TEM Journal. Volume 6, Issue 4, Pages 855-861, ISSN 2217-8309, DOI: 10.18421/TEM64-27, November 2017.

Smart Circuit Breaker Communication Infrastructure

Octavian Mihai MACHIDON [1], Radu Constantin ANDREI [2], Carmen GERIGAN [3]

[1]*Department of Electronics and Computers, Transilvania University of Brasov, Romania*

Abstract **– The expansion of the Internet of Things has fostered the development of smart technologies in fields such as power transmission and distribution systems (as is the Smart Grid) and also in regard to home automation (the Smart Home concept). This paper addresses the network communication infrastructure for a Smart Circuit Breaker system, a novel application at the edge of the two afore-mentioned systems (Smart Grid and Smart Home). Such a communication interface has high requirements from functionality, performance and security point of views, given the large amount of distributed connected elements and the real-time information transmission and system management. The paper describes the design and implementation of the data server, Web interface and the embedded networking capabilities of the smart circuit breakers, underlining the protocols and communication technologies used.**

Keywords **– smart grid, power system, circuit breaker, communication interface.**

1. Introduction

The electrical energy infrastructure’s physical security is a crucial element that must offer protection against risks such as supply interruption due to malfunctions or anomalies in the electrical power supply system. Effective protection measures are being developed to protect the infrastructure against such risks, especially given the very high incidence rates reported by studies. For example, it has been shown that in the Netherlands, 31% of all fires in household or industrial locations have electrical causes [1], and one the main causes for electric-related fires is the electric arc [2]. In this context, the effectiveness of the safety devices is vital, and one of the key protection elements in the case of the electrical grid is represented by circuit breakers.

Circuit breakers can be found in every home or office building around the world, as they represent one of the most basic pieces of power management hardware, playing a critical role to protect power equipment during faults [3]. They are the first device that electricity flows through when entering a home or building, and in today's world transcended by the Internet of Things, circuit breakers can be viewed as standing on the edge or border between Smart Grid and Smart Home acting as an actual gateway.

While the Smart Grid is being regarded as a smart distributed network for delivering electricity which stops at the consumer’s door (the smart meter), Smart Home is a technology targeting home automation by ensuring Internet connectivity of all the household appliances and control systems. The circuit breaker, given its location just beyond the power meter is an active element of protection for the appliances in the home, which makes it of vital importance in the Smart Grid – Smart Home synergy.

In this context, our efforts target the design and development of a Smart Circuit Breaker (SCB) capable of offering not only improved protection but also “smart” detection and management of grid faults. Becoming “smart” also means the circuit breaker is capable of enhanced communication over a network for data transmission and remote management.

The advantages of the proposed device are brought on by an improved protection in the case of faults through: high speed disconnection, automatic reconnection, data fusion (one device detects all events) and disconnection in the case of sensing voltage zero crossing which diminishes the generated perturbations. Last but not least, the data centralization infrastructure (all SCBs report their

DOI: 10.18421/TEM64-27
https://dx.doi.org/10.18421/TEM64-27

Corresponding author: Octavian Mihai Machidon, *Department of Electronics and Computers, Transilvania University of Brasov, Romania*
Email: octavian.machidon@unitbv.ro

Received: 03 September 2017
Accepted: 24 October 2017
Published: 27 November 2017

data to a server and the information is centralized in a database) allows for a real-time monitoring and reaction in the case of serious events and also generating statistics over time periods or at different locations. Remote user disconnection is also technically possible.

The communication infrastructure is the cornerstone of any Smart Grid-like distributed system, since it enables multiple, heterogeneous entities (smart devices, software, concentrators, etc.) to interact and exchange data [4]. It has been shown that having a reliable and pervasive communication infrastructure represents a vital issue in the operation and infrastructure of a Smart Grid [5][6].

In this paper, we describe in detail the communication infrastructure of the proposed system, on both ends: the networking capabilities and features embedded in each Smart Circuit Breaker and the Web Server infrastructure for centralizing the reported events. Also, the data flow and message structure is being presented.

The communication infrastructure described in this paper makes use of emerging web standards and the most widely adopted HTTP-based web services in order for the entire system to meet the interoperability and accessibility requirements of any Smart Grid - like distributed system, as established by the reference institutions in this field, namely the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C) [7]. Thus, the communication and data representation standards chosen foster a quick and easy integration of the designed system with any Internet-based technologies.

2. Design and Implementation Methodology

The architecture of the communication infrastructure we developed (shown in Figure 1.) is composed of a data concentrator, embedded networking capabilities in each SCB, and an online (Web-based) visualization and management interface.

Our SCB's functionality compared to a standard circuit breaker is different not only since it has on-board intelligence and Internet-connectivity, hence having enhanced communication capabilities and being remotely accessible, but it also provides improved safety features. It is capable of detecting and offering protection not only against overcurrent and short-circuit, nonetheless also against electric arcs [8], overvoltage and in the case of electrocution.

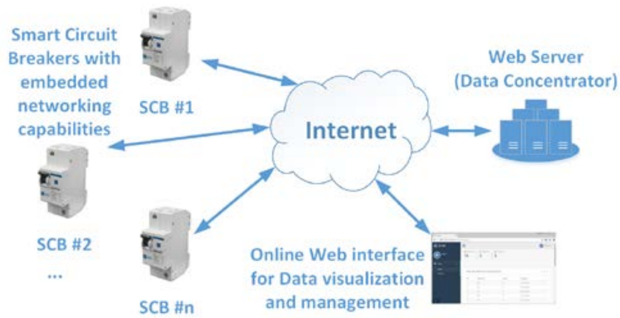

Figure 1. Communication infrastructure general architecture.

Each Smart Circuit Breaker contains an Arduino Uno R3 (ATMEGA328) microcontroller system, which acquires samples of voltage and load current, analyses them, and proceeds to load disconnection if the voltage or current values rise above the admissible levels. Ground leakage currents can also be detected since current values are acquired from both wires of the supply line by calculating their difference. The occurrence of an event will trigger load disconnection using a triac.

In order for these circuit breakers to become “smart” and thus have Internet connectivity for reporting their status and recorded events, each device embeds an Arduino Uno microcontroller platform connected to a networking shield, either Ethernet or Wi-Fi. Using one of these interfaces, the device is able to send an HTTP request to an URL that represents and API point of the data concentrator Web Server that handles incoming messages from all the SCBs installed at various locations.

The data concentrator for our Smart Circuit Breaker system is somewhat similar to the ones in advanced metering infrastructures (AMI), which collect electricity usage data from residential electricity meters, usually based on wireless or power line communication technologies [9]. In our case, the data concentrator is actually a Web Server providing and API (Application Programming Interface) used for communicating with the SCB devices by HTTP GET-POST requests that encapsulate data as JSON strings in the requests' body.

2.1 Web Server and Application

2.1.1 Server and Database

For implementing the data concentrator infrastructure, a Web Server has been implemented and deployed on a Linux machine in our department (running the http://etc.unitbv.ro domain). The implementation of this component has been accomplished using the Node.js framework and PostgreSQL database. Node.js has been chosen since it is an open-source server-side platform that fosters the rapid and easy development of Web apps integrating other Web technologies like HTML, CSS and AJAX. In the particular case of this application, one of the main advantages Node.js brings is the asynchronous operating mode, leading to an efficient handling of multiple connections from smart circuit breakers.

The Node.js Web Server has been deployed on a local port on the Linux machine which has been forwarded using Apache mod_proxy to an Internet address on the domain: http://etc.unitbv.ro/elsa .

For an easy data communication and transfer from the embedded devices (the Smart Circuit Breakers) to the database and forward on to the Web interface, a basic API has been implemented that is accessible by GET or POST requests to the URL http://etc.unitbv.ro/elsa/api/v1/reports . In both cases the data is encapsulated in the request's body as a JSON string.

When a SCB device wants to report an event to the server, it sends an HTTP POST request to the above-mentioned URL containing a JSON string with the following fields: device ID, timestamp, location and event. The Web Server handles such a request by taking the data fields from the JSON string and creating a new entry in the database with these values through a specific query as shown in Figure 2. If this operation has been successful, a corresponding message, also embodied in a JSON string, is being sent back to the reporting SCB device, otherwise a 500-status message is sent signalling that an error occurred.

```
app.post('/api/v1/reports', (req, res) => {
  const results = [];

  // take data from http request
  const data = req.body;

 // Get a Postgres client from the connection pool
  pg.connect(connectionString, (err, client, done) => {
    // Handle connection errors
    if(err) {
      done();
      console.log(err);
      return res.status(500).json({success: false, test:abc, data: err});
    }
    // SQL Query > Insert Data
    client.query('INSERT INTO elsadev(devid, timestamp, location, event) values($1, $2, $3, $4)',
    [data.devid, data.timestamp, data.location, data.event]);
    return res.json({"status":"success"});
  });
});
```

Figure 2. Handling POST requests and inserting a new entry in the database.

In a similar way a GET request is also handled. Such a request, targeting the same URL, is being served back the entire content of the events table. This is usually the case when a user with administrator privileges accesses the Web interface of the system and all the events in the database are being displayed in a table and both on a map, as it will be shown in the following sub-section. Basically, in this case the database is being queried and the results are encompassed in a JSON array of strings which is then sent back to the client embodied in the return message.

```
app.get('/api/v1/reports', (req, res) => {
  const results = [];
  // Get a Postgres client from the connection pool
  pg.connect(connectionString, (err, client, done) => {
    // Handle connection errors
    if(err) {
      done();
      console.log(err);
      return res.status(500).json({success: false, data: err});
    }

    const query = client.query('SELECT * FROM elsadev');
    // Stream results back one row at a time
    query.on('row', (row) => {
      results.push(row);
    });
    query.on('end', () => {
      done();
      return res.json(results);
    });
  });
});
```

Figure 3. Handling GET requests.

2.1.2 Web interface

The user interface has been implemented as a Web page residing on the same Web Server, by using several web technologies like JQuery, Bootstrap, CSS and HTML. This interface is used for displaying the recorded events in the database both as a table and also on a map, showing the location of each SCB that reported an event. The Web page has been designed in order to be easily scalable according to the resolution or displaying device (support for Mobile browsers included). While it is accessible on a public URL (http://etc.unitbv.ro/elsa), access to the events is being regulated using user authentication and privileges. We envisage the interface as providing data and management options on a role-based model, e.g. the home user has read-only access to the events recorded by the smart circuit breaker in his home, while an authoritative role - an employee from the electricity providing company - can not only view all the events from all devices, but also act based on them (like turning power back on in case of a decoupling event, or sending a field team for on-site investigation or repair, etc.).

For example, and administrator can view the total number of online SCBs, the distinct locations that they are reporting from and also the number of reported events, as shown in Figure 4.

Figure 4. Administrator web interface: showing general device and events statistics.

A table is displayed on the Web page showing all the recorded events and their properties: the unique ID of the reporting SCB, time and date of the event, the type and the location (latitude and longitude). The last parameter is being hard-coded in each SCB upon setup at its specific location. Such an event table is depicted in Figure 5.

Evenimente inregistrate in baza de date

Id	Id dispozitiv	Data	Eveniment	Locatie
34	508	21.08.2017 21:33:08	8	45.64861
35	509	21.08.2017 21:33:09	9	45.64861
36	40	11.09.2017	3	45.64 25.63
37	41	11.09.2017	2	45.644414 25.605016
38	42	11.09.2017	1	45.668769 25.567284
39	43	11.09.2017	4	45.652720 25.545211
40	44	11.09.2017	3	45.642911 25.638693
41	45	12.09.2017	2	45.644911 25.658693
42	46	14.09.2017	1	45.647811 25.659293
43	47	14.09.2017	1	45.647811 25.679293
44	47	14.09.2017	3	45.757811 25.879293

Figure 5. Web interface showing a table with all the recorded events.

The interface also includes a Google Map (displayed in Figure 6.) showing all the reported events as markers on the map, each with a different color according to the event type. Clicking a marker displays detailed information about that event (time and type).

Implementing this geographical display of events has been accomplished using the Google Maps JavaScript API by instantiating a Google Map on the Web page and populating it with markers for each event present in the database. Different marker colours and icons have been used to represent the events according to their type for a better visual representation. The code accomplishing this feature is shown in Figure 7.

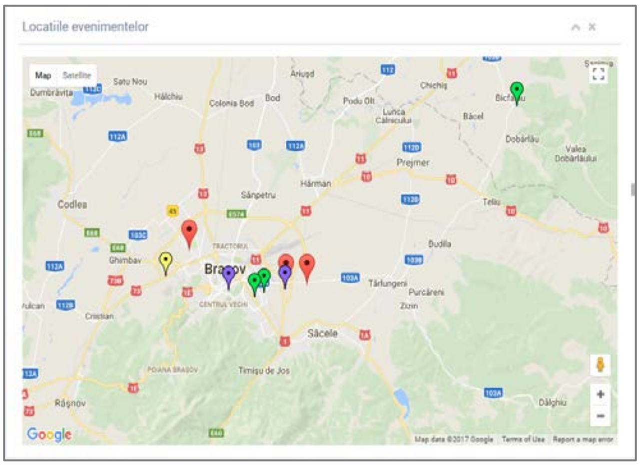

Figure 6. Events being shown on a Google Map.

```
if($scope.databaseData.length != undefined && $scope.databaseData.length > 0){
    for(var i = 0; i < $scope.databaseData.length; i++){
        var locatieGps = $scope.databaseData[i].location.split(" ");
        var latitude = parseFloat(locatieGps[0]);
        var longitude = parseFloat(locatieGps[1]);
        var pozitieGeografica = {lat: latitude, lng: longitude};

        var marker = new google.maps.Marker({
         position: pozitieGeografica,
         map: map,
         title: 'Dispozitiv ' + $scope.databaseData[i].devid
        });

        var eventType =  $scope.databaseData[i].event;
        if(eventType == 2){
          marker.setIcon('http://maps.google.com/mapfiles/ms/icons/purple-dot.png')
        }else if(eventType == 3){
          marker.setIcon('http://maps.google.com/mapfiles/ms/icons/green-dot.png')
        }if(eventType == 4){
          marker.setIcon('http://maps.google.com/mapfiles/ms/icons/yellow-dot.png')
        }
    }
}
```

Figure 7. Web interface source code: adding map markers for each event.

An event represents one of the 5 malfunctions that the Smart Circuit Breaker detects. These event types together with their unique code used to identify them in the JSON transmission and the corresponding marker color shown on the map are enlisted in Table 1.

Table 1. Event types, codes and corresponding marker colors

Event type	Code	Marker colour
Electric arc	0	Red
Short-circuit	1	Yellow
Electrocution	2	Blue
Overvoltage	3	Violet
Overcurrent	4	Green

2.2 Smart Circuit Breaker Embedded Communication

2.2.1 Ethernet

The Ethernet shield SHL-41 is a separate extension module of the Arduino Uno platform which has been connected to it and also to the Internet by an Ethernet cable connection to an online router.

The communication routine for sending data to the Web server has been developed using the Arduino IDE (Integrated Development Environment) using dedicated libraries and implements the following functions:

- Instantiating the required headers, libraries and the microcontroller's watchdog timer
- Setting up the variables for the Ethernet communication based on the shield's MAC value
- Instantiating an EthernetClient type object
- Creating the JSON string format used for encapsulating the data to be sent to the online Web server

The actual transmission of data is accomplished by a call to the *UploadData*() method, which initializes Ethernet using DHCP, creates the JSON string with the event and device data, and sends this data through a HTTP POST request to the Web server API at http://etc.unitbv.ro/elsa/api/v1/reports . The code for these actions is shown in Figure 8.

```
wdt_reset();
ethernetClient.stop();

int retValue = ethernetClient.connect(server,80);
if (retValue)
{
  ethernetClient.println("POST /elsa/api/v1/reports HTTP/1.1");
  ethernetClient.print("Host: ");
  ethernetClient.println(server);
  ethernetClient.println("User-Agent: Arduino/1.8.2"); //1.7.8
  ethernetClient.println("Content-Type: application/json");
  ethernetClient.print("Content-Length: ");
  ethernetClient.println(json.length());
  ethernetClient.println();
  ethernetClient.println(json);
  Serial.println("        JSON sent.");
```

Figure 8. Sending data from an SCB by a HTTP POST request.

The Ethernet-based data transfer between a SCB device and the data concentrator (Web server) has been tested on both ends. On the microcontroller side this has been accomplished by monitoring a data transmission using the Serial Monitor tool of the Arduino IDE, as shown in Figure 9.

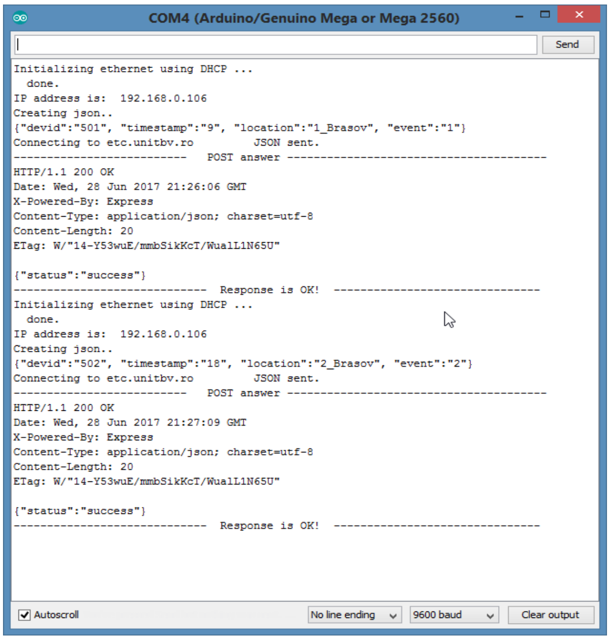

Figure 9. Ethernet communication - Monitoring the data transmission from the SCB and displaying the server's response message.

2.2.2 Wi-Fi

The Wi-Fi connectivity of the Smart Circuit Board has been implemented using the Wemos D1 Wi-fi ESP8266 shield which is compatible with the Arduino IDE leading to easy connectivity and software development.

The communication routine for sending data to the Web server has been developed using the ESP8266Wifi dedicated library and implements the following functions:

- Instantiating the required headers, libraries and the microcontroller's watchdog timer
- Setting up the variables for the Wi-Fi communication (including storing the SSID – Service Set Identifier – authentication)
- Establishing a connection with the server URL (etc.unitbv.ro) using the WifiClient client available through the specific library file and configuring the request parameters and settings
- Creating the JSON string format used for encapsulating the data to be sent to the online Web server

Following the above steps, the JSON string contains the same fields and values as described above for the Ethernet communication. Also, the Wi-Fi interface has been also validated using the same Serial Monitor tool as shown in Figure 10.

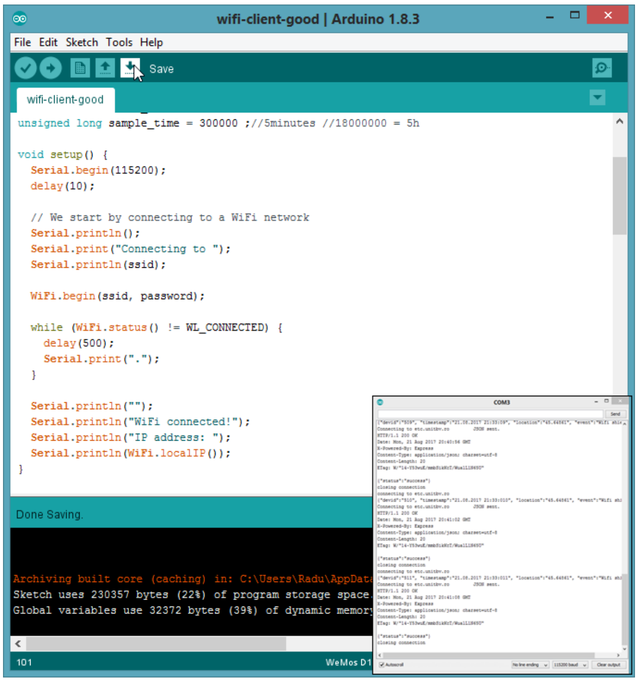

Figure 10. Wi-Fi communication - Monitoring the data transmission from the SCB and displaying the server's response message.

3. Results and discussion

By transcending from ordinary circuit breakers to smart monitoring and communication nodes inter-connected in a network, these devices relate to the concepts and paradigms of the Internet of Things [10]. This integration of smart circuit breakers in the Internet of Things brings on multiple advantages with regard to the security, scalability and inter-operability of the entire system. There is already a renowned concept of the Internet of Energy (IoE) [7] that encompasses Smart Grid and IoT visions and the smart circuit breaker infrastructure can also relate to this concept since it also meets the underlining requirements, among which:

- The possibility of easy, uniform interaction with different ICT (Information and Communication Technologies) due to the IP-based networking feature.
- Low-cost, low-power devices – as promoted by the recent trends in IoT communication [11] – a feature achieved by integrating in the smart circuit breaker low-power microcontrollers and shields.
- Communication technologies based on optimized Internet protocols [12] – the SCB's each have an IP address and communicate with the data concentrator via very basic HTTP requests (thus low power and low complexity) in order to overcome the inherent hardware/software limitations of embedded devices.
- Interoperability. This has always been a key requirement for any distributed computing infrastructure for obtaining a stable network in the case of numerous connected devices. By adopting an IP based communication using HTTP requests and JSON-structured messages the SCB system provides a seamless integration, cooperation and two-way communication with other Smart Grid systems.
- Scalability. Any smart distributed communication infrastructure requires the scalability of connecting a variable number of devices (which can vary dynamically from many to a few and vice-versa) and this is why the SCB system is using an IP-based network which is credited [13] as an effective solution for the communication needs of a smart grid.
- The protocols and communication standards used favor data exchange with other entities (e.g. utility companies, emergency services) that employ a REST-full communication paradigm for their networking interface [7].

Meeting the requirements from the technical and the usability points of view is achieved not just by implementing the appropriate network protocols, communication interfaces and data standards, but also integrating the entire infrastructure with a Web-based visualization interface for monitoring and control purposes, which is a practice in this field in order to make interaction with the distributed devices more easy by using a Web application (like the WebIoT app [14]). Thus, the Web application that was developed for interfacing the SCB system offers a geo-referenced visualization of the smart devices and also capabilities for monitoring the recorded events and taking necessary actions.

4. Conclusion

This paper described the communication interface of a Smart Circuit Breaker distributed system. The implementation is based on known IoT Web standards, protocols and services in order to meet the Smart Grid requirements in term of interoperability, scalability and accessibility.

Each Smart Circuit Breaker embeds an Arduino microcontroller coupled with a communication shield (either Ethernet or Wi-Fi) that makes each device IP-relatable in the Internet. The recorded events at the consumer location (power grid malfunctions like short-circuit, over-voltage, over-current, electric arc or electrocution) are sent via JSON-strings embodied in HTTP requests to a data concentrator (Web Server linked to a PostgreSQL database) for storage, analysis and monitoring.

The system also provides a Web application that enables an easy and efficient monitoring of any events recorded by the devices and thus permits quick actions when necessary.

For future work, we envision implementing the communication between the SCBs and data concentrator using the LoRa interface, adding advanced notifications in the case of events by SMS (Short Message Service) and e-mail, and also developing a mobile app for the end-user to monitor the events recorded by the SCB from his own home.

Acknowledgements

This work was developed in the framework of the national grant 62PED/2017 within the Romanian National Plan for Research, Development and Innovation for 2015-2020 (PNCDI III).

References

[1]. Kobes, M., Groenewegen, K., & Duyvis, M. G. (2009). Consumer fire safety: European statistics and potential fire safety measures. *Vienna Austrian Fed. Minist. Labor, Soc. Aff. Consum. Prot*, *31*.

[2]. Aciu, L. E. (2016). Studies on the Electromagnetic Disturbances Produced by Ac Electric Arc. *Bulletin of the Transilvania University of Brasov. Engineering Sciences. Series I*, *9*(1), 55.

[3]. Liu, J., Huang, G. M., Ma, Z., & Geng, Y. (2011). A novel smart high-voltage circuit breaker for smart grid applications. *IEEE Transactions on Smart Grid*, *2*(2), 254-264.

[4]. Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2013). A survey on smart grid communication infrastructures: Motivations, requirements and challenges. *IEEE communications surveys & tutorials*, *15*(1), 5-20.

[5]. Vaccaro, A., & Villacci, D. (2005). Performance analysis of low earth orbit satellites for power system communication. *Electric power systems research*, *73*(3), 287-294.

[6]. IEC-TC 57, (2003). Communication networks and systems in substations- Part 1: Introduction and overview, *IEC Standard IEC/TR 61850-1*, Edition 1.0.

[7]. Bui, N., Castellani, A.P., Casari, P. and Zorzi, M., (2012). The internet of energy: a web-enabled smart grid system. *IEEE Network*, *26*(4).

[8]. Ogrutan, P. L., Gerigan, C., & Nicoara, I. I. (2017, May). Studies on the implementation of algorithms for electric arc detection. In *Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), 2017 International Conference on* (pp. 1007-1012). IEEE.

[9]. Jacobsen, R. H., Torring, N., Danielsen, B. V., Hansen, M. T., & Pedersen, E. B. (2014, February). Towards an app platform for data concentrators. In *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES* (pp. 1-5). IEEE.

[10]. Zorzi, M., Gluhak, A., Lange, S., & Bassi, A. (2010). From today's intranet of things to a future internet of things: a wireless-and mobility-related view. *IEEE Wireless Communications*, *17*(6).

[11]. Bressan, N., Bazzaco, L., Bui, N., Casari, P., Vangelista, L., Zorzi, M. (2010). The deployment of a smart monitoring system using wireless sensor and actuator networks. In *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, (pp. 49-54). IEEE.

[12]. Castellani, A. P., Gheda, M., Bui, N., Rossi, M., & Zorzi, M. (2011, June). Web Services for the Internet of Things through CoAP and EXI. In *2011 IEEE International Conference on Communications Workshops (ICC)*, (pp. 1-6). IEEE.

[13]. Lobo, F., Cabello, A., Lopez, A., Mora, D., & Mora, R. (2008). Distribution network as communication system. In *SmartGrids for Distribution, IET-CIRED*. CIRED Seminar, pp. 1-4.

[14]. Castellani, A. P., Dissegna, M., Bui, N., & Zorzi, M. (2012). WebIoT: A web application framework for the internet of things. In *Wireless Communications and Networking Conference Workshops (WCNCW)*, IEEE (pp. 202-207).