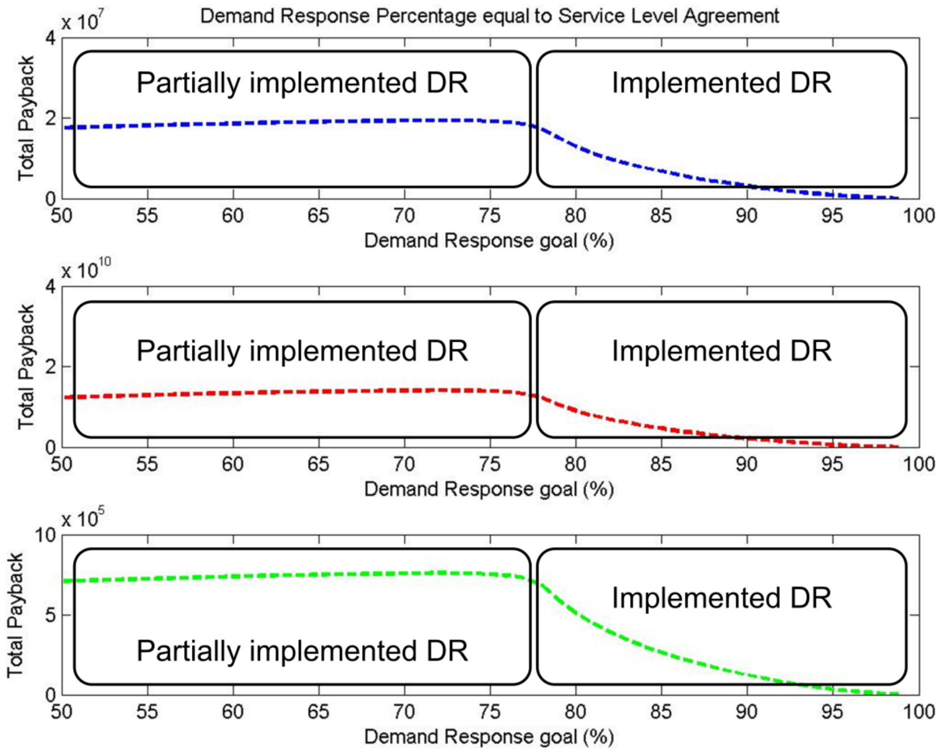

Figure 7.18: *Total payback for the minimum numbers of Subscribers scenario*

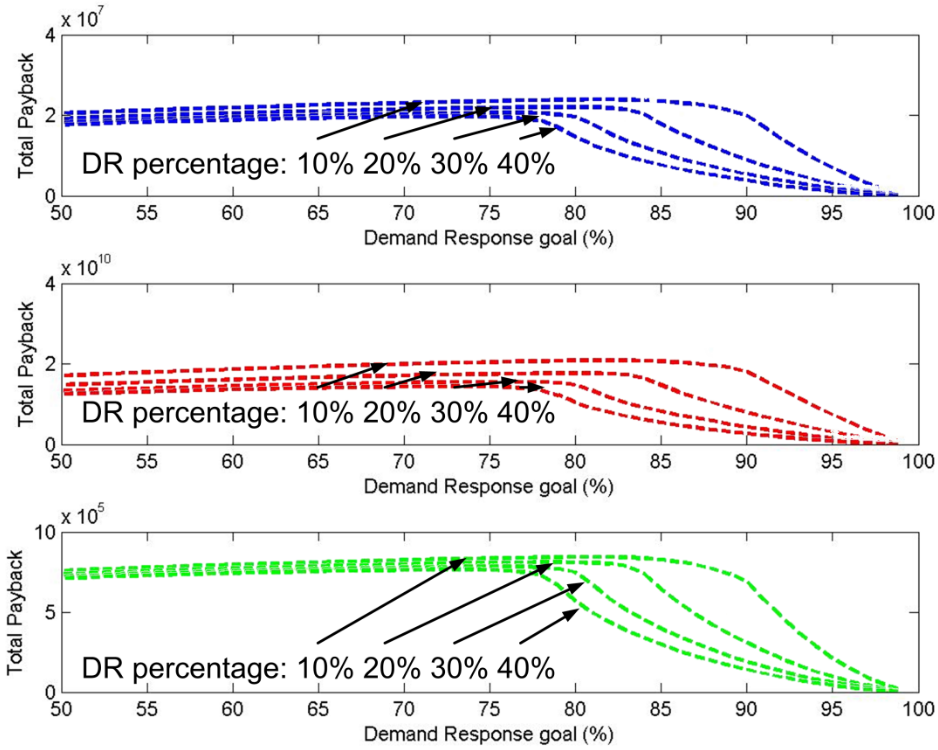

Figure 7.19: *Total payback for the maximum numbers of Subscribers scenario*

7.4.4 "High Reduction First" - Average payback results

For the following the figures, the blue line is for the total payback resulting from the linear utility function, the red line is for the total payback resulting from the convex utility function and the green line is for the total payback resulting from the concave utility function. Figure 7.20 shows the average payback for the minimum number of Subscribers scenario and figure 7.21 shows the average payback for the maximum number of Subscribers scenario.

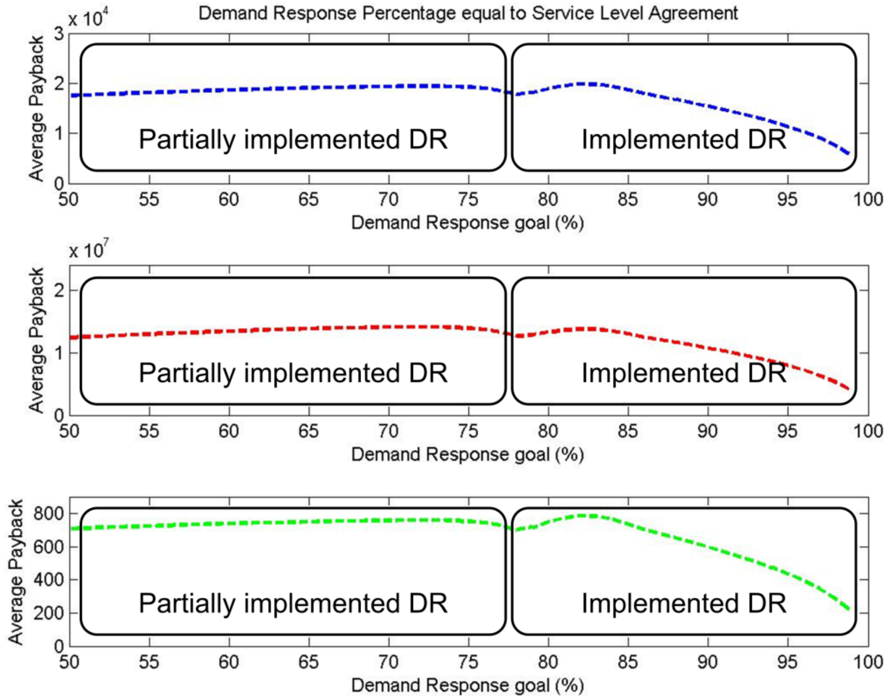

Figure 7.20: *Average payback for the minimum numbers of Subscribers scenario*

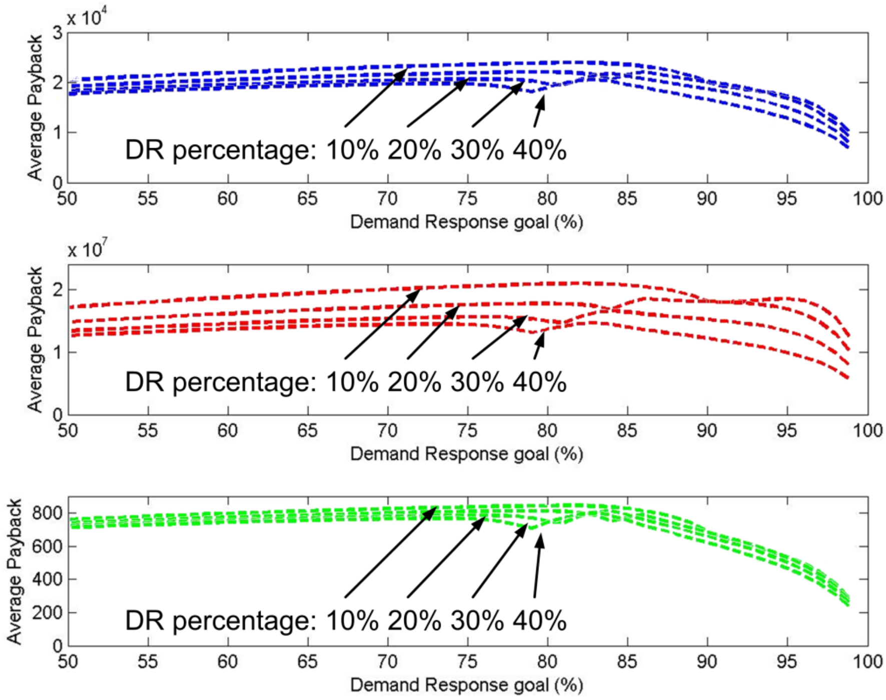

Figure 7.21: *Average payback for the maximum numbers of Subscribers scenario*

7.4.5 "High Reduction First" - Utility gain results

Figure 7.22 shows the utility gain for the minimum number of Subscribers scenario and figure 7.23 shows the utility gain for the maximum number of Subscribers scenario.

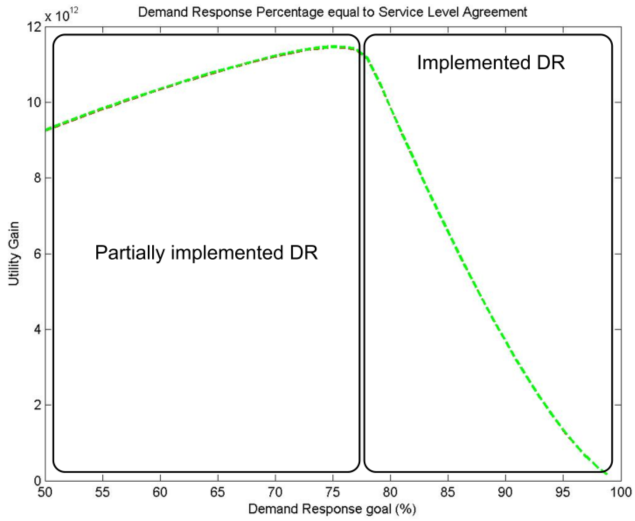

Figure 7.22: *Utility gain for the minimum numbers of Subscribers scenario*

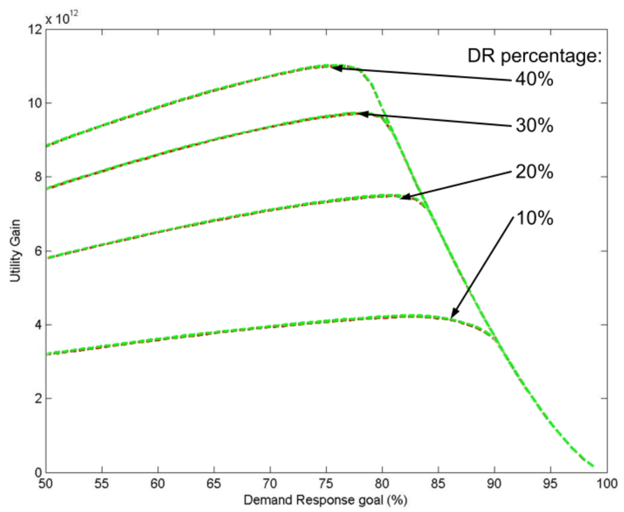

Figure 7.23: *Utility gain for the maximum numbers of Subscribers scenario*

7.4.6 "High Reduction First" - Demand Response successfulness results

For the following figure, the results are from right to left for load reduction percentage from 5% to 50% with 5% step. Figure 7.24 shows the Demand Response successfulness for all scenarios.

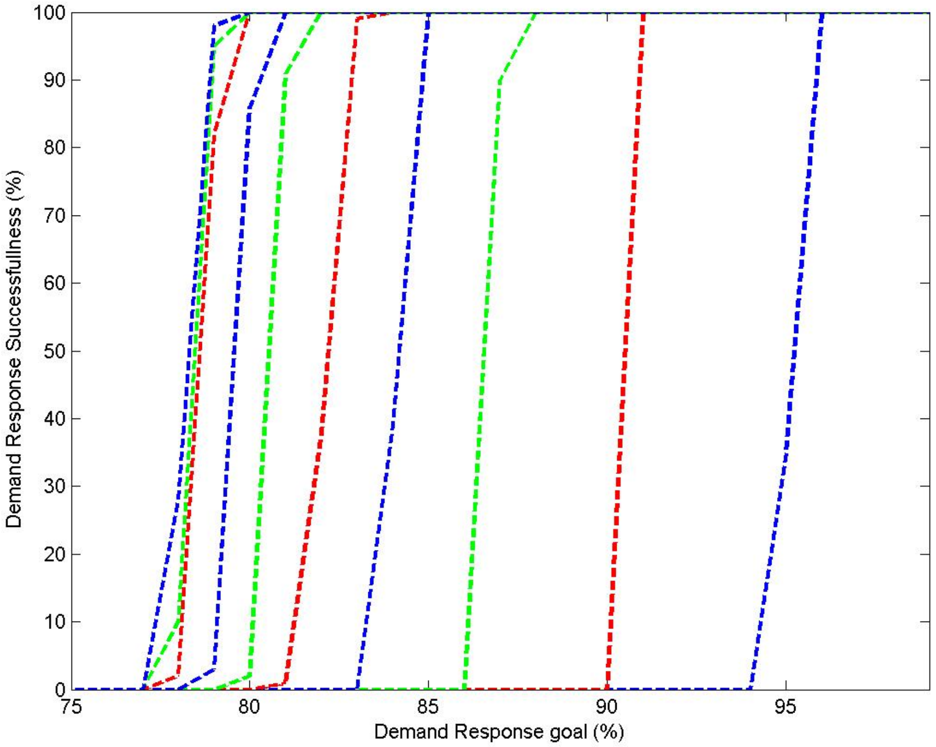

Figure 7.24: *Demand Response successfulness for all scenarios*

7.4.7 "Low Reduction First" - Number of used subscribers results

Figure 7.25 shows the number of used Subscribers for the minimum number of Subscribers scenario and figure 7.26 shows the number of used Subscribers for the maximum number of Subscribers scenario.

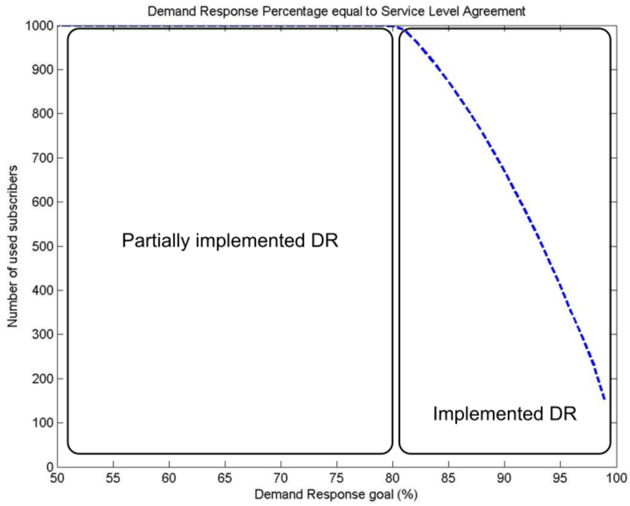

Figure 7.25: *Number of used Subscribers for the minimum number of Subscribers scenario*

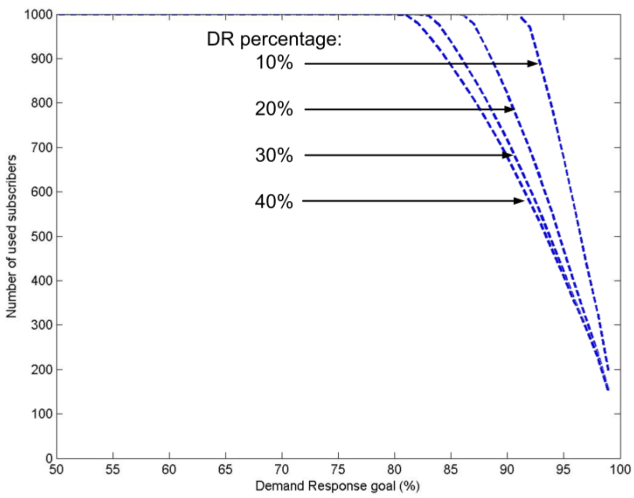

Figure 7.26: *Number of used Subscribers for the maximum numbers of Subscribers scenario*

7.4.8 "Low Reduction First" - Quality of Service results

For the following two figures, the blue line is for the average Quality of Service of the environment of Subscribers and the red line is for the average Quality of Service of the environment of Subscribers that suffer a Demand Response event. Figure 7.27 shows the Quality of Service for the minimum number of Subscribers scenario and figure 7.28 shows the Quality of Service for the maximum number of Subscribers scenario.

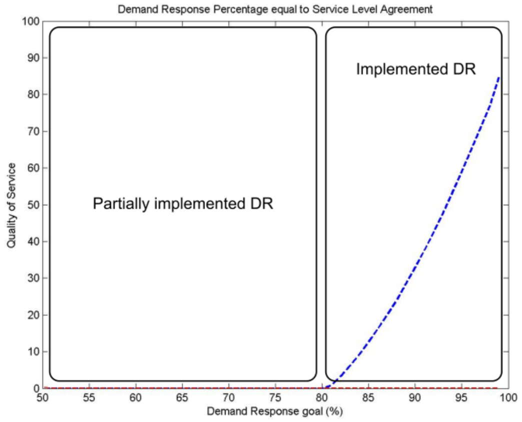

Figure 7.27: *Quality of Service for the minimum numbers of Subscribers scenario*

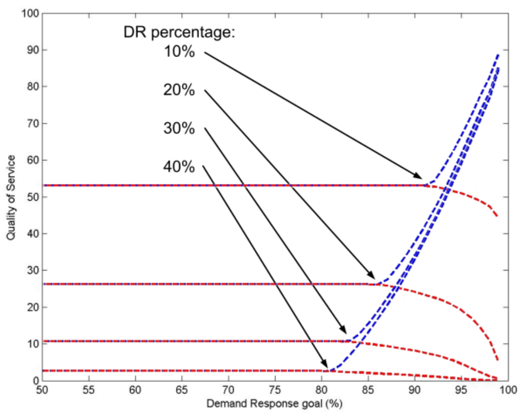

Figure 7.28: *Quality of Service for the maximum numbers of Subscribers scenario*

7.4.9 "Low Reduction First" - Total payback results

For the following two figures, the blue line is for the total payback resulting from the linear utility function, the red line is for the total payback resulting from the convex utility function and the green line is for the total payback resulting from the concave utility function. Figure 7.29 shows the total payback for the minimum number of Subscribers scenario and figure 7.30 shows the total payback for the maximum number of Subscribers scenario.

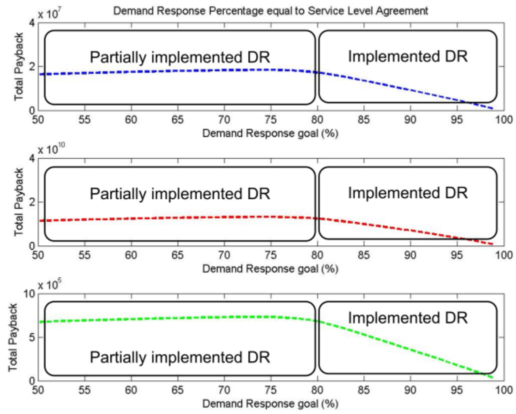

Figure 7.29: *Total payback for the minimum numbers of Subscribers scenario*

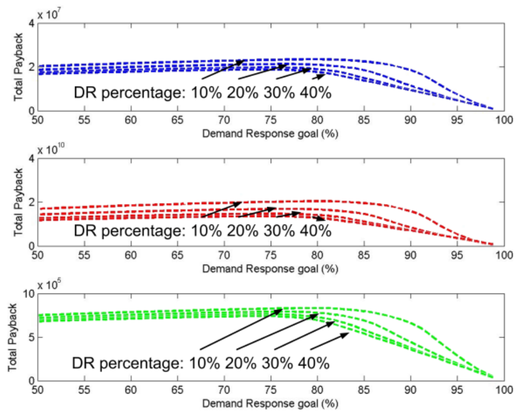

Figure 7.30: *Total payback for the maximum numbers of Subscribers scenario*

7.4.10 "Low Reduction First" - Average payback results

For the following the figures, the blue line is for the total payback resulting from the linear utility function, the red line is for the total payback resulting from the convex utility function and the green line is for the total payback resulting from the concave utility function. Figure 7.31 shows the average payback for the minimum number of Subscribers scenario and figure 7.32 shows the average payback for the maximum number of Subscribers scenario.

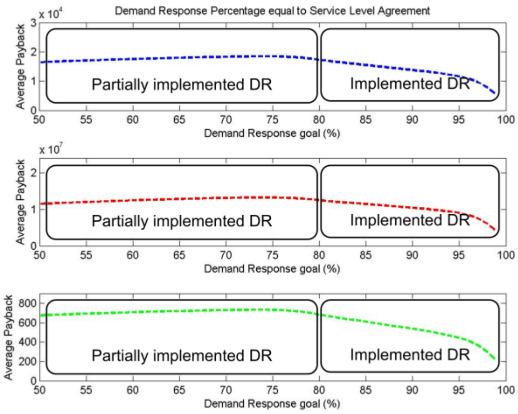

Figure 7.31: *Average payback for the minimum numbers of Subscribers scenario*

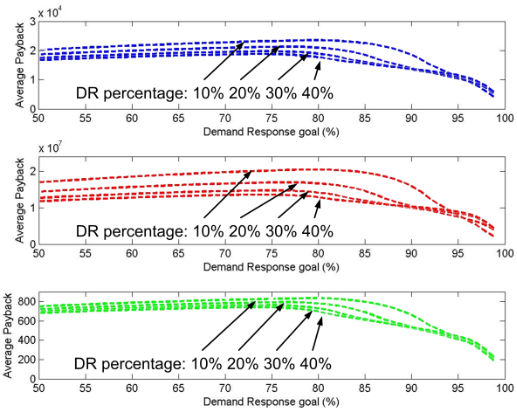

Figure 7.32: *Average payback for the maximum numbers of Subscribers scenario*

7.4.11 "Low Reduction First" - Utility gain results

Figure 7.33 shows the utility gain for the minimum number of Subscribers scenario and figure 7.34 shows the utility gain for the maximum number of Subscribers scenario.

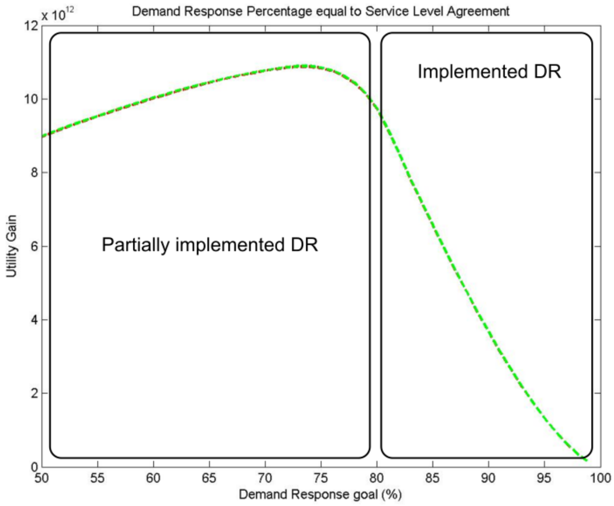

Figure 7.33: *Utility gain for the minimum numbers of Subscribers scenario*

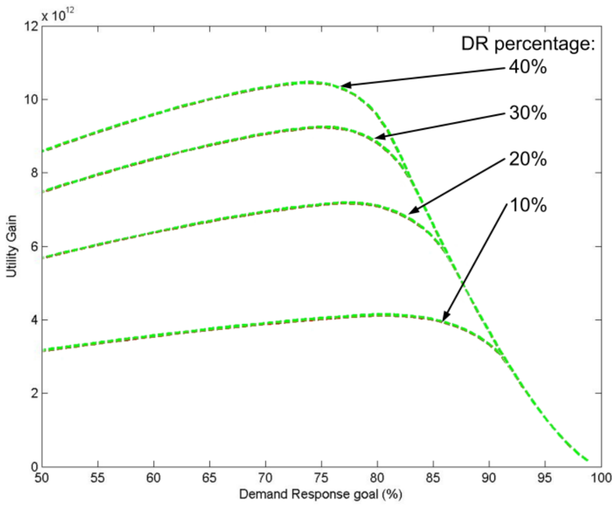

Figure 7.34: *Utility gain for the maximum numbers of Subscribers scenario*

7.4.12 "Low Reduction First" - Demand Response successfulness results

For the following figure, the results are from right to left for load reduction percentage from 5% to 50% with 5% step. Figure 7.35 shows the Demand Response successfulness for all scenarios.

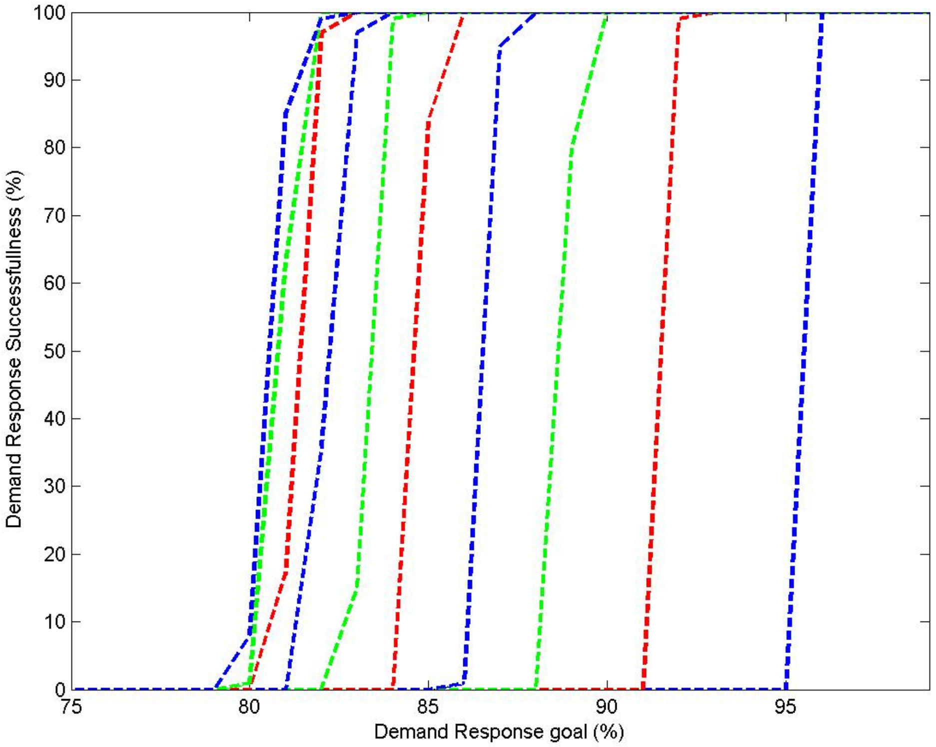

Figure 7.35: *Demand Response successfulness for all scenarios*

7.4.13 "Random" Admission - Number of used subscribers results

Figure 7.36 shows the number of used Subscribers for the minimum number of Subscribers scenario and figure 7.37 shows the number of used Subscribers for the maximum number of Subscribers scenario.

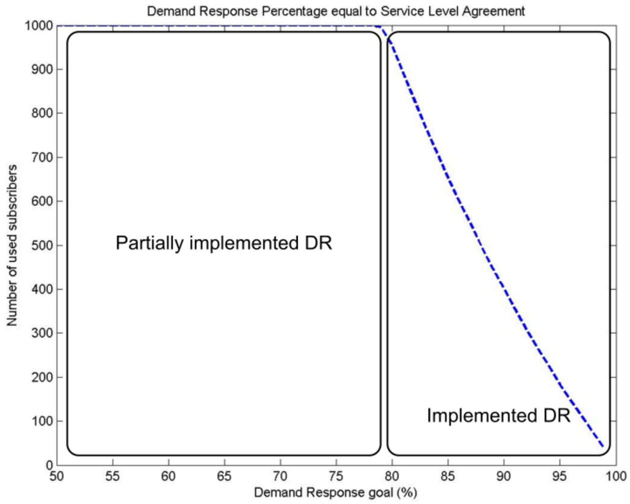

Figure 7.36: *Number of used Subscribers for the minimum number of Subscribers scenario*

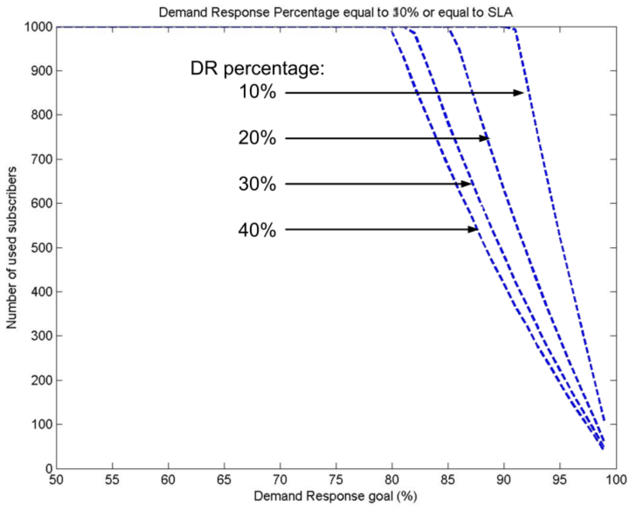

Figure 7.37: *Number of used Subscribers for the maximum numbers of Subscribers scenario*

7.4.14 "Random" Admission - Quality of Service results

For the following two figures, the blue line is for the average Quality of Service of the environment of Subscribers and the red line is for the average Quality of Service of the environment of Subscribers that suffer a Demand Response event. Figure 7.38 shows the Quality of Service for the minimum number of Subscribers scenario and figure 7.39 shows the Quality of Service for the maximum number of Subscribers scenario.

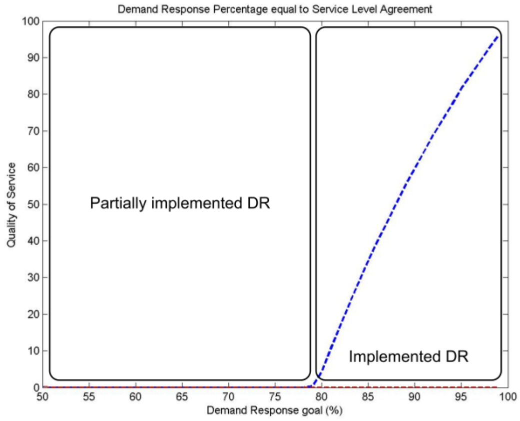

Figure 7.38: *Quality of Service for the minimum numbers of Subscribers scenario*

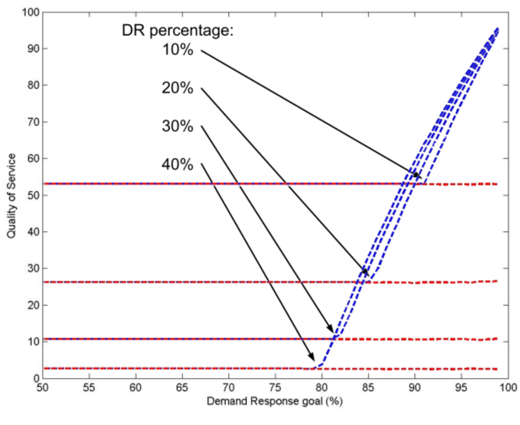

Figure 7.39: *Quality of Service for the maximum numbers of Subscribers scenario*

7.4.15 "Random" Admission - Total payback results

For the following two figures, the blue line is for the total payback resulting from the linear utility function, the red line is for the total payback resulting from the convex utility function and the green line is for the total payback resulting from the concave utility function. Figure 7.40 shows the total payback for the minimum number of Subscribers scenario and figure 7.41 shows the total payback for the maximum number of Subscribers scenario.

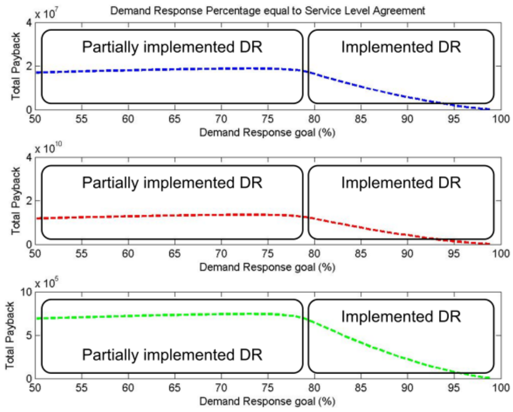

Figure 7.40: *Total payback for the minimum numbers of Subscribers scenario*

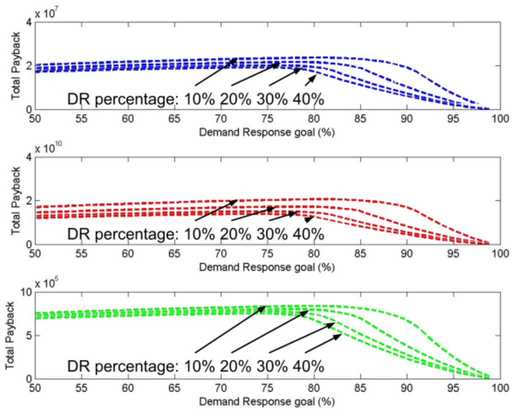

Figure 7.41: *Total payback for the maximum numbers of Subscribers scenario*

7.4.16 "Random" Admission - Average payback results

For the following the figures, the blue line is for the total payback resulting from the linear utility function, the red line is for the total payback resulting from the convex utility function and the green line is for the total payback resulting from the concave utility function. Figure 7.42 shows the average payback for the minimum number of Subscribers scenario and figure 7.43 shows the average payback for the maximum number of Subscribers scenario.

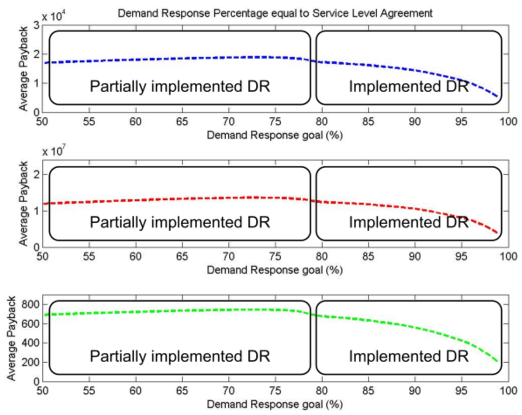

Figure 7.42: *Average payback for the minimum numbers of Subscribers scenario*

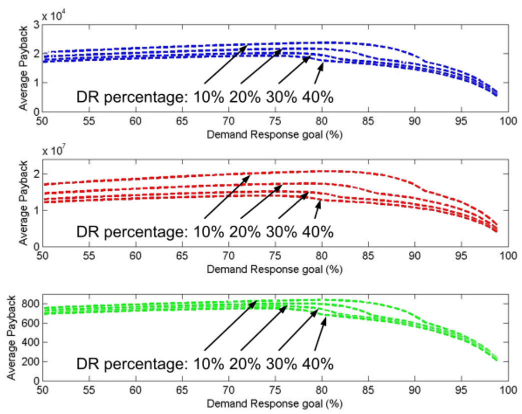

Figure 7.43: *Average payback for the maximum numbers of Subscribers scenario*

7.4.17 "Random" Admission - Utility gain results

Figure 7.44 shows the utility gain for the minimum number of Subscribers scenario and figure 7.45 shows the utility gain for the maximum number of Subscribers scenario.

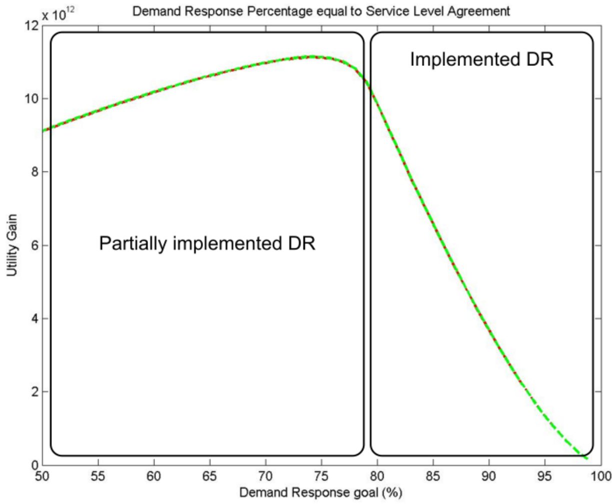

Figure 7.44: *Utility gain for the minimum numbers of Subscribers scenario*

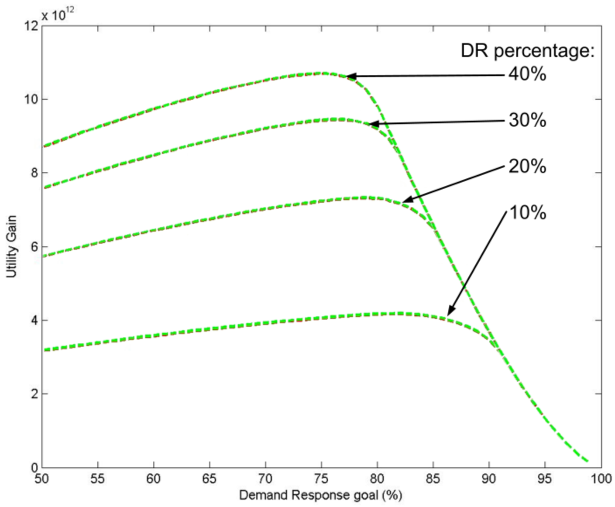

Figure 7.45: *Utility gain for the maximum numbers of Subscribers scenario*

7.4.18 "Random" Admission - Demand Response successfulness results

For the following figure, the results are from right to left for load reduction percentage from 5% to 50% with 5% step. Figure 7.46 shows the Demand Response successfulness for all scenarios.

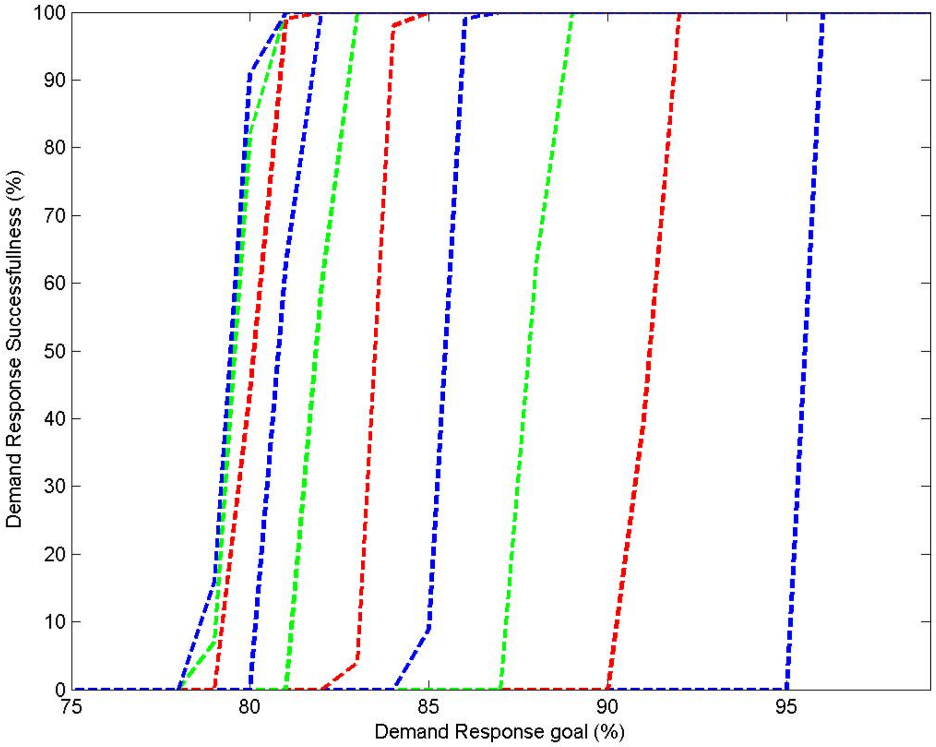

Figure 7.46: *Demand Response successfulness for all scenarios*

8 Conclusions and future work

In the two previous chapters we propose a Demand Response algorithm for grid reliability using admission control for a fair scheme and we showcase the simulation of the proposed algorithm.

In this chapter we will have an in dept discussion upon every performance metric that we extracted from the simulation of the proposed algorithm and try to provide an insightful explanation upon the behaviour of the proposed algorithm.

8.1 Discussion upon the number of used Subscribers

The different fashion under which the three Admission Control schemes operate heavily influence the number of used Subscribers that need to be admitted in order to successfully implement a Demand Response goal.

The first outcome is that as the maximum allowed by each Admission Control scenario Demand Response percentages are getting higher, the number of used Subscribers admitted by each Admission Control scheme for successfully implementing of a Demand Response goal is getting lower. This analogy is explained by the fact that when the allowed Demand Response percentages are getting higher each individual Subscriber can save more energy by accepting a Demand Response event. Thus, the number of Subscribers that needs to be admitted for a successful implementation of a Demand Response goal is getting lower as the allowed Demand Response percentages are getting higher.

A second outcome is that for the successfully implementation of the same Demand Response goal and under all Admission Control scenarios, the "High Reduction First" Admission Control scheme admits less subscribers than the "Low Reduction First" Admission Control scheme. The "High Reduction First" Admission Control scheme initially admits the Subscribers that can provide the higher load reduction where as the "Low Reduction First" Admission Control scheme initially admits the Subscribers that can provide the lower load reduction. Thus, for the same Demand Response goal the "High Reduction First" Admission Control scheme admits less subscribers than the "Low Reduction First" Admission Control scheme. The "Random" Admission Control due to its random nature behaves as the average of the other two Admission Control schemes.

For example, when implementing the "minimum number of Subscribers" Admission Control scenario, the "High Reduction First" Admission Control scheme admits 206 Subscribers for successfully implementing a Demand Response goal equal to 90%. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme admits 669 Subscribers for successfully implementing a Demand Response goal equal to 90%. Finally, the "Random" Admission Control scheme admits 400 Subscribers for successfully implementing a Demand Response goal equal to 90%.

Also for example, when implementing the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%, the "High Reduction First" Admission Control scheme admits 351 Subscribers for successfully implementing a Demand Response goal equal to 90%. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme admits 755 Subscribers for successfully implementing a Demand Response goal equal to 90%. Finally, the "Random" Admission Control scheme admits 537 Subscribers for successfully implementing a Demand Response goal equal to 90%.

Figure 8.1 shows the number of used Subscribers for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario and figure 8.2 shows the Number of used Subscribers for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%.

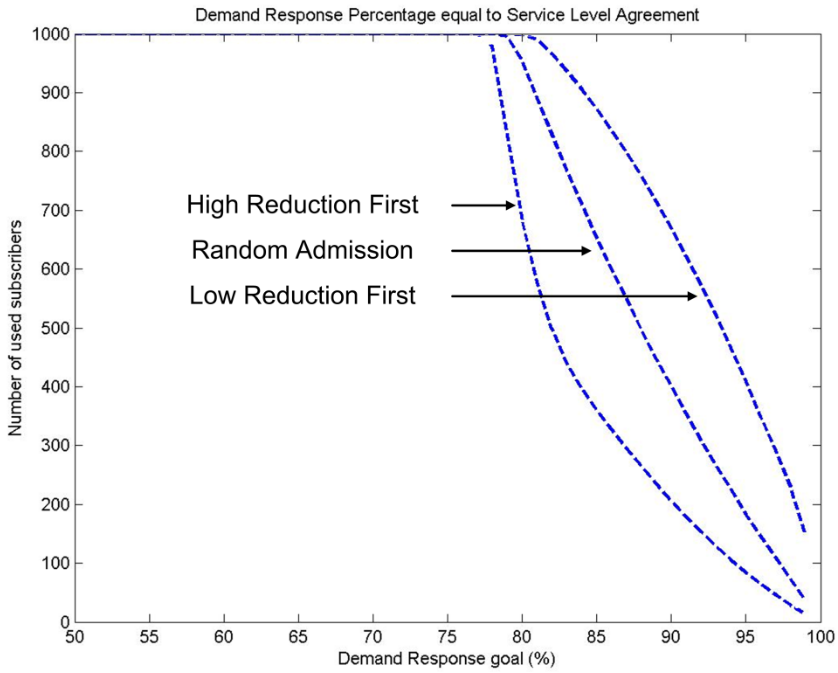

Figure 8.1: *Number of used Subscribers for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario*

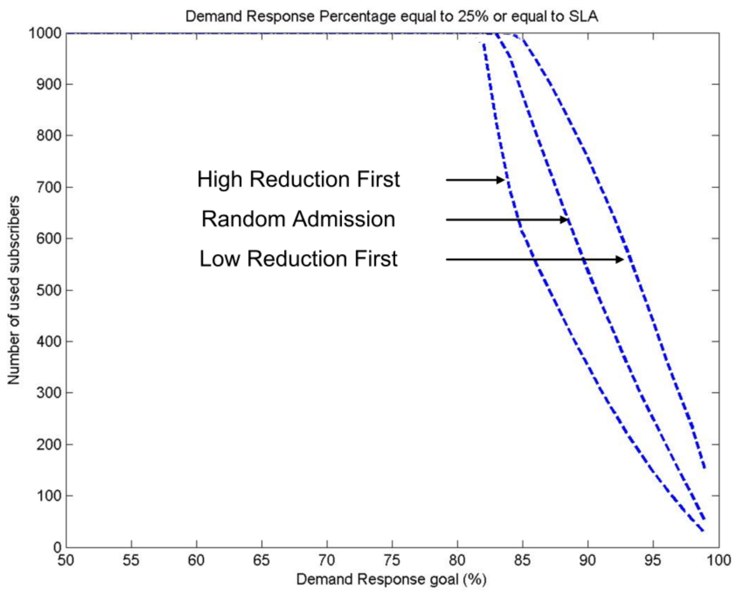

Figure 8.2: *Number of used Subscribers for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

8.2 Discussion upon the Demand Response successfulness

All three Admission Control schemes under all different Admission Control scenarios where successful in implementing the proposed Demand Response algorithm up to a certain point.

The first outcome is that as the maximum allowed by each Admission Control scenario Demand Response percentages are getting higher, the successfulness of each Admission Control scheme is getting higher. This analogy is explained by the fact that when the allowed Demand Response percentages are getting higher each individual Subscriber can save more energy by accepting a Demand Response event. Thus, the environment of Subscribers can successfully respond to higher Demand Response goals as the allowed Demand Response percentages are getting higher.

A second outcome is that under all implemented Admission Control scenarios the "High Reduction First" Admission Control scheme manage to be successful for higher Demand Response goals than the "Low Reduction First" Admission Control scheme and the "Random" Admission Control scheme. Also under all implemented Admission Control scenarios the "Low Reduction First" Admission Control scheme manage to be successful for lower Demand Response goal compared to the "High Reduction First" Admission Control scheme and the "Random" Admission Control scheme. Finally the successfulness of the "Random" Admission Control scheme is under every implemented Admission Control scenario between the successfulness of the "High Reduction First" Admission Control scheme and the "Low Reduction First" Admission Control scheme.

For example, when implementing the "minimum number of Subscribers" Admission Control scenario, the "High Reduction First" Admission Control scheme manage to be successful 100% for a Demand Response goal equal to 80%. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme manage to be successful 100% for a Demand Response goal equal to 83%. Finally, the "Random" Admission Control scheme manage to be successful 100% for a Demand Response goal equal to 81%.

Also for example, when implementing the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%, the "High Reduction First" Admission Control scheme manage to be successful 100% for a Demand Response goal equal to 84%. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme manage to be successful 100% for a Demand Response goal equal to 86%. Finally, the "Random" Admission Control scheme manage to be successful 100% for a Demand Response goal equal to 85%.

As described earlier, the "High Reduction First" Admission Control scheme admits the Subscribers more efficiently than the other two schemes. For a successfully implementation of a small Demand Response goals few Subscribers were admitted. Thus, there are Subscribers left to be admitted for a successful implementation of higher Demand Response goals. The "Low Reduction First" Admission Control scheme admits many Subscribers in order to successfully implement small Demand Response goals. Thus there are no Subscribers left to be admitted for a successful implementation of higher Demand Response goals. Finally, the "Random" Admission Control due to its random nature behaves as the average of the other two Admission Control schemes.

Figure 8.3 shows the Demand Response successfulness for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario and figure 8.4 shows the Demand Response successfulness for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%.

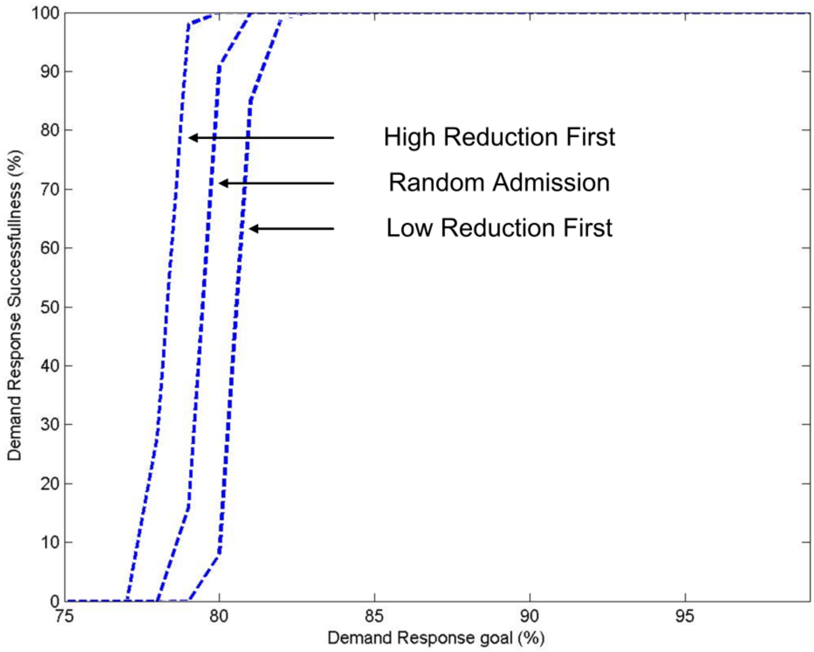

Figure 8.3: *Demand Response successfulness for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario*

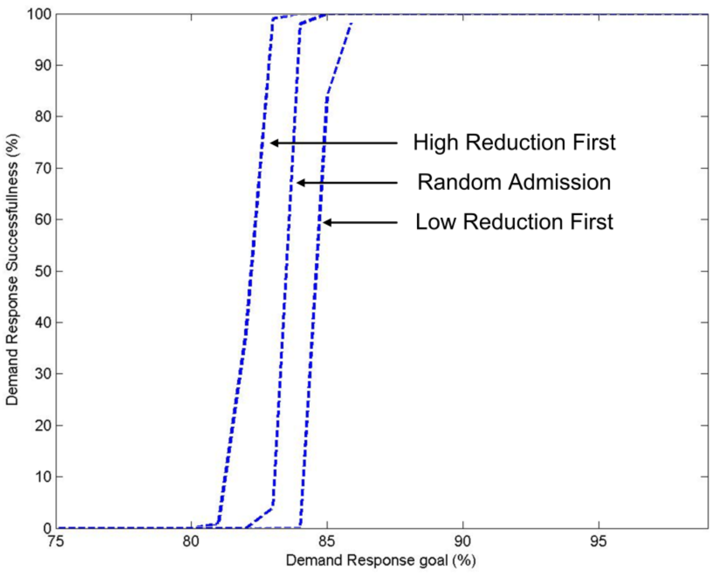

Figure 8.4: *Demand Response successfulness for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

8.3 Discussion upon the Quality of Service

The main difference of the three Admission Control schemes is the logic under which the admission of the Subscribers takes place. As preciously described, the "High Reduction First" Admission Control scheme admits the Subscribers with a more effective way than the "Low Reduction First" scheme. The "Random" Admission Control scheme due to its random nature behaves as the average of the other two Admission Control schemes. Thus, the Quality of Service is heavily influenced from the number of used Subscribers.

The first outcome is that as the Demand Response goal percentages are getting higher, the average Quality of Service of the environment of Subscribers is getting lower. This analogy is explained by the fact that when the Demand Response goal percentages are getting higher more Subscribers were admitted and suffer a Demand Response event in order to successfully implement the Demand Response goal. Thus, as the Demand Response goal percentages are getting higher, the average Quality of Service of the environment of Subscribers is getting lower.

A second outcome is that for the successfully implementation of the same Demand Response goal and under all Admission Control scenarios, the "High Reduction First" Admission Control scheme provides higher average Quality of Service of the environment of Subscribers than the "Low Reduction First" Admission Control scheme. The "High Reduction First" Admission Control scheme admits less Subscribers than the "Low Reduction First" Admission Control scheme for successfully implementing a Demand Response goal. Thus, for the same Demand Response goal the "High Reduction First" Admission Control provides higher average Quality of Service of the environment of Subscribers than the "Low Reduction First" Admission Control scheme. The "Random" Admission Control due to its random nature behaves as the average of the other two Admission Control schemes.

For example, when implementing the "minimum number of Subscribers" Admission Control scenario, the "High Reduction First" Admission Control scheme provides an average Quality of Service of the environment of Subscribers equal to 79% for successfully implementing a Demand Response goal equal to 90%. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme provides an average Quality of Service of the environment of Subscribers equal to 33% for successfully implementing a Demand Response goal equal to 90%. Finally, the "Random" Admission Control scheme provides an average Quality of Service of the environment of Subscribers equal to 60% for successfully implementing a Demand Response goal equal to 90%.

Also for example, when implementing the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%, the "High Reduction First" Admission Control scheme provides an average Quality of Service of the environment of Subscribers equal to 73% for successfully implementing a Demand Response goal equal to 90%. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme provides an average Quality of Service of the environment of Subscribers equal to 36% for successfully implementing a Demand Response goal equal to 90%. Finally, the "Random" Admission Control scheme provides an average Quality of Service of the environment of Subscribers equal to 56% for successfully implementing a Demand Response goal equal to 90%.

A third outcome is that as the Demand Response goal percentages are getting higher and under the "High Reduction First" Admission Control scheme, the average Quality of Service of the environment of Subscribers that suffer a Demand Response event is getting lower. This analogy is explained by the fact that when the Demand Response goal percentages are getting higher under the "High Reduction First" Admission Control scheme more Subscribers were admitted and suffer a Demand Response event with a Demand Response percentage that is close to the Service Level Agreement in order to successfully implement the Demand Response goal. Thus, as the Demand Response goal percentages are getting higher under the "High Reduction First" Admission Control scheme, the average Quality of Service of the environment of Subscribers is getting lower. For example, when implementing the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%, the "High Reduction First" Admission Control scheme provides an average Quality of Service of the environment of Subscribers equal to 27%, 24% and

21% for successfully implementing a Demand Response goal equal to 95%, 90% and 85% respectively.

On the other hand, as the Demand Response goal percentages are getting higher and under the "Low Reduction First" Admission Control scheme, the average Quality of Service of the environment of Subscribers that suffer a Demand Response event is also getting higher. This analogy is explained by the fact that when the Demand Response goal percentages are getting higher under the "Low Reduction First" Admission Control scheme more Subscribers were admitted and suffer a Demand Response event with a Demand Response percentage that is much lower than the Service Level Agreement in order to successfully implement the Demand Response goal. Thus, as the Demand Response goal percentages are getting higher under the "High Reduction First" Admission Control scheme, the average Quality of Service of the environment of Subscribers is getting higher. For example, when implementing the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%, the "Low Reduction First" Admission Control scheme provides an average Quality of Service of the environment of Subscribers equal to 10%, 14% and 17% for successfully implementing a Demand Response goal equal to 95%, 90% and 85% respectively.

Figure 8.5 shows the average Quality of Service of the environment of Subscribers for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario and figure 8.6 shows the average Quality of Service of the environment of Subscribers for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%.

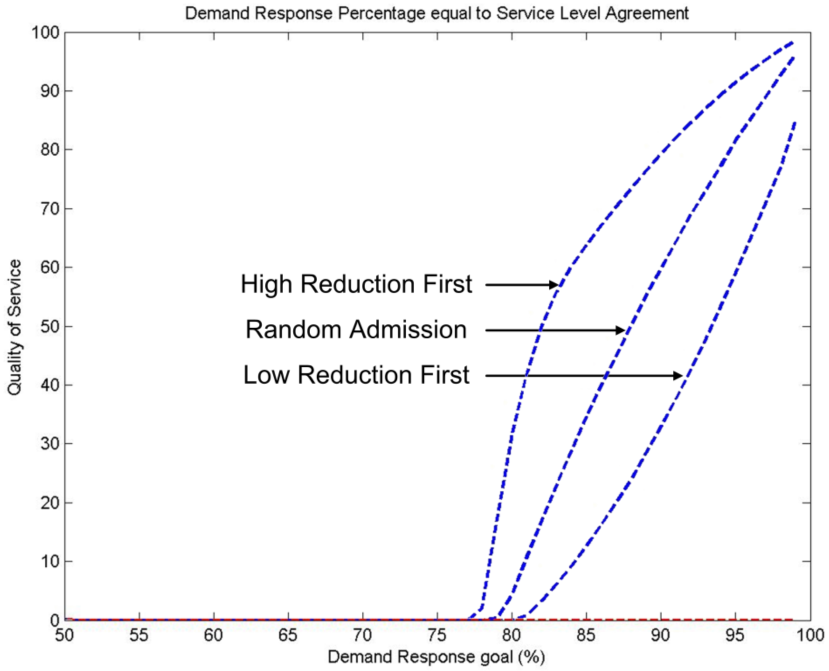

Figure 8.5: *Average Quality of Service of the environment of Subscribers for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario*

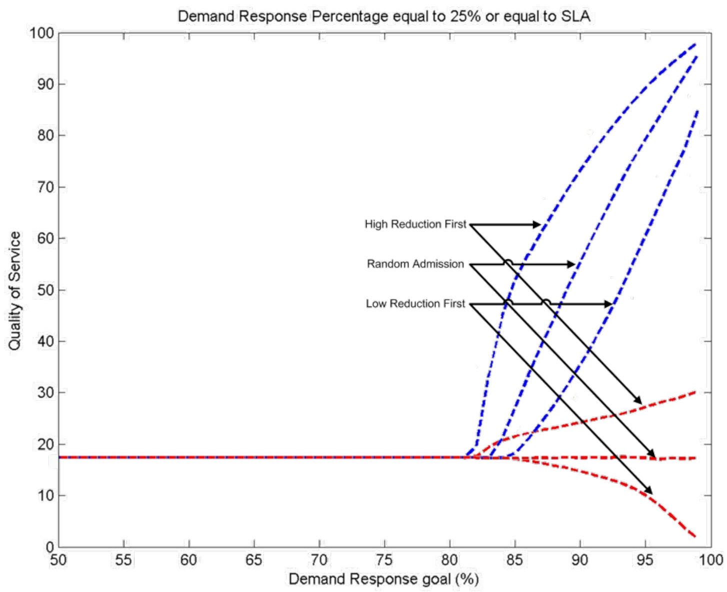

Figure 8.6: *Average Quality of Service of the environment of Subscribers for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

8.4 Discussion upon the total and average payback

Once again, the number of used Subscribers and the way that the Subscribers are admitted by every Admission Control scheme heavily influence the outcome of this performance metric.

The first outcome is that as the Demand Response goal percentages are getting higher, the total payback is getting higher. This analogy is explained by the fact that when the Demand Response goal percentages are getting higher more Subscribers were admitted and suffer a Demand Response event in order to successfully implement the Demand Response goal. Thus, as the Demand Response goal percentages are getting higher, the total payback is getting higher.

A second outcome is that for the successfully implementation of the same Demand Response goal and under all Admission Control scenarios, the "High Reduction First" Admission Control scheme provides lower total payback than the "Low Reduction First" Admission Control scheme. The "High Reduction First" Admission Control scheme admits less Subscribers than the "Low Reduction First" Admission Control scheme for successfully implementing a Demand Response goal. Thus, for the same Demand Response goal the "High Reduction First" Admission Control provides lower total payback than the "Low Reduction First" Admission Control scheme. The "Random" Admission Control due to its random nature behaves as the average of the other two Admission Control schemes.

For example, when implementing the "minimum number of Subscribers" Admission Control scenario, the "High Reduction First" Admission Control scheme provides total payback equal to $3 * 10^6$, $2 * 10^9$ and $1 * 10^5$ currency units for successfully implementing a Demand Response goal equal to 90% with a linear, a convex and a concave utility function respectively. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme provides total payback equal to $9 * 10^6$, $7 * 10^9$ and $3 * 10^5$ currency units for successfully implementing a Demand Response goal equal to 90% with a linear, a convex and a concave utility function respectively. Finally, the "Random" Admission Control scheme provides total payback equal to $6 * 10^6$, $4 * 10^9$ and $2 * 10^5$ currency units for successfully implementing a Demand Response goal equal to 90% with a linear, a convex and a concave utility function respectively.

Also for example, when implementing the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%, the "High Reduction First" Admission Control scheme provides total payback equal to $7 * 10^6$, $6 * 10^9$ and $2 * 10^5$ currency units for successfully implementing a Demand Response goal equal to 90% with a linear, a convex and a concave utility function respectively. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme provides total payback equal to $1 * 10^7$, $8 * 10^9$ and $4 * 10^5$ currency units for successfully implementing a Demand Response goal equal to 90% with a linear, a convex and a concave utility function respectively. Finally, the "Random" Admission Control scheme provides total payback equal to $8 * 10^6$, $7 * 10^9$ and $3 * 10^5$ currency units for successfully implementing a Demand Response goal equal to 90% with a linear, a convex and a concave utility function respectively.

A third outcome is that for all Admission Control schemes and under all Admission Control scenarios the total payback with a linear utility function is always an order of magnitude greater than the total payback with a concave utility function. Also, the total payback with a convex utility function is always three orders of magnitude greater than the total payback with a linear utility function. Thus, for maximum payback the concave utility function must be utilized.

Figure 8.7 shows the total payback for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario and figure 8.8 shows total payback for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%.

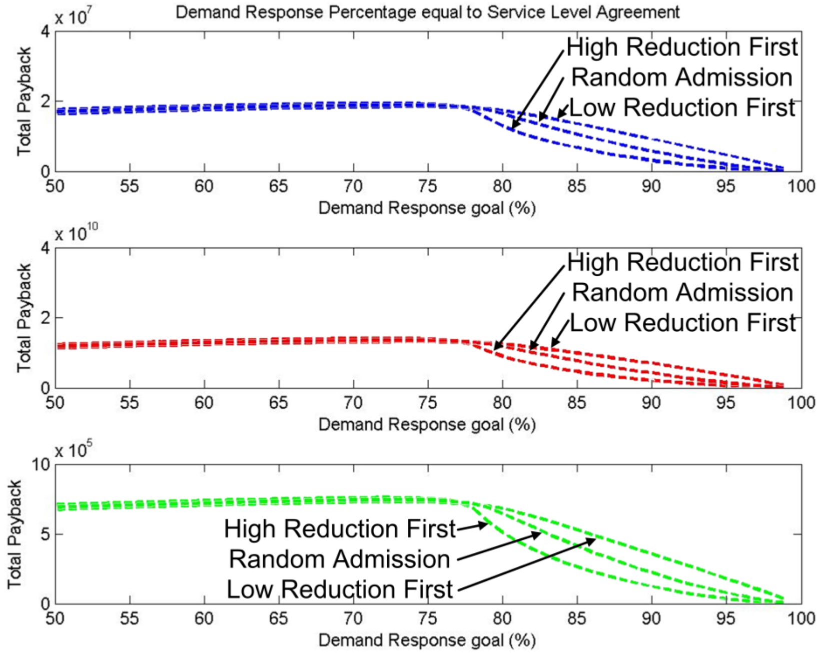

Figure 8.7: *Total payback for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario*

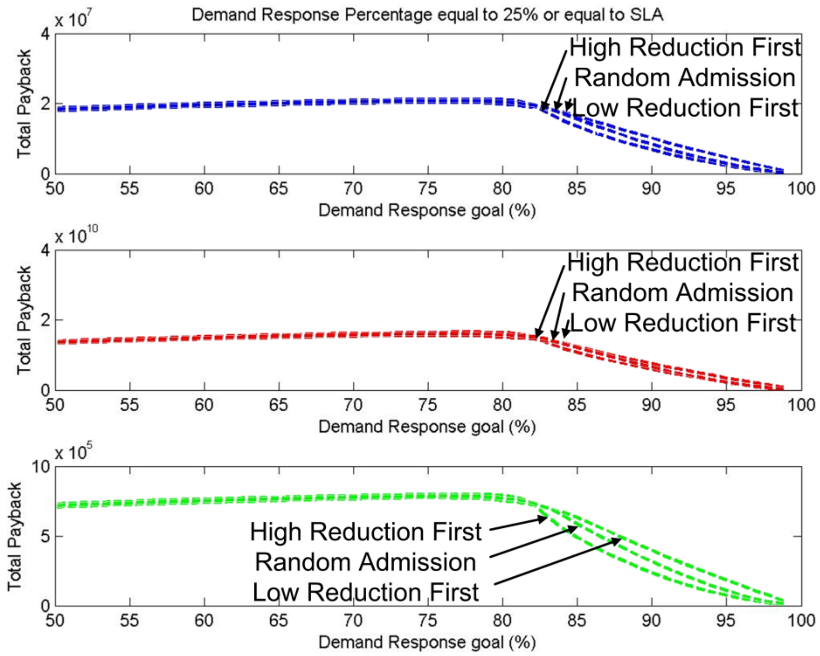

Figure 8.8: *Total payback for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

A fourth outcome is that for all Admission Control schemes, as the Demand Response percentages allowed to be accepted by Subscribers from the implemented Admission Control scenarios are getting higher, the total payback is getting lower. This analogy is explained by the fact that when the Admission Control scenarios allow higher Demand Response percentages to be accepted by Subscribers less Subscribers are admitted for successfully implementing the same Demand Response goal. Thus, as the Demand Response percentages allowed to be accepted by Subscribers from the implemented Admission Control scenarios are getting higher, the total payback is getting lower.

Figure 8.9, 8.10 and 8.11 shows the total payback for the "High Reduction First", the "low Reduction First" and the "Random" Admission Control Scheme under the "minimum number of Subscribers" Admission Control scenario and the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%.

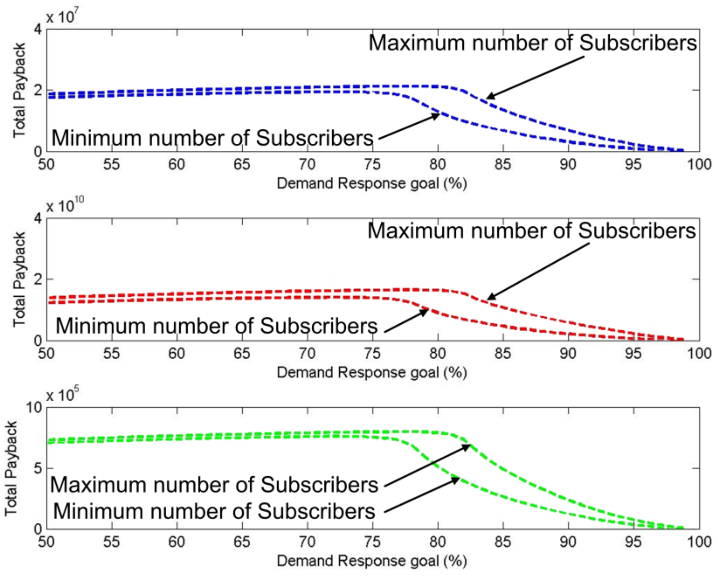

Figure 8.9: *Total payback for the "High Reduction First" Admission Control Scheme under the "minimum number of Subscribers" Admission Control scenario and the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

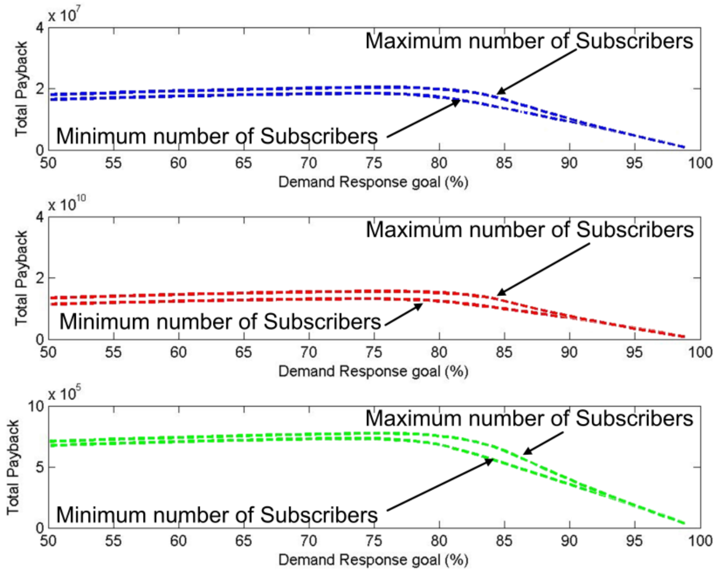

Figure 8.10: *Total payback for the "Low Reduction First" Admission Control Scheme under the "minimum number of Subscribers" Admission Control scenario and the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

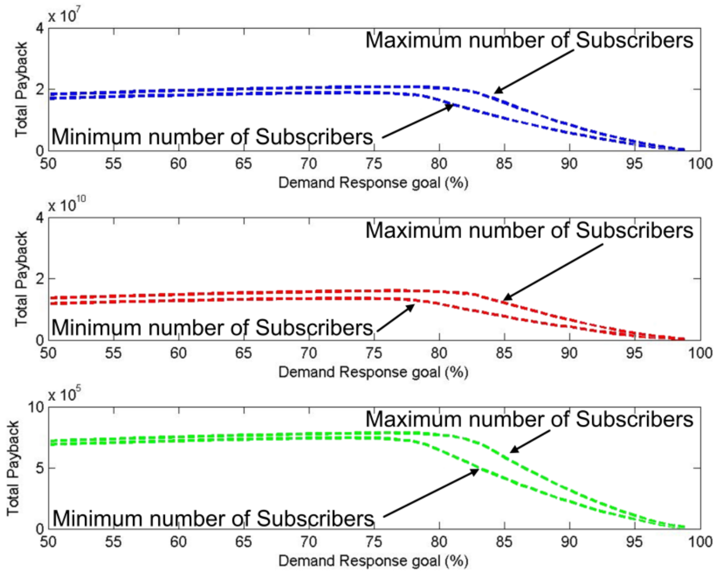

Figure 8.11: *Total payback for the "Random" Admission Control Scheme under the "minimum number of Subscribers" Admission Control scenario and the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

Finally, as far as the average payback is concerned, as the Demand Response goal is getting higher the average payback is also getting higher. This holds for the "Low Reduction First" and "Random" Admission Control schemes and for all Admission Control scenarios implemented under the before mentioned schemes. Under the "High Reduction First" Admission Control scheme and for all Admission Control scenarios implemented under the before mentioned scheme, as the Demand Response goal is getting higher the average payback is also getting higher until a maximum value. After the maximum value a small drop is observed. This drop created because under the "High Reduction First" Admission Control scheme the last Subscribers that will we admitted will provide a low reduction thus receiving low payback. Subsequently the average payback is suffering the observed drop.

Figure 8.12 shows the average payback for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario and figure 8.13 shows average payback for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%.

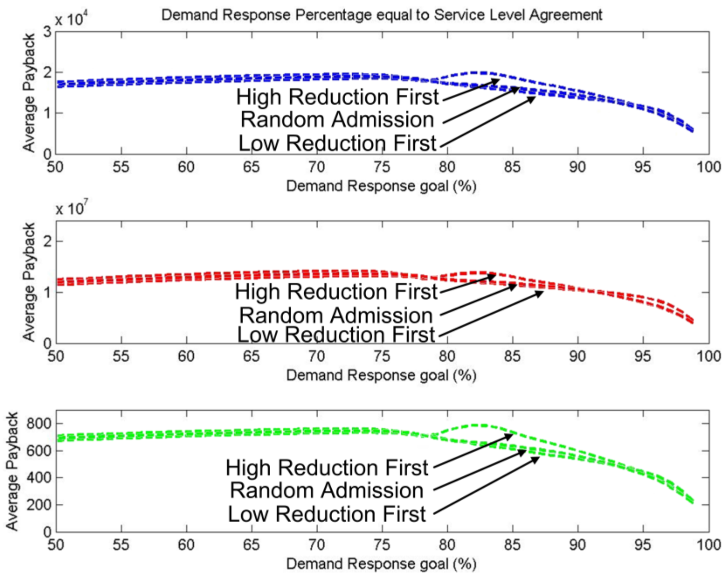

Figure 8.12: *Average payback for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario*

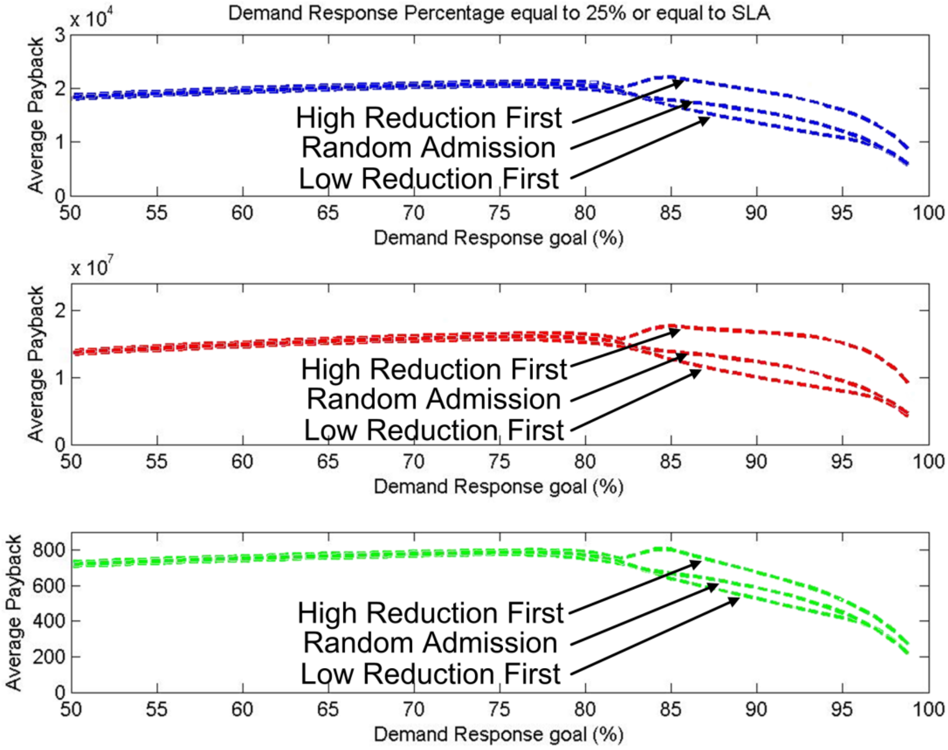

Figure 8.13: *Average payback for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

8.5 Discussion upon the Utility gain

As stated earlier, the dominant factor that influence the performance of every Admission Control scheme is the number of used Subscribers and the way that the Subscribers are admitted by every Admission Control scheme.

The first outcome is that as the Demand Response goal percentages are getting higher, the Utility gain is getting higher. This analogy is explained by the fact that when the Demand Response goal percentages are getting higher more Subscribers were admitted and suffer a Demand Response event in order to successfully implement the Demand Response goal. This result to a lower load production by the Utility after the implementation of the Demand Response and subsequently in lower cost after the Demand Response. Thus, as the Demand Response goal percentages are getting higher, the Utility gain is getting higher. This holds for all Admission Control schemes and Admission Control scenarios.

The analogy between the Demand Response goal and the Utility gain holds beyond the point where the implemented Admission Control scheme is successful. The analogy holds as long as long as the admitted Subscribers are utilized in order to provide peak load reduction. For greater Demand Response goals, where the admitted Subscribers are utilized in order to provide non peak load reduction, the Utility gain is dropping.

The second outcome is that the greater Utility gain can be achieved by implementing the "High Reduction First" Admission Control scheme. The "High Reduction First" Admission Control scheme can always admit the Subscribers by a more efficient way than the "Low Reduction First" Admission Control scheme and the "Random" Admission Control scheme. Thus, as explained before, the maximum value in which the Utility gain peaks by implementing the "High Reduction First" Admission Control scheme is greater than the maximum value in which the Utility gain peaks by implementing the "Low Reduction First" Admission Control scheme or by implementing the "Random" Admission Control scheme.

For example, when implementing the "minimum number of Subscribers" Admission Control scenario, the "High Reduction First" Admission Control scheme provide a maximum Utility gain equal to $1.15 * 10^{13}$ currency units. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme provide a maximum Utility gain equal to $1.09 * 10^{13}$ currency units. Finally, the "Random" Admission Control scheme provide a maximum Utility gain equal to $1.12 * 10^{13}$ currency units.

Also for example, when implementing the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%, the "High Reduction First" Admission Control scheme provide a maximum Utility gain equal to $8.7 * 10^{12}$ currency units. Under the same Admission Control scenario, the "Low Reduction First" Admission Control scheme provide a maximum Utility gain equal to $8.3 * 10^{12}$ currency units. Finally, the "Random" Admission Control scheme provide a maximum Utility gain equal to $8.5 * 10^{12}$ currency units.

The third outcome is that the Utility gain is six orders of magnitude greater than the total payback by the linear utility function, three orders of magnitude greater than the total payback by the convex utility function and seven orders of magnitude greater than the total payback by the concave utility function. Never the less, the greater Utility gain is provided if the concave utility function is utilized.

Figure 8.14 shows the Utility gain for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario and figure 8.15 shows Utility gain for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%.

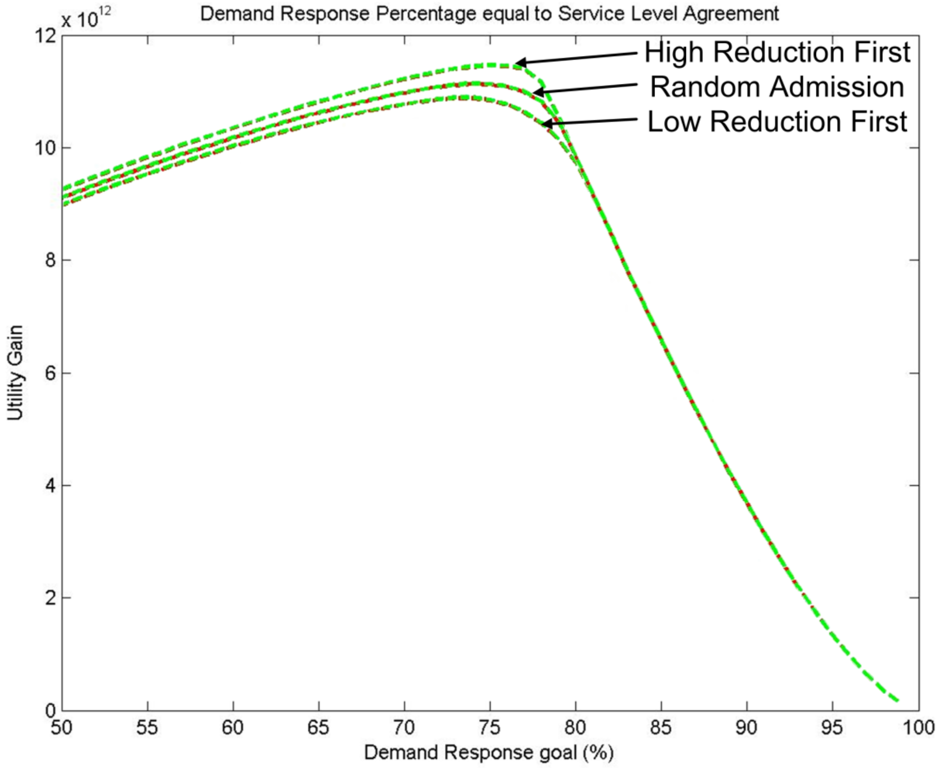

Figure 8.14: *Utility gain for all Admission Control Schemes under the "minimum number of Subscribers" Admission Control scenario*

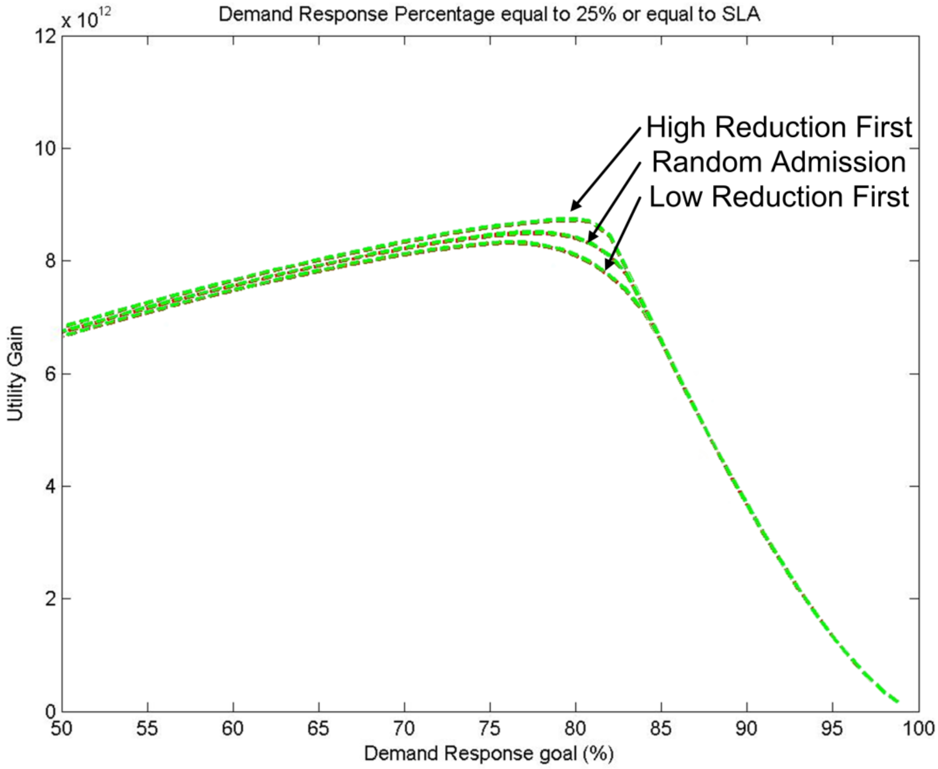

Figure 8.15: *Utility gain for all Admission Control Schemes under the "maximum number of Subscribers" Admission Control scenario with Demand Response percentage equal to 25%*

8.6 Summary of conclusions

- The number of Subscribers that needs to be admitted for a successful implementation of a Demand Response goal is getting lower as the allowed Demand Response percentages are getting higher.

- For the successfully implementation of the same Demand Response goal and under all Admission Control scenarios, the "High Reduction First" Admission Control scheme admits less subscribers than the "Low Reduction First" Admission Control scheme.

- The environment of Subscribers can successfully respond to higher Demand Response goals as the allowed Demand Response percentages are getting higher.

● Under all implemented Admission Control scenarios the "High Reduction First" Admission Control scheme manage to be successful for higher Demand Response goals than the "Low Reduction First" Admission Control scheme and the "Random" Admission Control scheme.

● As the Demand Response goal percentages are getting higher, the average Quality of Service of the environment of Subscribers is getting lower.

● For the successfully implementation of the same Demand Response goal and under all Admission Control scenarios, the "High Reduction First" Admission Control scheme provides higher average Quality of Service of the environment of Subscribers than the "Low Reduction First" Admission Control scheme.

● As the Demand Response goal percentages are getting higher and under the "High Reduction First" Admission Control scheme, the average Quality of Service of the environment of Subscribers that suffer a Demand Response event is getting lower.

● As the Demand Response goal percentages are getting higher and under the "Low Reduction First" Admission Control scheme, the average Quality of Service of the environment of Subscribers that suffer a Demand Response event is also getting higher.

● As the Demand Response goal percentages are getting higher, the total payback is getting higher.

● For the successfully implementation of the same Demand Response goal and under all Admission Control scenarios, the "High Reduction First" Admission Control scheme provides lower total payback than the "Low Reduction First" Admission Control scheme.

● For all Admission Control schemes and under all Admission Control scenarios the total payback with a linear utility function is always an order of magnitude greater than the total payback with a concave utility function. Also, the total payback with a convex utility function is always three orders of magnitude greater than the total payback with a linear utility function. Thus, for maximum payback the concave utility function must be utilized.

- For all Admission Control schemes, as the Demand Response percentages allowed to be accepted by Subscribers from the implemented Admission Control scenarios are getting higher, the total payback is getting lower.

- As the Demand Response goal percentages are getting higher, the Utility gain is getting higher.

- The greater Utility gain can be achieved by implementing the "High Reduction First" Admission Control scheme.

- The Utility gain is six orders of magnitude greater than the total payback by the linear utility function, three orders of magnitude greater than the total payback by the convex utility function and seven orders of magnitude greater than the total payback by the concave utility function. Never the less, the greater Utility gain is provided if the concave utility function is utilized.

Table 8.1 and 8.2 shows a summary of the evaluation of the performance metrics of the proposed algorithm in respect to the advantages and disadvantages towards the Utility and the Subscribers.

	Advantage for the Utility				Disadvantage for the Utility			
Admission Control Scheme:	Utility Gain	Total Payback	DR Successfulness	QoS	Utility Gain	Total Payback	DR Successfulness	QoS
High Reduction First	✓	✓	✓	✓				
Low Reduction First					✓	✓	✓	✓
Random Admission								
Admission Control Scenario:								
Minimum Number of Subscribers	✓	✓	✓					✓
Maximum Number of Subscribers				✓	✓	✓	✓	
Utility Function:								
Linear								
Convex	N/A		N/A	N/A	N/A	✓	N/A	N/A
Concave		✓						

Table 8.1: *Summary of the evaluation of the proposed algorithm in respect to the advantages and disadvantages towards the Utility*

	Advantage for the Subscribers		Disadvantage for the Subscribers	
Admission Control Scheme:	Total Payback	QoS	Total Payback	QoS
High Reduction First		✓	✓	
Low Reduction First	✓			✓
Random Admission				
Admission Control Scenario:				
Minimum Number of Subscribers			✓	✓
Maximum Number of Subscribers	✓	✓		
Utility Function:				
Linear		N/A		N/A
Convex	✓	N/A		N/A
Concave		N/A	✓	N/A

Table 8.2: *Summary of the evaluation of the proposed algorithm in respect to the advantages and disadvantages towards the Subscribers*

8.7 Future work

After the exhausting analysis of the three Admission Control schemes we conclude that the best among them is the "High Reduction First" Admission Control scheme. An obvious question that arise is if there can be another Admission Control scheme that can provide better results.

By fine tuning and by enhancing with intelligence the admission control logic of the "High Reduction First" Admission Control scheme we believe that the performance can be improved.

We will showcase the proposed changes with the following hypothetical example. Let the requested energy reduction be equal to 10 Watts. Also, let the Subscriber S_1 being able to provide energy reduction equal to 5 Watt, Subscriber S_2 being able to provide energy reduction equal to 4 Watt, Subscriber S_3 being able to provide energy reduction equal to 3 Watt, Subscriber S_4 being able to provide energy reduction equal to 2 Watt and finally Subscriber S_5 being able to provide energy reduction equal to 1 Watt.

The proposed Demand Response algorithm under the "High Reduction First" Admission Control scheme will first admit the Subscriber S_1, afterwards will admit the Subscriber S_2 and finally will admit the Subscriber S_3. Those three Subscribers combined will provide and energy reduction equal to 12 Watts. Thus, the request for energy reduction is successfully implemented. The down side to this is that by blindly admitting the next in priority Subscriber the Utility will pay for an energy reduction that is unwanted and the Subscriber S_3 will suffer a reduction on his Quality of Service in excess of what is really needed. Figure 8.16 shows a hypothetical implementation of the "High Reduction First" Admission Control scheme.

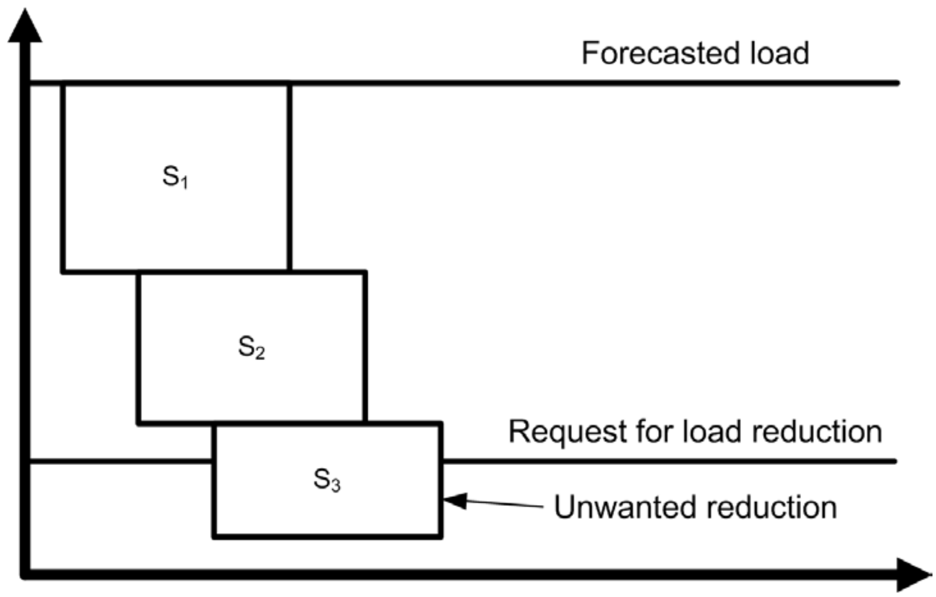

Figure 8.16: *Implementation of the "High Reduction First" Admission Control scheme*

We propose an improvement upon the "High Reduction First" Admission Control scheme that can adapt and shape the logic of admission of the Subscribers according to the current state of the system. Thus, the term "Adaptive" Admission Control scheme emerge.

The "Adaptive" Admission Control scheme is executed in the same way as the "High Reduction First" Admission Control scheme up to the point where a Subscriber or a group of Subscribers can provide sufficient energy reduction in order for the request for energy reduction to be successfully implemented while at the same time minimizing the cost for the Utility. Figure 8.17 shows a hypothetical implementation of the "Adaptive" Admission Control scheme.

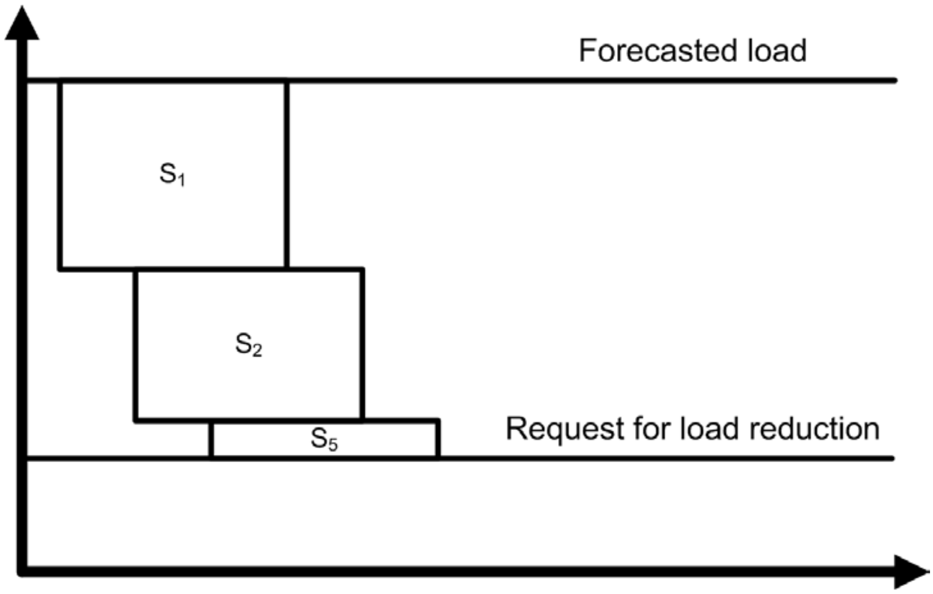

Figure 8.17: *Implementation of the "Adaptive" Admission Control scheme*

We believe that the "Adaptive" Admission Control scheme can provide higher Utility gain and higher Quality of Service. At the same time, the "Adaptive" Admission Control scheme can be as successfully, use the Subscriber in the same fashion and provide fairness towards the Subscribers as the "High Reduction First" Admission Control scheme. We advantages of the "Adaptive" Admission Control scheme are maximized if the Subscribers are admitted while being grouped in large numbers.

References

[1] George Koutitas, School of Science and Technology, International Hellenic University, Lecture 1, "Lecture Notes on Green ICT", MSc in ICT Systems, 2012-2013

[2] Gary Sorkin, "Smart Grid Business Development", Smart Grid Test Lab, Pleasanton, CA, USA

[3] "An Overview of the Evolving Smart Grid, the NIST Smart Grid Interoperability Program and the Smart Grid Interoperability Panel", National Institute of Standards and Technology, U.S. Department of Commerce

[4] Barry Haaser, Chris Kotting, Rolf Bienert, "Creating Smart Grid Ready Products Using A Modular Communication Interface", USNAP Alliance

[5] Michael Caramanis, Boston University, "Flexible Demand Response in Smart Grid Supported Power Markets", 25th Annual CCW, Hyannis MA, USA, Smart Grid Session, October 10, 2011

[6] Jorge Vasconcelos, NEWES, New Energy Solutions, "Smart Grid And Demand Response", Northwestern Workshop on Challenges for Future Electricity Markets Northwestern University, May 5, 2012, Evanston, IL, USA

[7] Rajit Gadh, UCLA, Smart Grid Energy Research Center

[8] "Smart Grid, Energy Storage & Demand Response Deployment and Plans", October 27, 2010, gridSMART, American Electric Power

[9] Andres Carvallo, "Smart Grid Network Management System", Proximetry, January, 2012

[10] Srinivasan Keshav, Catherine Rosenberg, "How Internet Concepts and Technologies Can Help Green and Smarten the Electrical Grid", Green Networking 2010, August 30, 2010, New Delhi, India.

[11] George Koutitas, School of Science and Technology, International Hellenic University, Lecture 2, "Lecture Notes on Green ICT", MSc in ICT Systems, 2012-2013

[12] Zhong Fan, "A Distributed Demand Response Algorithm and Its Applications to PHEV Charging in Smart Grid.", IEEE Trans. on Smart Gird, vol. 3, num. 3, pp. 1280-1290, 2012

[13] Wang, J., S. Kennedy, and J. Kirtley, "A new wholesale bidding mechanism for enhanced demand response in smart grids." Innovative Smart Grid Technologies (ISGT), 2010. 2010. 1-8. © 2010 Institute of Electrical and Electronics Engineers

[14] Manisa Pipattanasomporn, Murat Kuzlu and Saifur Rahman, "An Algorithm for Intelligent Home Energy Management and Demand Response Analysis", IEEE Trans. on Smart Gird

[15] David Kathan, "Assessment of Demand Response and Advanced Metering", Staff Report, Federal Energy Regulatory Commission, December 2012

[16] "Automated Demand Response Using OpenADR, Application Guide", Rev. 3, July, 2011, SIEMENS

[17] Ed Koch and Tariq Samad, "Demand Response and Energy Efficiency for the Smart Grid", Stanford University, 16 May 2011

[18] Nirmal Nair, "Demand Response and Smart Grid", Power Systems Group (PSG), Electrical & Computer Engineering Department, New Zealand Wind Energy Association Workshop, "Connecting Renewable: Grids, Networks and Markets", 24th October 2012, Wellington, The University of Auckland

[19] "Demand Response Measurement & Verification, Applications for Load Research", AEIC Load Research Committee, March 2009, Association of Edison Illuminating Companies

[20] JC Fuller, N Prakash Kumar and CA Bonebrake, "Evaluation of Representiative Smart Grid Investment Grant Project Technologies: Demand Response", Prepared for the U.S. Department of Energy under Contract DE-AC05-76RL01830, Pacific Northwest National Laboratory

[21] "Fast Demand Response", White Paper, PARC, Coyote Hill Road, Palo Alto, California, USA

[22] Zheng Wen, Hamid Reza Maei and Dan O'Neill, "Optimal Demand Response Using Device Based Reinforcement Learning", June 12, 2012, Department of Electrical Engineering, Stanford University

[23] "Smart Demand Response: A Discussion Paper", energyuk

[24] Matthew Light, "Electricity Sector Cybersecurity Risk Management Process Guideline, Infrastructure Security and Energy Restoration", 9 January 2012, Office of Electricity Delivery and Energy Reliability, U.S. Department of Energy

[25] Ben Boyd, "Recent Regulatory Actions and Potential Requirements, New Laws/Rules Concerning Cyber Security and Privacy", Smart Grid Educational Webcasts, February 13, 2011, EnerNex

[26] Barry Haaser and Rolf Bienert, "Enabling The Standard for Automated Demand Response", Welcome to the OpenADR Seminar Austin, The OpenADR Alliance

[27] George Koutitas, School of Science and Technology, International Hellenic University, Lecture 2, "Lecture Notes on Green ICT", MSc in ICT Systems, 2012-2013

[28] Barry Haaser and Rolf Bienert, "Enabling The Standard for Automated Demand Response", The OpenADR Alliance

[29] "The OpenADR Primer, An introduction to Automated Demand Response and the OpenADR Standard", The OpenADR Alliance

[30] Charles McParland, "Home Network Technologies and Automating Demand Response", Environmental Energy Technologies Division, December 2008, Ernest Orlando Lawrence Berkeley National Laboratory

[31] S. Kiliccote, M.A. Piette, J.H. Dudley, E. Koch and D. Hennage, "Open Automated Demand Response for Small Commercial Buildings", July 2009, Ernest Orlando Lawrence Berkeley National Laboratory

[32] "OpenADR 2.0 Profile Specification A Profile", Revision Number: 1.0, The OpenADR Alliance

[33] "OpenADR 2.0 Profile Specification B Profile", Revision Number: 1.0, The OpenADR Alliance

[34] Roger Levy, Chuck Goldman and Sila Kiliccote, "Demand Response", NARUC Webinar, May 4, 2011, Demand Response Research Center, Lawrence Berkeley National Laboratory

[35] David Clark, "Overview of the ZigBee Cluster Library", ZigBee Webinar, November 14th, 2007, ZigBee Alliance

[36] Mike Bourton, "SEP1.x to SEP2.0 Migration for the Home Area Network (The Why's and How 2's)", GRID2HOME, Smart Energy Solutions

[37] Tobin Richardson, Robby Simpson, Skip Ashton, Jeff Gooding and Tobin Richardson, "Smart Energy Profile 2 (SEP 2) – IP based Energy Management for the Home", Webinar, May 14, 2013, ZigBee Alliance

[38] Girish Ghatikar and Ed Koch, "Deploying Systems Interoperability and Customer Choice within Smart Grid", Ernest Orlando Lawrence Berkeley National Laboratory, Grid-Interop 2012, December 3-6, 2012, Irving, TX, USA

[39] Rumen Kyusakov, Jens Eliasson, Jan van Deventer, Jerker Delsing and Robert Cragie, "Emerging Energy Management Standards and Technologies - Challenges and Application Prospects", Lulea University of Technology, Sweden

[40] "Assessment of Communication Standards for Smart Appliances: The Home Appliance Industry's Technical Evaluation of Communication Protocols", October 2010, Association of Home Appliance Manufacturers, Washington, USA

[41] K. Herter, S. Wayland and J. Rasin, "A Successful Case Study of Small Business Energy Efficiency and Demand Response with Communicating Thermostats", International Energy Program Evaluation Conference 2009, August 12-14, 2009, Portland, OR, USA

[42] Carey Williamson, "Call Admission Control", Department of Computer Science, University of Calgary

[43] Georgios I. Tsiropoulos, Dimitrios G. Stratogiannis and Eirini Eleni Tsiropoulou, "Call Admission Control in Mobile and Wireless Networks", National Technical University of Athens, Greece

[44] Sanchita Ghosh and Amit Konar, "Call Admission Control in Mobile Cellular Networks", Studies in Computational Intelligence, Springer

[45] Robert Akl and Asad Parvez, "Global versus Local Call Admission Control in CDMA Cellular Networks", University of North Texas, USA

[46] George Koutitas, "Control of Flexible Smart Devices in the Smart Grid", IEEE Transactions On Smart Grid, VOL. 3, NO. 3, September 2012

[47] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", third edition, The MIT Press

Appendix A: List of figures

Appendix B: List of tables

Appendix C: List of code snippets

Number	Title	Page
4.1	The oadrResponse payload	62
4.2	The oadrDistributeEvent payload	62
4.3	The oadrCreatedEvent payload	62
4.4	The oadrRequestEvent payload	63
4.5	The EiEvent payload	64
4.6	The eventDescriptor payload	65
4.7	The eiActivePeriod payload	65
4.8	The eiTarget payload	65
4.9	Various IDs payloads	66
4.10	The responseDescirption payload	66
4.11	The eiResponse payload	66
4.12	The eventResponse payload	67
4.13	The eiCreateEvent payload	67
4.14	The oadrPoll payload	78
7.1	Importing the daily load profile form the source file and multiply by 1000	103
7.2	Setting the number of Subscribers equal to 1000	104
7.3	Implementation of the random normal distributed time offset	105
7.4	Implementation of the random normal distributed amplitude factor	105
7.5	Setting the random normal distributed Service Level Agreement and effective Demand Response time	107
7.6	Defining the total load forecast of all subscribers	108
7.7	Defining the requested reduction percentage and the request for load reduction	108
7.8	Defining the total forecasted load production cost	109
7.9	Pre-processing of data, Admission Control logic and Admission Control algorithm implementation	111
7.10	Implementation of the alternative Admission Control scenarios	112
7.11	Defining the number of subscribers that suffer a Demand Response event	117
7.12	Defining the average Quality of Service of all subscribers and the average Quality of Service of all subscribers that suffer a Demand Response event	117
7.13	Defining the total and average Payback from the Utility to the Subscribers	118
7.14	Defining the Utility gain	118
7.15	Defining the Demand Response successfulness	118