ZigBee Wireless Sensor Network with Relative Proximity Estimation Capability

ANTHONY S. DEESE, JULIAN DAUM
Department of Electrical and Computer Engineering
The College of New Jersey (TCNJ)
2000 Pennington Road Pennington, New Jersey 08628
UNITED STATES OF AMERICA (USA)
deesea@tcnj.edu, http://www.anthonydeese.com

Abstract: – The ZigBee/IEEE 802.15.4 Standard is very useful for the design of low-power wireless sensors. However, these devices lack the ability to determine their location with respect to other nodes within the wireless network. Such information may be beneficial for applications like smart home automation. The objective of this work is to develop a set of wireless sensor/actuator modules that utilize integrator-based sensors to collect measurement data between transmission bursts as well as received signal strength indicators (RSSIs) to perform triangulation-based proximity estimation with respect to other ZigBee modules. This triangulation technique employs an innovative method to assign and reassign the parent/router of an end node through network configuration and forced radio resets.

Key-Words: - XBee, ZigBee, wireless networks, home automation, sensors

1 Introduction

The ZigBee/IEEE 802.15.4 Standard is very useful for the design of low-power wireless sensors. However, these devices lack the ability to determine their location with respect to other nodes within the wireless network. Such information may be beneficial for applications like smart load automation and demand response (DR) in power systems. Demand response may be used to increase or decrease the consumption of individual loads based on actual or projected electricity costs [1, 2]. The objective is to operate non-essential loads that perform time insensitive tasks at times (e.g. late night) when electricity prices are low, providing customers an opportunity to reduce their average billing rate.

Wireless devices with the ability to actuate electric loads as well as observe their behaviour, like smartplugs, are essential to effective demand response initiatives. This paper is motivated by potential applications to DR.

2 Problem Statement

There are many commercially-available smartplugs with the ability to sense the environments and actuate electric loads wirelessly. However, most lack the ability to: 1) measure both proportional and integral values and 2) estimate their relative location to other devices within the wireless network. Furthermore, commercial devices often employ proprietary software that prevents users from accessing their basic hardware functionality and implementing new ideas or algorithms. The authors of this work have made their software available open-source and downloadable here: http://bit.ly/2y97GFi.

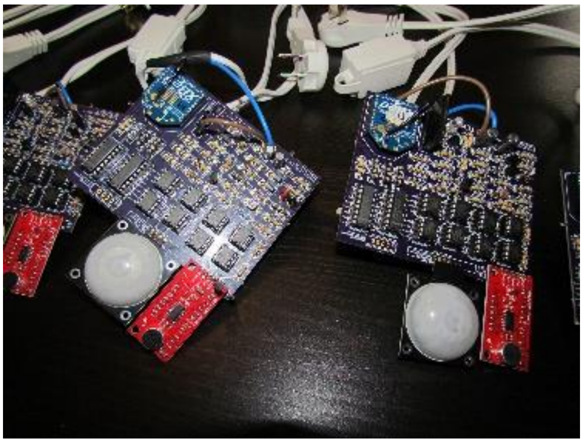


Fig. 1: Completed ZigBee-Based Sensor-Actuator Module Prototype with Motion Sensor (Large White Globe) and Sound Detector (Red Board) Shown.

3 Objectives

The objective of this work is to develop a set of wireless sensor/actuator modules that utilize integrator-based sensors to more efficiently store measurement data between transmission bursts as well as received signal strength indicators (RSSIs) to perform triangulation-based proximity estimation with respect to other ZigBee modules. This triangulation technique employs an innovative method to assign and reassign the parent/router of an end node through network configuration and forced radio resets. Refer to Fig. 3. The author's hardware relies on the ZigBee Series 2 wireless communication protocol; their software is developed in Python using the Digi Application Programming Interface (API) [6, 7]. The authors hope that their work will provide other researchers with the tools required to further their work in optimal load control as it related to demand response.

4 Proximity Estimation Algorithm

The ZigBee/IEEE 802.15.4 Standard traditionally relies on a star network structure, like that shown in Fig. 2 below [8, 9]. This work addresses start-type structures only, although newer technologies do allow meshing [6, 10, 7]. To understand the proximity estimation algorithm, there are several key ZigBee S2 module parameters that must be introduced.

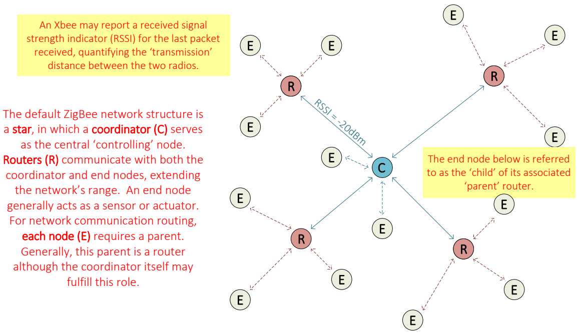


Fig. 2: Basic ZigBee Star Structure

4.1 Node Join Time (NJ) Parameter

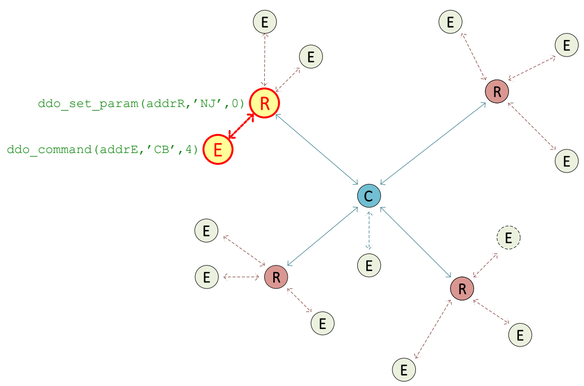
All ZigBee radios have a parameter called node join time (NJ). This parameter defines the length of time following its initialization that a parent node allows children to join. The user may define a maximum value of $NJ = 255$, allowing children to join at any time. She may also define a minimum value of $NJ = 0$, preventing this parent from accepting any new children while maintaining any pre-existing parent/child relationship. This NJ parameter is essential to manipulation of the communication network structure, as demonstrated by the text below [6, 7].

4.2 Software Commissioning Button (CB)

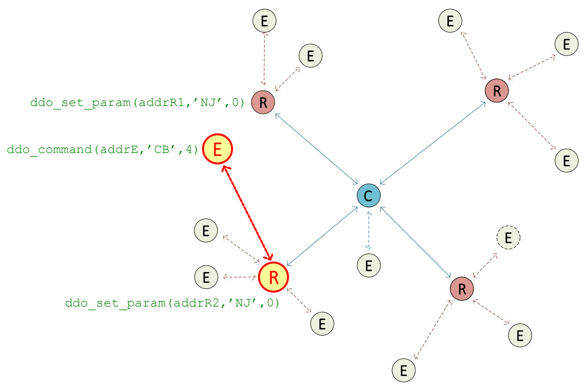
The commissioning button (CB) command within the Digi Python API is powerful, emulating the effect of pressing the physical commissioning button on a Series 2 radio. Two useful variations exist. One is ddo_command(addressX,'CB',2). This command emulates the behaviour of physically pressing the commissioning button two times, forcing all radios within a network to search for new parent nodes and migrate should a better parent exist. Note that a better parent is one with a greater received signal strength indicator ($RSSI$). For one minute this command overrides any otherwise-defined NJ parameters, opening any and all nodes to accept children. A second useful variation is ddo_command(addressX,'CB',4). This command emulates the behaviour of pressing the commissioning button four times, forcing the radio at addressX to perform a software reset. Once reset is complete, this radio (and this radio only) will search for the best parent available based on $RSSI$ [6, 7].

4.3 Forced Network Reconfiguration

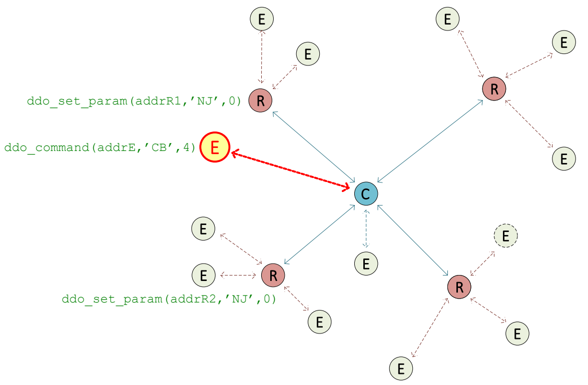
Assume that a child node is currently associated with its optimal parent, that with the greatest $RSSI$. How can a network manager or developer determine what that child's second-best option is? First, the developer must set the node join time ($NJ = 0$) for the current parent to zero. Second, she must perform a software reset of this child itself via commissioning button ($CB = 4$). Once the reset is complete, this child will attempt to associate with its previous parent. However, it will not be able to because of that parent's newly defined NJ value. The child will then search for the best available parent and associate itself with this radio; the new parent is its second-best option.



(a)



(b)



(c)

Fig. 3: Authors' User-Derive Network Reconfiguration Technique Determining Three Potential Parents for Highlighted End-Node. (a) First Parent. (b) Second Parent. (c) Third parent. Note that the first topology shows the end node associated with its best parent while the third and final topology shows the end node associated with the coordinator (its third-choice parent).

The authors developed a forced network reconfiguration technique for the ZigBee communication protocol that allows users to determine what potential parents exist for an end node. The ability to connect an end node with other routers or potential parents as well as quantify their mutual *RSSI* values is essential to the proximity estimation algorithm. Fig. 3 illustrates how the highlighted and bolded end node may be connected with three different potential parents: two different routers and the network coordinator node. This is done by systematically setting the NJ value for the current parent to zero and resetting the end node via remote commissioning button actuation. To understand the authors' network reconfiguration method, it is important to also understand what happens when a module joins or leaves the ZigBee network because of software reset or power loss:

- *When Coordinator Joins / Leaves* – This cannot happen. A network can have one and only one coordinator. If the coordinator leaves, the network will cease to function [7];
- *When a Router Joins* – It simply registers itself with the coordinator as an available router. The coordinator does not ask end nodes to re-evaluate their choice of parent, even though the newest router may provide a stronger communication pathway for end node *X* than its current parent. The result is end nodes that, at least initially, may not be associated with the best parent [7];
- *When an End Node Joins* – It searches for the best potential parent based on *RSSI*. The result is an end node that, initially, is associated with the best parent [7];
- *When a Router Leaves* – Any orphaned children must re-join the network and find a new best potential parent. These end nodes will find the best remaining parent, although it may not provide the same communication path strength [7];
- *When an End Node Leaves* – it does not affect any other nodes, except for the coordinator that will remove it from the available radio node list [7].

4.3 Received Signal Strength Indicator

To estimate radios' locations with respect to one another, the developer must have access to a two-dimensional array like that shown in Fig. 4 below. Information regarding the *RSSI* between multiple sets of parents and children will be used [9, 8].

#	Parent Addr	Child Addr	RSSI

Fig. 4: Sample 2D Array to Estimate Node Locations and Proximities

Before diving into an algorithm, one must understand exactly what the *RSSI* value returned by a radio means. It is a quantification of signal strength for the last message received. Because end nodes receive messages from their parent router or coordinator only, there is no question what the returned *RSSI* value means. For routers, however, *RSSI* may describe several transmission paths including coordinator to router, router to child A, router to child B, etc… For this reason, the user is advised to send a token message, such as a firmware version request, to the target child to ensure that the final message received is from its parent. This is more important for routers than end nodes [7].

The best way to understand how *RSSI* is quantified and recorded by radios in Python is to examine the function funQuantifyRSSIwParent within the library downloadable here: http://bit.ly/2y97GFi.

5 Proximity Estimation

5.1 Test System and Data Collection

To test the proximity estimation algorithm defined above, the authors used the residential test system shown in Fig. 5.

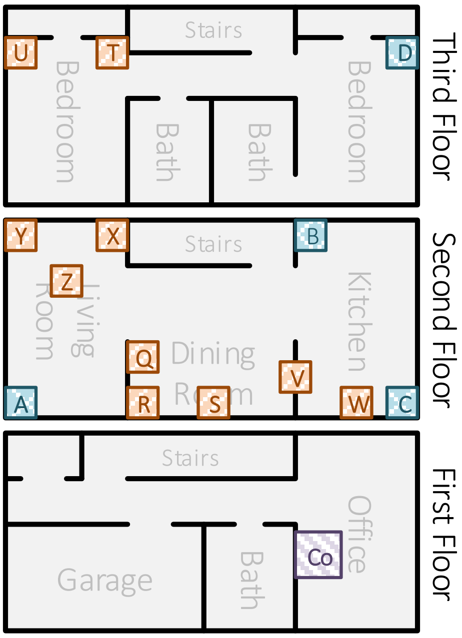


Fig. 5: Residential Test System for Proximity Estimation Test Showing the XBee Coordinator (Co), Routers (A – D), and End Nodes / Sensors (Q – Z).

The $1600ft^2$ space is divided into three floors, each approximately $16ft$ wide and $35ft$ long. It contains one XBee coordinator module (Co), four routers ($A - D$), and ten end nodes serving as sensors ($Q - Z$). The user-driven network reconfiguration technique described Fig. 3 was used to measure the $RSSI$ between each end node and three potential parents. The transmission power loss (P_{ik}^{TLoss}) between router i and end node k is then calculated as shown in (1), as the difference between the transmission power of this router (P_i^{Xmit}) and $RSSI$ observed by the corresponding end node (P_k^{RSSI}) in dBm. All $RSSI$ measurements are made in dBm by the XBee hardware. Note that transmission power may change with the radio's power level (PL) and power mode (PM) settings as well as the specific model. Also note that the S2 PRO transmits with a higher maximum power than the standard S2 model [8, 9].

$$P_{ik}^{TLoss} = P_i^{Xmit} - P_k^{RSSI} \; in \; dBm \qquad (1)$$

The result is a table containing thirty P_{ik}^{TLoss} values, three for each of the ten end nodes.

5.2 Results

The results of this test are displayed in Table 1. Each element of this table represents a value referred to in this paper as the average absolute difference in transmission losses between end nodes i and k for all common potential parents ($\Delta P_{ik}^{AvgLoss}$). For example, assume that the user-driven network reconfiguration technique shown in Fig. 3 observes the transmission losses (in dBm) shown below in (2) between end node Q and routers A, B, and D.

$$P_{AQ}^{TLoss} = 5, P_{BQ}^{TLoss} = 7, P_{DQ}^{TLoss} = 13dBm \qquad (2)$$

It also observes the transmission losses shown below in (3) between end node R and routers A, B, and C.

$$P_{AR}^{TLoss} = 7, P_{BR}^{TLoss} = 4, P_{CQ}^{TLoss} = 10dBm \qquad (3)$$

The end nodes Q and R share two common potential parents (A and B). As such, their proximity to one another may be defined by the value $\Delta P_{QR}^{AvgLoss}$ as defined below in (4) with a numerical result of $2.5dBm$.

$$\Delta P_{QR}^{AvgLoss} = \frac{1}{2}\left(\left|P_{AQ}^{TLoss} - P_{AR}^{TLoss}\right| + \left|P_{BQ}^{TLoss} - P_{BR}^{TLoss}\right|\right) \qquad (4)$$

Table 1 presents the $\Delta P_{ik}^{AvgLoss}$ values for every combination of end nodes $Q - Z$ shown in Fig. 5, where all end nodes share at least two potential parents. The results demonstrate how such value may be used to predict the relative proximities of nodes to one another.

For example, there are four clusters of end nodes located throughout the residential space: 1) QRS, 2) TU, 3) VW, and 4) XYZ. Table 1 presents an average $\Delta P_{ik}^{AvgLoss}$ value of $5.5dBm$ for all nodes within the ZigBee network. The average value for nodes within a given cluster is only $0.65dBm$. The two end nodes located farthest from one another (U and W) yield a $\Delta P_{UW}^{AvgLoss}$ value of $9.33dBm$, almost double the network average. A comparison of Table 1 and Fig. 5 reveals a strong relationship between $\Delta P_{ik}^{AvgLoss}$ and distance between two nodes. There are outliers like $\Delta P_{QX}^{AvgLoss}$, an unusually large value given the short distance between radios Q and X. The authors hypothesize that this may be attributed to the placement of the end nodes, both behind rather thick walls.

Table 1: $\Delta P_{ik}^{AvgLoss}$ Values for All Combinations of End Nodes $Q - Z$ in dBm. Note that level of shading increases with cell values. Note that row and column lines are essential to illustrate mapping.

	Q	R	S	T	U	V	W	X	Y	Z
Q	X	0.33	1.00	7.67	7.33	5.67	3.00	14.33	8.00	7.67
R	X	X	0.67	6.33	8.00	6.00	7.00	9.00	6.33	8.33
S	X	X	X	7.00	8.33	4.33	2.33	8.00	7.67	8.67
T	X	X	X	X	0.67	9.00	9.33	4.33	6.00	4.67
U	X	X	X	X	X	4.33	8.66	4.67	5.00	5.00
V	X	X	X	X	X	X	1.00	12.00	7.00	6.33
W	X	X	X	X	X	X	X	6.33	6.67	6.67
X	X	X	X	X	X	X	X	X	0.00	0.00
Y	X	X	X	X	X	X	X	X	X	1.33
Z	X	X	X	X	X	X	X	X	X	X

6 Applications

The main contribution of the authors in this work is the algorithm described in Fig. 3 and its potential to facilitate clustering of radios and rough estimation of the relative proximities. The authors do not attempt to quantify the numerical distance between two radios from their relative $\Delta P_{ik}^{AvgLoss}$ value as their initial attempts to do so proved unsuccessful. This type of method is limited by the need to properly model any obstacles and walls between two nodes.

6.1 Module Hardware

In this section, the authors briefly discuss other innovative aspects of their wireless sensor/actuator design as shown in Fig. 1 [11]. This includes the ability to:

- *Measure Environmental Values* – The authors' wireless sensor/actuator module measures instantaneous values for ambient light and load current draw as well as integral values for detected motion and sound. Low-voltage operational amplifiers are used for buffering, signal conditioning, as well as integration. A BJT is used to apply a short across the integrating capacitors for reset. These transistors may be actuated wirelessly to restart the integrating window and control its duration. For most testing, the authors employed an integrator time-constant of $\tau = 10ms$. This provided sufficient sensitivity while minimizing the effect of saturation.
- *Acquire Eight Analogue Signals* –The ZigBee S2 module provides designers only four ADCs for analogue signal measurement. The authors employ a set of multiplexers to double the number of potential analogue measurements to eight [10, 7].
- *Actuate Load Status* – The hardware employs a $120V/8A$ solid-state relay to control the flow of power to a connected load. The control signal itself is generated by the ZigBee module and supplied to relay control circuit based on the appropriate BJT. Like many commercially-available units, a current rating of $8A$ is used, as opposed to the full $15A$ available from the wall socket. This is to improve safety and prevent its use for control of larger machines like washers and dryers.
- *Communicate Wirelessly* – The design is compatible with several ZigBee RF modules, including the S2 and S2 PRO. Obviously, these modules provide wireless communication capability. Channels DIO4 and DIO5 are utilized actuate the BJTs used for integrator reset and AC relay control, respectively [7].

An overview of the sensor/actuator module design is provided in Fig. 6 below.

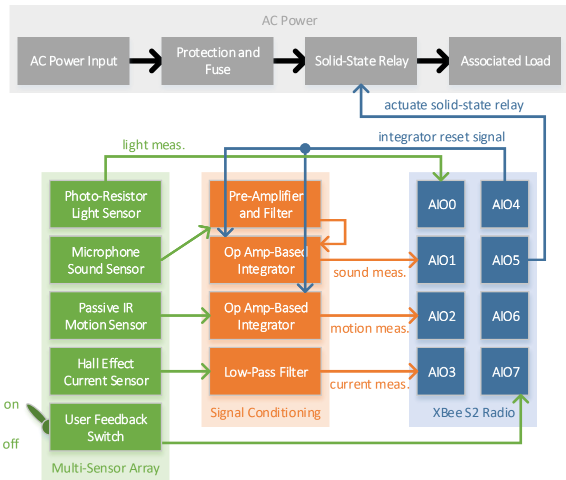


Fig. 6: Overview of Sensor-Actuator Module Design.

6.2 Calibration and Measurement Error

One important part of the sensor-actuator design process was to quantify any errors associated with data collected from the ZigBee module [12, 7]. This one was done by comparing voltage measurements acquired remotely, via software, from the S2 to those acquired from a standard two-channel oscilloscope, calibrated within the last year. Fig. 7 presents the results of this test. For small values, the S2 exhibits a measurement error more than 10% on certain channels. This error may be dramatically reduced through calibration, adding to all measurements made by the S2 and offset of $10.2mV$. Although the exact offset required differs from ZigBee module to module, the offset of $10.2mV$ does reduce measurement error for all S2 radios the authors examined. This offset was embedded within the system software.

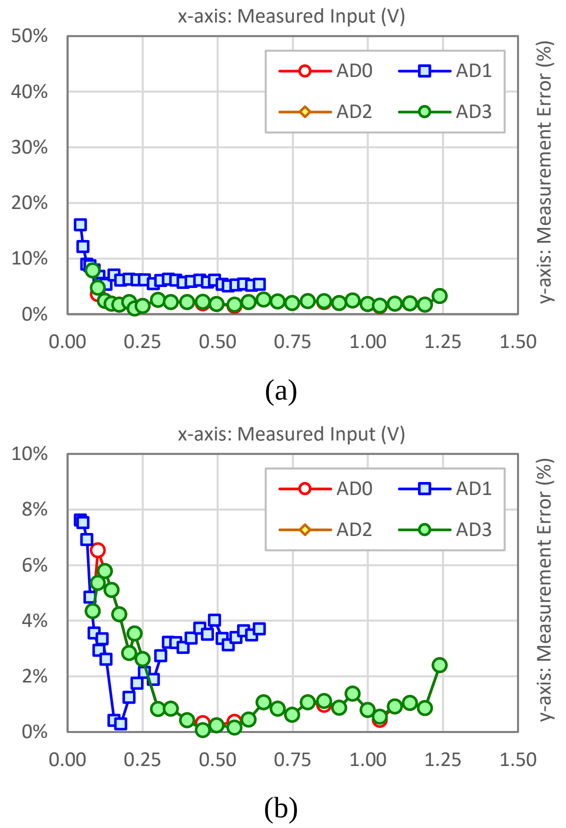


Fig. 7: (a) Measurement Error Between XBee / Python and Oscilloscope Measurements Without Calibration; (b) Measurement Error with Offset 0.0102 Adjustment (Calibration)

7 Design Notes

The ZigBee S2 module requires a $3.3V_{DC}$ supply, that may be regulated from a standard $5V_{DC}$ supply. The authors' testing showed that the radio injects significant harmonics into both the $3.3V_{DC}$ and $5V_{DC}$ supplies, seriously affecting board performance. Such harmonics are especially detrimental to sensors. The authors suggest that: 1) the sensors be placed near the $5V_{DC}$ power supply and 2) that each sensor be equipped with its own voltage regulator [10, 7].

8 Conclusion

This work presents the authors' work in ZigBee wireless sensor development, with focus on the utilization of integrator-based sensors to more efficiently store measurement data between transmission bursts as well as $RSSIs$ to perform triangulation-based estimation of the proximity between radios. The results presented in Section 4 reinforce the authors' hypothesis that rough proximity estimation for node clustering is possible. The relationship between $\Delta P_{ik}^{AvgLoss}$ and physical distance, however, is not well suited to quantify the latter value.

As discussed in the introduction, one application of a wireless sensor/actuator hardware like that described here is demand response (DR). Design concepts from consumer electronics and the Internet of Things (IoT) must be employed to implement effective DR. A significant percentage of power consuming devices within a system must have some data acquisition, communication, and remote actuation capability [1]. Unfortunately, this is not the case. The hardware described in Fig. 1 may act as an interface between the loads within a residential or commercial building and the external smart grid, providing the basic capabilities required for participation of these devices in demand response initiatives. This interaction will be the focus of the authors' future research.

References:

[1] D.-L. Lee and C.-L. Hsu, "An Implementation of Intelligent Energy Saving System," in IEEE/ACM International Conference on Green Computing and Communications (GreenCOM), Sichuan, China, 2011.

[2] Q. Luo, A. H.-Y. Lam, D. Wang, D. Wai-Tin Chan, Y. Peng and X. Peng, "Demo Abstract: Towards a Wireless Building Management System with Minimum Change to Building Protocols," in 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems (ICCPS), Beijing, China, 2012.

[3] A. H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober and A. Leon-Garcia, "Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid," IEEE Transactions on Smart Grid, vol. 1, no. 3, pp. 320-331, 2010.

[4] M. R. Kleinberg, K. Miu and H. D. Chiang, "Improving Service Restoration of Power Distribution Systems Through Load Curtailment of In-Service Customers," IEEE Transactions on Power Systems, vol. 26, no. 3, pp. 1110-1117, 2011.

[5] P. Romanos, N. Hatziargyriou, J. Schmid and J. Mutale, "Single Agents in Smart Grids," in 7th Mediterranean Conference and Exhibition on Power Generation, Transmission, Distribution and Energy Conversion (MedPower 2010), 2010.

[6] "Digi API Reference Guide," 1 April 2016. [Online]. Available: ftp://ftp1.digi.com/support/documentation/html/90001120/api_reference.html. [Accessed April 2016].
[7] R. Fauldi, Building Wireless Sensor Networks: With ZigBee, XBee, Arduino, and Processing, New York, NY: O'Reilly Publishing, 2010.
[8] A. S. Deese, B. Carrigan, E. Klein and E. Stein, "Automation of Residential Load in Power Distribution Systems with Focus on Demand Response," Institute of Engineering and Technology (IET) Research Journal of Generation, Transmission, and Distribution, vol. 7, no. 4, pp. 357-365, 2013.
[9] K. Gill, S. H. Yang, F. Yao and X. Lu, "A ZigBee-Based Home Automation System," IEEE Transactions on Consumer Electronics, vol. 55, no. 2, pp. 422-430, 2009.
[10] E. Spanò, L. Niccolini, S. D. Pascoli and G. Iannacconeluca, "Last-Meter Smart Grid Embedded in an Internet-of-Things Platform," IEEE Transactions on Smart Grid, vol. 15, no. 2, pp. 1109-1113, 2015.
[11] G. Goncalo and S. Helena, "Indoor Location System Using ZigBee Technology," in Third International Conference on Sensor Technologies and Applications (SENSORCOMM '09), Athens, Greece, 2009.
[12] L. Brás, M. Oliveira, N. B. d. Carvalho and P. Pinho, "Low Power Location Protocol Based on ZigBee Wireless Sensor Networks," in 2010 International Conference on Indoor Positions and Indoor Navigation, Zurich, Switzerland, 2010.
[13] Y. t. Lee, W. h. Hsiao, C. m. Huang and S. c. T. Chou, "An Integrated Cloud-Based Smart Home Management System with Community Heirarchy," IEEE Transactions on Consumer Electronics, vol. 62, no. 1, pp. 1-9, 2016.
[14] Z. Wen, D. O'Neill and H. Maei, "Optimal Demand Response Using Device-Based Reinforcement Learning," IEEE Transactions on Smart Grid, vol. 15, no. 2, pp. 37-47, 2015.
[15] B. Qela and H. T. Mouftah, "Observe, Learn, and Adapt (OLA) - An Algorithm for Energy Management in Smart Homes Using Wireless Sensors and Artificial Intelligence," IEEE Transactions on Smart Grid, vol. 3, no. 4, pp. 2262-2272, 2012.