A New Vehicle-to-Grid System for Battery Charging exploiting IoT protocols

Contribution:
P.Chu, R. Gadh
Department of Mechanical and Aerospace Engineering, UCLA
Los Angeles, California, USA

A. Fachechi, L. Mainetti, L. Palano, L. Patrono, M.L. Stefanizzi, R. Vergallo
Department of Innovation Engineering, University of Salento
Lecce, Italy

Acknowledgement
This material is based upon work supported by the Department of Energy under Award Number DE-OE0000192.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, the Los Angeles Department of Water and Power, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

A New Vehicle-to-Grid System for Battery Charging exploiting IoT protocols

A. Fachechi, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, R. Vergallo
Dept. of Innovation Engineering
University of Salento
Lecce, Italy
{alessio.fachechi, luca.mainetti, laura.stefanizzi, luca.palano, luigi.patrono, roberto.vergallo}@unisalento.it

P. Chu, R. Gadh
Dept. of Mechanical and Aerospace Engineering
University of California Los Angeles (UCLA)
Los Angeles, CA
{peterchu, gadh}@ucla.edu

***Abstract*— The continuously rising demand for electricity has prompted governments and industries to research more effective energy management systems. The Internet of Things (IoT) paradigm is a valuable add-on for controlling and managing the energy appliances such as Plug-in Electrical Vehicles (PEV) charging stations. In this paper, we present a Demand Response (DR) implementation for PEV charging stations able to use Wireless Sensor Network technologies based on the Constrained Application Protocol (CoAP). We developed a self-service kiosk system by which the user can autonomously swipe his/her credit card and choose the charging station to enable. When a user plugs his/her vehicle to the station, s/he subscribes his availability to share a portion of its energy. When the grid requests a contribution from the PEVs, the kiosk sends a CoAP message to the available stations and the energy flow is inverted (Vehicle-to-Grid). At the end of the charging process, the user's credit card gets charged with a discounted bill.**

***Index Terms*—CoAP; Wireless Sensor Networks; Demand Response; 6LowPAN; IPv6; Internet of Things.**

I. Introduction

In recent years, the increasing global demand for electricity, as well as the growing of carbon emissions, has prompted governments and energy providers to promote the use of renewable energies and their integration within the existing transmission and distribution systems. The technological paradigm behind the realization of the future electrical grid is known as Smart Grid (SG). Specifically, a SG is an advanced power grid able to provide high reliability, availability, and safety through the use of modern sensing and communication technologies. It spreads the intelligence of the energy distribution and control system from some central core to many peripheral nodes, thus enabling more accurate monitoring of energy losses and a bidirectional interaction among producers, consumers and other entities.

This new vision of the power grid fits into the broader concept of Internet of Things (IoT) [1]. Through the IoT, consumers, manufacturers and utility providers will uncover new ways to manage devices, conserve resources and save money. Basing the integrated information and communication infrastructure of the SG on the Internet model will allow energy providers to access to energy information in a simple, standardized and cost-efficient manner. On the other hand, also consumers will benefit from such an infrastructure, since it will enable them to observe their energy consumption in real time and operate their devices in a way that minimizes consumption and reduces costs.

Such features represent the basis for the success of the Demand Response (DR) market. In electricity grid, DR is a mechanism for achieving energy efficiency through managing customer consumption of electricity in response to supply conditions (e.g., end users reduce their demand at critical times or in response to market prices). An international standardization effort is currently running to guarantee full interoperability among the involved heterogeneous communication architecture. In this contest, Smart Energy Profile (SEP) version 2.0 [2] and Open Automated Demand Response (OpenADR) version 2.0 [3] represent two of the most promising solutions. Their scopes, however, are different: SEP is focused on Home Area Network devices, while OpenADR is designed for high level communications between the Automated Metering Infrastructure (AMI) Systems, utilities' communication networks and Independent System Operators (ISOs). Today, utilities and governments worldwide are using OpenADR to manage the growing demand for electricity and peak capacity of the electric systems. This low cost communications infrastructure is used to improve the reliability, repeatability, robustness, and cost-effectiveness of DR systems.

The latest OpenADR version (2.0) supports a small number of transport protocols to accommodate different deployment scenarios: simple HTTP, and XMPP [4]. Nevertheless, the use of small, cheap, and resource-constrained devices able to monitor the energy grid will advocate the use of specifically designed transport protocols such as the ones defined in the Wireless Sensor Network (WSN) field. WSN is deemed to be the key enabling technology to make the SG implementable in practice [5]. To this end, the recent efforts carried out by Internet standardization organizations such as the IETF and W3C towards the IoT vision are particularly relevant. The main outcomes of such research efforts are the definition of an adaptation layer for the IPv6 protocol over Low-power Wireless Personal Area Networks (6LoWPAN) [6] and the

Constrained Application Protocol (CoAP) [7], an HTTP-like protocol especially designed for resource constrained devices. The use of these emerging IoT technologies can provide a viable solution towards the realization of an Internet-based Smart Grid system [8].

In this paper, we propose the use of CoAP as an alternative transport protocol at the edges of an OpenADR-inspired architecture. Our publish/subscribe architecture implements the OpenADR two-way signaling systems, providing the servers, which publish information, and the clients, which subscribe the information. As a use case, we considered the process of a user charging his/her Plug-in Hybrid Electric Vehicle (PHEV). PHEVs and Plug-in Electric Vehicles (PEVs) are important elements of the next SG. They are seen as a significant transportation option to reduce greenhouse emissions and have attracted a lot of attention in both academia and industry. However, the electricity demand of these electric vehicles could pose significant challenges to the grid [9]. In this respect, a new technological concept, known as Vehicle-to-Grid (V2G), has been recently introduced [10]. It enables the utilization of vehicle batteries for grid-side benefit. Specifically, the basic idea behind the concept of V2G is to use EVs as a source of reserve power while the vehicles are parked. Indeed, PHEVs/PEVs have the potential to transfer power to the grid, thus allowing the control of the energy demand during critical peak situations. In our architecture, we designed and developed a prototypal self-service Kiosk that accepts credit cards and control the charging processes. Charging stations are connected to the Kiosk using the 6LowPAN WSN technology. We used CoAP as application-level protocol to control the station's behavior. The architecture implements a DR technique by which each charging station limits or inverts (V2G) the current flow during the charging process. A DR server sends the Kiosk the current control message when a critical condition is detected on the grid. As soon as the credit card has been verified, the charging process is enabled and the Kiosk subscribes the PEV's availability to share a portion of the retained energy. At the end of the charging process, the user's credit card gets charged with a discounted bill.

Our work is aimed at proofing that not only CoAP fits very well within an OpenADR architecture, but using its REST approach, it lets the definition of DR scenarios very comfortable and natural. The rest of the paper is organized as follows. Section II provides an overview about the considered standards. In Section III the architectural design behind our OpenADR implementation is described. Section IV presents a validation scenario. Finally, in Section V, we summarize the collected results as well as sketching future research.

II. Technological Background

A. OpenADR 2.0

The Open Automated Demand Response (OpenADR) is described in its formal specification [3] as 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 this data model is to enable a power network infrastructure where each demand response event is fully automated, with no manual intervention. Moreover, the concept of an open specification allows anyone to implement the signaling systems, providing the automation server or the automation clients.

In more detail, OpenADR standard specifies the APIs of the Demand Response Automation Server (DRAS), which serves as the common platform between all providers and consumers of electricity. For instance, assume that a Virtual Top Node (VTN) such as a utility or an ISO is providing information to the DRAS about what has changed and what to schedule, e.g. the absolute price or a change in the price per kilowatt-hour, a fixed amount or a percentage of load to shed or shift, reliability and emergency signals, and so on. Then, one or more Virtual End Nodes (VENs) of some end-use customer may be notified of this and consequently decide if to intervene or not, e.g. an Energy Management System (EMS) programmed to temporarily offset building temperatures by several degrees, and to dim or turn-off non-essential lights.

B. Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP) is a specialized Web transfer protocol defined by the IETF Constrained RESTful Environments (CORE) working group to allow the implementation of REST mechanisms on constrained devices.

Today, it is one of the most used communication protocol in the IoT. The main idea of this protocol is to provide a lightweight access to physical resources in order to meet the limited capabilities of embedded devices. Its design is similar to that of HTTP, since CoAP provides a request/response model interaction between two end-points and it includes key concepts of the Web as URI and media types. CoAP, like HTTP provides the following four methods for resources' manipulation: (i) GET, to retrieve a representation of the resource identified by the request URI; (ii) POST, to request that the representation enclosed in the request is processed; (iii) PUT, to request that the resource identified by the URI is updated or created with the transmitted representation; (iv) DELETE, to request that the resource identified by the specified URI is deleted.

In addition to the HTTP features, it offers a built-in mechanism for the resources discovery, it supports the IP multicast, and it natively provides a server-push model and an asynchronous exchange of messages. It also has a small-size header in order to be used on low-power networks like 6LoWPAN [21] over IEEE 802.15.4 [22]. It can also run on top of proprietary networks that are connected to IPv6 Internet. Moreover, it bases the communication on the UDP for reducing the communication costs.

CoAP has been standardized as RFC 7252 since June 2014 [7].

III. Architectural Design

In Fig. 1, the overall system architecture is reported. Each of the service stations is provided with a Kiosk client connected to the Internet and a variable number of charging stations (i.e., 4). The Kiosk is a self-service unattended client by which the user can autonomously swipe his/her credit card and choose the station to enable. An HTTPS connection is

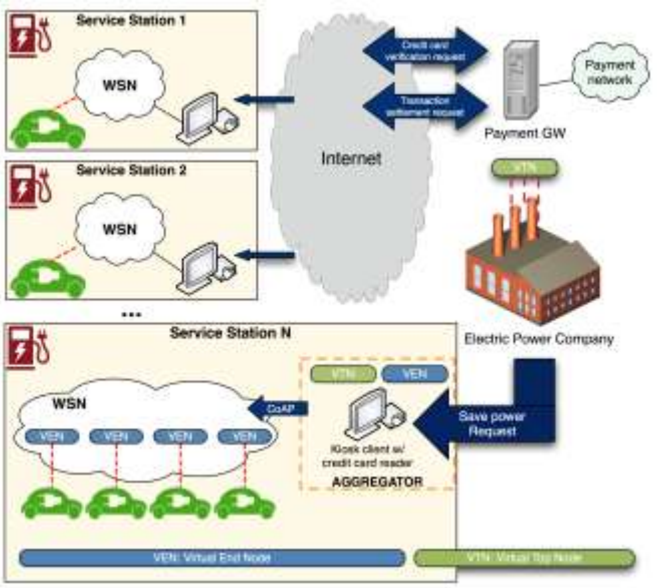

Fig. 1. The proposed system architecture. Each service station is composed of a Kiosk client equipped with a card reader.

needed to interact with a Payment Gateway (GW) through secure SOAP Web services. The Payment GW is able to decrypt the masked data incoming from the card reader and forward it to the payment network (e.g. Visa, Mastercard).

The Electric Power Company bundles a Demand Response Automation Server (DRAS), which is able to address all the stations that are absorbing current at any time. The Kiosks constantly update the DRAS by the means an HTTP link. From this point of view, the kiosks play the role of VEN in our OpenADR implementation. Notification messages (i.e., DR signals) are sent back to the Kiosks when a need to reduce or invert the electric current flow from the grid to the PEV is addressed.

The Electric Power Company also bundles a VTN, which is able to detect critical conditions in the grid and send adequate requests to the service stations.

The Kiosk is also a VTN by the side of the service station, because it is the agent driving the charging stations (VEN) behavior. The use of CoAP over the 6LowPAN protocol in the edge portions of an OpenADR architecture enables a resource-based control of the charging stations following the REST paradigm. Three REST resources are defined in each charging station: (i) the 'switch', that is used to torn on or off the station; (ii) the 'regulator', used to shape the current flow after a DR message is received; and (iii) the 'current', an observable resource that tells the instant current flowing from the charging station. Let us observe that further resources can be simply made available as soon as they will be needed in the future. Moreover, such resources are hidden from outside the service station domain. We made this choice to prevent security issues.

CoAP resources can be controlled only by CoAP messages. To this aim, the Kiosk embeds an HC proxy, that is a two-way HTTP-CoAP cross-protocol proxy, which translates HTTP requests from the Kiosk to the corresponding CoAP requests for the station. The proxy is able to handle GET requests, POST requests, and WebSocket (WS) connections incoming from the Kiosk HTTP client. GETs and POSTs are used to respectively fetch or update the 'switch' and 'regulator' resources status. A WS allows the WSN node to establish a persistent connection with the Kiosk in order to receive continuous data stream from the node itself. In our case, a WS is used to monitor observable CoAP resources (i.e. the 'current').

The interactions among the Kiosk, the HC proxy and a charging station for the 'switch' resource, identified by the URI https://<hostname>(:port)/OpenADR2/CoAP/switch, is reported in Fig. 2. In particular, the request involves the following interactions:

1. The Kiosk sends an HTTP POST request to the HC proxy to update the 'switch' resource with the value 'ON';
2. The HC proxy receives the request, converts it into a CoAP CON POST request and forwards it to the charging station node by using IPv6;
3. The charging station replies to the CoAP request by an ACK message;
4. The charging station sends a CON (confirmable) containing a CoAP 2.04 message (resource status changed);
5. The HC proxy replies to the charging station by an ACK message;
6. The HC proxy replies to the Kiosk by a response

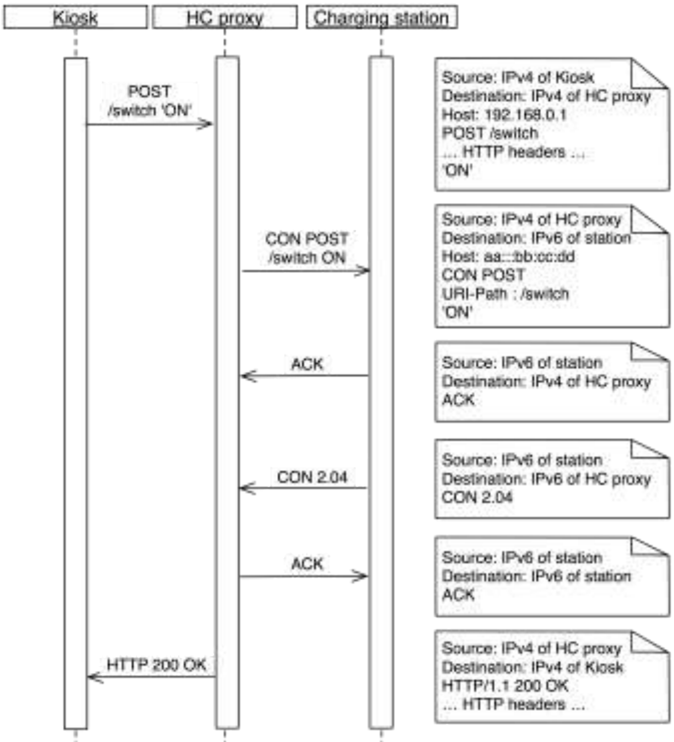

Fig. 2. UML sequence diagram modeling the interaction among the Kiosk, the HC proxy and the charging station for the updating of the 'switch' CoAP resource

message containing the code HTTP 200 within the header and the payload.

The Kiosk application retains the status of the stations seen by the HC proxy. To this aim, the Kiosk application instances a Station object for each charging station. A station can lie in one of the following states: (i) "initialized", when the Kiosk just started; (ii) "ready", when the Kiosk successfully connected to the station; station; (iii) "enabled" when the station has been enabled for being used by the user (i.e. a POST message containing the payload 'ON' has been sent to the station's switch 'resource'); (iv) "charging", when the user has plugged in the PEV to the station.

The order by which these states are activated follows the UML state machine diagram depicted in Fig. 3. The functions connect(), enable(), activate() and disable() represent state transitions between two states. We did not foresee a function connecting the charging state (source) and the enabled state (target) for security purposes. In fact, once the user finishes charging his PEV, the station must be re-enabled, i.e. the user has to pay before using it again.

IV. VALIDATION

In this section, the procedure adopted to validate the prototype implementation of a DR program for PEVs charging stations based on the 6LowPAN WSN technology and CoAP transport protocol is described. The proposed scenario meets the real-time requirements of charging stations services.

A. Test environment

The test environment includes five MB851 [11] WSN boards developed by ST Microelectronics, equipped with a 32-bit ARM® Cortex™-M3 MCU operating up to 24 MHz and embedding 16-KB RAM and 256-KB eFlash ROM. The board integrates a 2.4 GHz wireless transceiver compliant with the IEEE 802.15.4 standard. The MCU is equipped with a temperature sensor, MEMS, an infrared led and 2 light leds. In particular, the temperature sensor has been used to simulate the current metering. Furthermore, the operating system (OS) chosen to develop the firmware for the WSN node is the Contiki OS [12]. It is a open-source operating system targeted to small microcontroller architectures and developed by the Swedish Institute of Computer Science. In the considered scenario, four WSN boards have been used as CoAP servers able to reply to CoAP requests coming from HC proxy and one board has been used as border router, connected to the HC proxy via an USB cable.

The HC proxy was developed in Java 7 by using the Jersey framework [13] and the Californium (Cf) CoAP framework

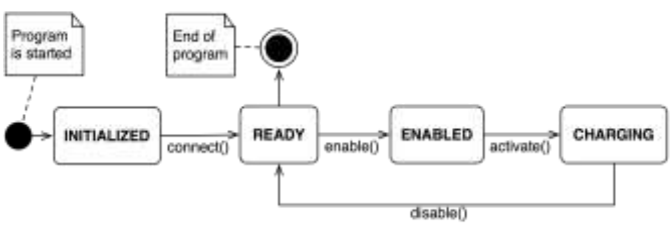

Fig. 3. UML state machine diagram of the station's lifecycle

[14]. The proxy runs as a Web application but it may be modified to run also as standalone Java application. It was deployed on an Apache Tomcat 7 servlet container.

The Kiosk application was developed using the .NET framework v4.5 using the C# programming language. SQL Server 2008 R2 has been used as the application's DBMS. The Kiosk application was installed on a Sahara NetSlate® a510 tablet computer, equipped with a 12.1" touch screen, an Intel 1.8GHz Atom processor, and 4GB memory. The Kiosk's operating system is Windows 7 Professional. This model was selected also because of its good performances in harsh environments (operating temperature range: 41°F ~ 95°F, operating humidity range: 20% ~ 80%).

The Kiosk tablet is also equipped with a Magtek Magnetic Stripe Reader (MSR) expansion module with encrypting heads. It was configured to work in keyboard emulation mode, so no special libraries are needed to talk with the MSR.

The Kiosk application also embeds the HTTP client needed to talk with the HC proxy. To this aim, the WebSocket4Net libraries [15] were used.

B. Proof-of-concept

We considered a particular target service as proof-of-concept of the correctness of the proposed solution: the use of PEVs as energy buffers to support the grid during peak demand periods. The goal is therefore to demonstrate that in such a situation the system is able to reverse the current flow of those assets (V2G) in order to avoid a potential collapse of the grid. The case study depicts the scene of some clouds that overshadow the photovoltaic panels of a building during a hot summer afternoon, as summarized in Fig. 4. Here we report the main steps:

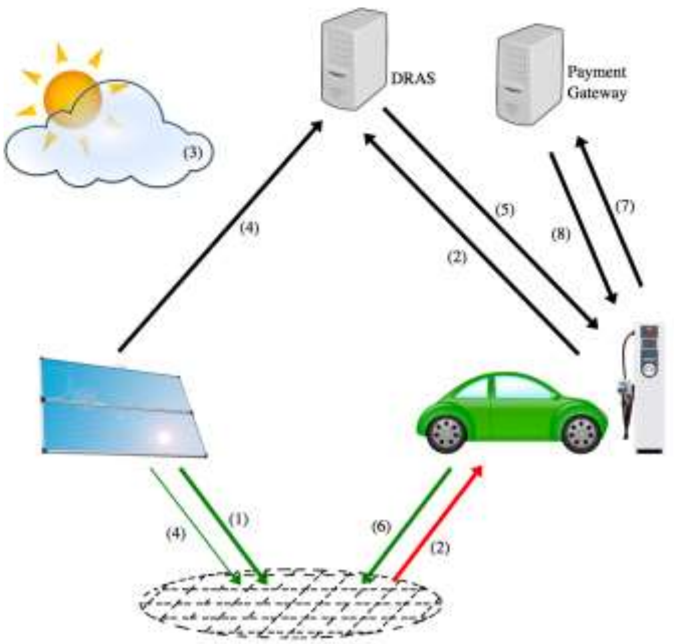

Fig. 4. Validation scenario: the use of PEVs as energy buffers to support the grid during peak demand periods. In such a situation the system is able to exercise a control over the loads in order to avoid a potential collapse of the

1) *The SG is in an equilibrium state*
   The photovoltaic panels are generating enough energy to support the overall load of the grid.
2) *A user asks for a service to the SG*
   A user asks for his/her PEV to be charged. Using the Kiosk application, swipe his/her credit card and it get authorized by the payment GW. So, the PEV begins to absorb energy from the grid (G2V mode) by the means of a PEV charging station. This fact is communicated to the DRAS server, which for example evaluates that a control procedure is at the moment not needed.
3) *A cloud overshadows the photovoltaic panels*
4) *The SG is not in an equilibrium state anymore*
   The photovoltaic panels can now generate less energy than before. This fact is again communicated to the DRAS, which evaluates that a control procedure is at this time needed, because the overall load of the grid falls below a pre-defined threshold.
5) *The new SG configuration is computed*
   The DRAS computes the optimized load variations. Then, it sends the load control DR messages to each controllable node of the grid.
6) *The new SG configuration is applied*
   The PEV charging station receives the corresponding DR message and inverts the current flow from G2V to V2G, therefore becoming a generator for the grid.
7) *The service is completed*
   The service finishes when the PEV is totally charged or when the user decides to stop the charging for some reason. Then the Kiosk computes the service price considering only the amount of energy actually given to the PEV. So the Kiosk contacts the payment GW in order to complete the payment procedure.
8) *The payment is completed*
   Finally, the payment gateway notifies the exit status of the procedure.

Fig. 5 and Fig. 6 report the details of the validation scenario. The green LED powered on the left side of Fig. 5 indicates that the 'switch' resource has status 'ON', hence the station is supplying the service (step 2 in the evaluation scenario). The red LED powered on the right side of Fig. 5 indicates that the 'regulator' resource has been updated so the charging station is working in V2G mode (step 6 in the evaluation scenario).

Fig. 6 shows some relevant screenshots of the Kiosk application. In Fig. 6.a the user can decide whether to charge his PEV by credit card or by an existing online account. In Fig. 6.b the user can choose a free (not already occupied) charging station to use. Then, after the credit card validation, the service starts in G2V mode, and the charging status is shown (Fig. 6.c). Later, the DRAS asks the PEV charging station to invert the current flow, so the updated charging status is shown to the user (Fig. 6.d). Notice the IPv6 station address appears in the 'Station ID' field.

Fig. 5. A MB851 WSN board installed in each charging station. On the left, the station is enabled and ready for charging (green LED is on). On the right, the station is in V2G mode, that is current flow is controlled (shaped or inverted) (red LED is on).

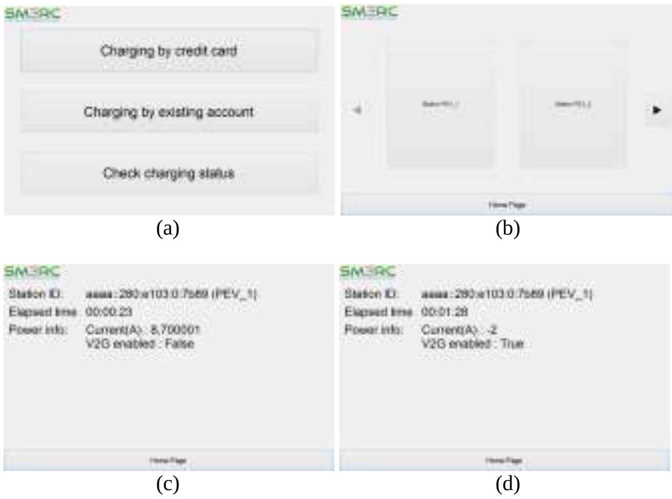

Fig. 6. Kiosk screenshots. In (a) the user can decide whether to charge his PEV by credit card or by an existing online account. In (b) the user chooses a free charging station to use. The charging status is shown to the user (c). In (d), V2G is enabled and charging status is updated.

V. Conclusions

In this paper we have presented an OpenADR implementation based on the 6LowPAN WSN technology and the CoAP transport protocol. Our publish/subscribe architecture implements the OpenADR two-way signaling systems, providing the servers, which publish information (Virtual End Nodes or VENs) to the automated clients, which subscribe the information (Virtual Top Nodes, or VTNs). The adoption of CoAP as an alternative transport protocols (instead of the suggested HTTP and XMPP) perfectly complies with the reduced computation capabilities of the small and cheap devices that will monitor the energy grid of the future following the Internet of Things paradigm.

In order to proof the reliability and suitability of CoAP in an OpenADR architecture, we considered the process of a user charging his/her Plug-in Hybrid Electric Vehicle (PHEV). We designed and developed a self-service Kiosk that can accept credit cards and control the charging processes. The Kiosk is sensible to DR signals incoming from the Demand Response Automation Server (DRAS) defined in our OpenADR implementation. Basing on the specific DR signal received, the Kiosk will ask one or more charging stations to limit or invert

(V2G) the current flow during the charging process. Charging stations are connected to the Kiosk using the 6LowPAN WSN technology and can be referenced using their IPv6 univocal address. We used CoAP as transport mechanism to control the stations' behavior. The use of CoAP over the 6LowPAN protocol enables a resource-based control of the charging stations following the REST paradigm, hence allowing a more natural and human-readable definition of DR programs. We defined three resources for each charging station: (i) the 'switch', that is used to torn on or off the station; (ii) the 'regulator', used to shape the current flow after a DR message is received; (iii) the 'current', an observable resource that tells the instant current flowing from the charging station.

The proposed OpenADR implementation also includes a market integration module by which it is possible to decrypt the masked data incoming from the card reader and forward it to the payment network (e.g. Visa, Mastercard). The system dynamically computes the service price basing on the amount of current the PEV gave back to the SG (V2G) during the charging process.

The work presented in this paper is part of a bigger research effort, which will use semantic technologies to detect and react to critical grid conditions. Next research efforts will be spent to integrate the presented work in such Internet of Energy semantic architecture.

ACKNOWLEDGMENTS

This work has been sponsored in part by a grant from the LADWP/DOE fund 20699 & 20686, (Smart Grid Regional Demonstration Project). This material is based upon work supported by the United States Department of Energy under Award Number DE-OE0000192 and the Los Angeles Department of Water and Power. Neither the United States Government nor any agency thereof, the Los Angeles Department of Water and Power, nor any of their employees make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey", Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] Smart Energy Profile 2. Available: http://www.zigbee.org/

[3] OpenADR Alliance, 2013, OpenADR 2.0 Specifications – Profile B.

[4] XMPP (Extensible Messaging and Presence Protocol), Available: http://www.ietf.org/rfc/rfc3920.txt, http://xmpp.org/

[5] Yide Liu, "Wireless Sensor Network Applications in Smart Grid: Recent Trends and Challenges," International Journal of Distributed Sensor Networks, vol. 2012, Article ID 492819. doi:10.1155/2012/492819.

[6] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," RFC 4919, August 2007

[7] The Constrained Application Protocol (CoAP), RFC 7252, June 2014.

[8] N. Bui, A.P. Castellani, P. Casari, M. Zorzi, "The Internet of Energy: a web-enabled smart grid system", IEEE Network, Vol. 26 , No.4, pp. 39-45, 2012.

[9] S. Hadley and A. Tsvetkova, "Potential impacts of plug-in hybrid electric vehicles on regional power generation," Oak Ridge National Laboratory, Tech. Rep., 2008.

[10] W. Kempton and J. Tomic, "Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy", J. Power Sources, vol. 144, no. 1, pp. 280-294, Jun. 2005.

[11] MB851 User Manual. Available: http://www.st.com.

[12] A. Dunkels, B. Gronvall, and T. Voigt, " Contiki – a lightweight and flexible operating system for tiny networked sensors", IEEE Workshop on Embedded Networked Sensors, Nov. 2004.

[13] Jersey RESTFull framework. Available: http://jersey.java.net/

[14] Californium (Cf) CoAP framework. Available: http://people.inf.ethz.ch/mkovatsc/californium.php

[15] WebSocket4Net libraries - http://websocket4net.codeplex.com/.