Evaluation of OPC-UA Framework Applied to Microgrid Energy Dispatch Management

Roberto Alexandre Dias
IFSC
Av. Mauro Ramos, 950
Centro – Florianópolis, Brasil

Gregory das Chagas Gomes
IFSC
Av. Mauro Ramos, 950
Centro – Florianópolis, Brasil

Marcelo Lobo Heldwin
UFSC
Campus Reitor João
David Ferreira Limas _
Trindade, Florianópolis - SC

ABSTRACT

This work presents a review on the OLE FOR PROCESS CONTROL – UNIFIED ARCHITECTURE OPC-UA framework and standards aiming to use OPC-UA in SmartGrids applications. The support for Service Oriented Architecture (SOA) and native security implementation of the OPC-UA are analyzed as an option to supervise and control small Microgrids, like microgeneration farm in Green Datacenters. Based in these features a communication model based on OPC-UA framework was proposed. The main contribution of this work is a performance evaluation of the OPC-UA transmission time between Microgrids devices (electrical converters and inverters, power meters and controllers) and the comparison with IEC 1646 and IEC 61850 standards requirements.

General Terms

Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords

OPC-UA, SmartGrids, Microgrids, Green datacenters, Cybersecurity.

1. INTRODUCTION

According to [16], the increase in the electricity price and efforts for fossil fuels reduction has placed the SmartGrids technologies in evidence worldwide, especially in the field of distributed electric generation.

The implementation of small-distributed generation arrays with energy resources that provide supply autonomy, called microgrid, has been widely promoted by the governments of several countries, especially in Europe, to mitigate the greenhouse effect. In Brazil, the regulatory frameworks of distributed generation are beginning to be established [24], [25].

The basis for such systems lies on the integration of energy and information networks to form automated, secure, reliable and efficient energy supply systems. The microgrids control, supervision and automation functions provide the means to achieve those objectives.

Thus, the coordination of large-scale microgrids demands efforts in providing a secure communication infrastructure for managing them opening a promising field of research in this area.

The choice of an efficient and low complexity management platform for supervision and control of small microgrids is a great challenge. In this sense, this study aims to present the standard OPC-UA as an alternative to microgrid management, leading to a simple, cost competitive, efficient and safe system.

Some papers are related to this scenario. In [21] the authors present the OPC-UA functionalities, like information modeling, communication services and information security applied to smartgrids.

In [22] an optimized OPC-UA middleware support to IEC 61850 is proposed for energy automation applications. In this paper, the authors use OPC-UA to implement a vertical communication in an IEC 61850 ecosystem. This approach is proposed too in [23]. In this paper a multi-agent system prototype is implemented for a multi-agent control systems based design, using IEC 61850 and OPC UA integration.

In [24] an adaptive microgrid operation based on IEC 61850 is presented. The main contribution of this paper are the data modeling of IEDs (Intelligent Electronic Devices) that compose the microgrid for plug and play IED integration.

The work is organized as follows: section II presents a review of the OPC-UA standard with focus on the characteristics that are important to microgrid applications. Section III presents a literature review on SmartGrids and microgrids. Section IV presents an architectural proposal for a data communication system for managing microgrids based on OPC-UA. Section V presents a performance analysis of OPC-UA and its feasibility of use for managing SmartGrids, where the achieved results are compared with state-of-the-art approaches and standards for electric equipment. Finally, section VI presents conclusions and a proposal for future work.

2. OPC UA STANDARD

The Object Linking and Embedded for Process Control - Unified Architecture (OPC-UA) is the new standard sponsored by the OPC Foundation [1] coming to succeed the conventional OPC technology based on COM/DCOM from Microsoft. According to Cerami, Ethan [2], it was created to promote interoperability across heterogeneous systems. Unlike the previous standard based on DCOM, it is cross platform and natively adhering to the service-oriented architecture (SOA) paradigm, based on Web services technologies. Thus, OPC-UA provides transparency, application decoupling, and may be encapsulated in XML/SOAP messages over HTTP(S) or compressed in binary mode called UA-Binary, developed specifically for this new standard by providing a considerable increase performance and safety [3].

The main motivations for the design OPC-UA are [4]:

Robustness and fault tolerance;

- Redundancy;
- Platform-independency (runs on Windows, Linux and small embedded systems);
- High scalability;
- Reduced fingerprint;
- High performance, particularly when compared with Web technologies;
- Ease integration with the Internet through firewalls;
- Native Implementation of end-to-end security by creation of secure channels and access control;
- Interoperability with heterogeneous systems and backward compatibility capacity;
- Common data model to standard conventional OPC;
- Object Orientation;
- Use of metadata for describing services and support for data types and complex methods.

OPC-UA had it first specification in 2008 by the OPC Foundation and was internationally standardized by the IEC in 2011 as IEC 62541 standard, recently updated and extended in 2015 [5].

2.1 OPC-UA Architecture

The OPC-UA standard, as a middleware for RPC operates in the client/server paradigm. The IEC 62541 does not provide one client and server API, but the OPC Foundation offers to its members, client SDKs and open basic server in ANSI C language, based on the standard. Many solution providers sell SDKs that provide friendly APIs (stack wrappers) on several platforms (like C, C++, C# and Java) for development of client and server applications. Figure 1 shows the architecture of standard OPC-UA.

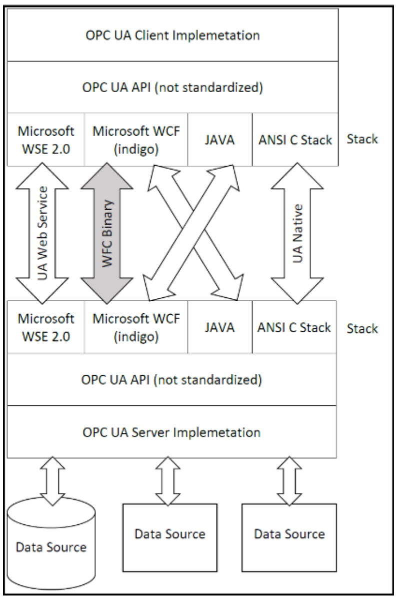

Figure 1 - OPC-UA Architecture

The client application uses the API to send service requests. The messages exchanged between the client and server can use XML/SOAP encapsulation in the HTTP(S) for Web Service, favoring interoperability with enterprise systems such as ERP. However, applications using Web Services or even the DPWS specification (Device Profile for Web Service) that allow the implementation of service-oriented architecture in embedded systems with reduced processing power, have performance problems due to the complexity of SOAP messages, as demonstrated in [18], particularly where the establishment of secure channels is required. Thus, the messages exchanged between client and server can be encapsulated over a new binary protocol called TCP-UA or UA Native Binary.

This new approach enables enables message mapping directly in a compressed package over TCP. Current applications call this opc.tcp protocol. The URL syntax for access an OPC-UA server could be opc.tc://server_name/resource_url:port. The performance of UA Native, even over the Internet is comparable to traditional OPC in the LAN.

In both mappings (XML/SOAP and UA Native) it is possible to connect a device on the shop floor or in a Microgrid bay, directly through an OPC-UA client, with high performance and security, cross firewalls, as shown in figure 2.

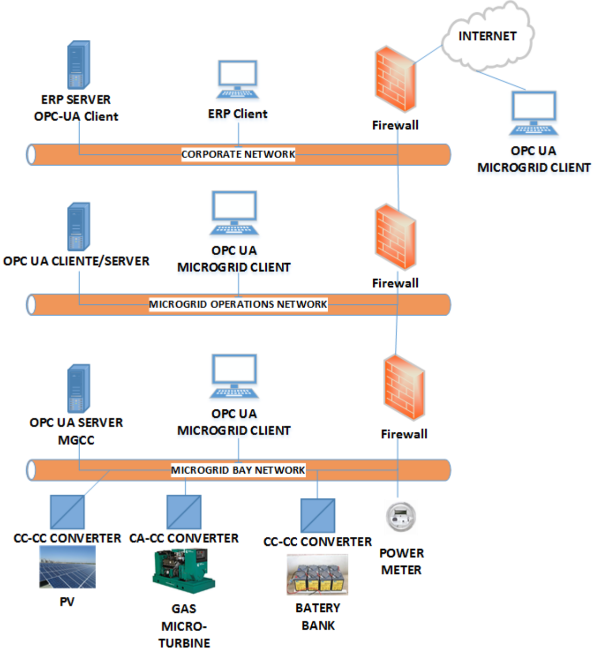

Figure 2 - OPC-UA network

OPC-UA extended the functionality of traditional OPC. The main new functionalities are self-discovery features, event subscription for monitoring items and security services.

3. DISPACH OPTIMIZATION IN SMALL MICROGRIDS AND GREEN DATACENTERS

The main dominant factors defining electrical systems consist of: government policies, such as regulatory frameworks and government programs; efficiency needs of the utilities and demand side management; use of new hardware and software technologies, as well as environmental demands for energy efficiency, pollution reduction strategies, such as the use of alternative sources to fossil fuels.

The evolution of information and communication technologies (ICTs) as distributed systems and sensors , two-way secure communication channels, advanced software for large-scale data management (big data), intelligent autonomous control systems paves the way for modern electric power systems in the XXI century [14].

In this context, the concept of SmartGrids [14] arises: "The SmartGrid can be defined as an electrical system, which uses the information with secure two-way communication technologies, and the use of computational intelligence in an integrated way across generation, transmission, substations, distribution and consumption in order to get a clean, secure, reliable, resilient, efficient and sustainable.“ Also according to [14], the full SmartGrids adherence still needs changes so that interoperability requirements and implementation of plug-and-play functionality can be added to then. Thus, the SmartGrids will meet the following characteristics:

1. Auto recovery: automatic, proactive fault detection with system reconfiguration and re-routing of power flow;
2. Flexibility: fast and safe reconnection generation systems distributed in any system point and at any time;
3. Predictability: the use of machine learning, weather forecast, stochastic analysis for reconfiguration of the preventive system;
4. Interactivity: ability to interact with the comprehensive information system from the utilities to the final consumer so that all players in the system can contribute to its configuration and optimization;
5. Optimization: power flow optimization possibility in the autonomous system in order to minimize costs and management on the demand side;
6. Security: adoption of secure communication mechanisms for establishing secure channels end-to-end, mitigating cyber-attacks on the electrical system.

3.1 Microgrids and Green Datacenters

A microgrid [15] is a local energy network involving energy sources and energy storage systems. These systems must include local control features. They should also be able to detect main network failures and some other types of events to disconnect from the grid. In these situations, they generate energy autonomously for local loads in an operating mode called "islanded" in addition to the connected to grid operation mode. A microgrid requires a strong control and supervision system in order to achieve such features. In addition, microgrids are key elements in future Smart Grids and, as such, will be typically required to exchange data and features with distribution operation centers.

With the growth of cloud, data processing, large information technology facilities operating in network environment of massively distributed computers are proven one of the major consumers of electricity [16], operating 24x7, requiring uninterrupted power supply. This context motivates the use of uninterruptible power systems (UPS) technologies in datacenters. Energy consumption growth in these facilities has motivated the implementation of the so-called “Green Datacenter" (GDC) concept, which leads to data centers that adopt techniques of efficient and environmentally friendly energy consumption [16]. GDCs might employ microgeneration arrangements using alternative non-fossil energy sources such as photovoltaic, wind and others sources of energy. Green data centers are characterized by incorporating mainly non-linear electrical loads such as computers and communication network equipment, which can be advantageously powered from DC power networks [26], [27].

Moreover, energy intensive components in these applications are air conditioning systems, typically powered by AC distribution networks. Green datacenters energy supply systems can strongly benefit from the microgrid concept.

In order to meet the requirements of energy and economic efficiency of microgrids comes the need for centralized management architectures, employing a microgrid control node. This is typically named Microgrid Central Controller - MGCC [17]. Figure 3 shows the architecture of a microgrid employing a MGCC.

In this scenario, the MGCC needs to be data network connected with other Microgrid devices and, in the case of distributed generation, with the energy supply systems. In both cases, the performance and safety requirements to SmartGrids communication systems must be observed. The MGCC performs the following functions [17]:

1. Establishes and manages the best technical and/or economic policy for dispatch control, and management of the energy resources and also demand side management (demand response techniques);
2. Manages the operation modes according to given requirements, e.g., island operation, connected operation, maintenance/test operation, among others.

Based on this context, the scope of this work is focused on a small island operation-capable DC green datacenter microgrid, with a photovoltaic generation system, employing a MGCC. The communication infrastructure uses the OPC UA as middleware communication, between the MGCC and all other microgrid devices.

4. OPC-UA COMMUNICATION INFRASTRUCTURE

As discussed in section IV, SmartGrids must present interoperability and security requirements in order to allow its remote monitoring and management, often using the Internet infrastructure in micro distributed generation arrangements.

In this perspective, as shown in section III, OPC-UA enables compliance with the microgrids requirements, such as robust and fault tolerant operation, as well as two-way, end-to-end secure communication infrastructure. OPC-UA provides a natively service-oriented architecture paradigm. This feature favors the integration of microgrids to the corporate management system, such as ERPs (Enterprise Resources Planning) applications, with support to supervision and control tools such as SCADA (Supervisory, Control and Data Acquisition).

The data communication model shown in figure 4 was adopted in this work.

As shown in figure 4, the model presupposes the use of intelligent inverters and converters, supporting the OPC-UA or by the use of gateways that convert the legacy protocols to OPC-UA. The second option was adopted in this proposal.

The use of OPC-UA security features is assumed to create secure channels between MGCC (OPC-UA client) and the microgrid devices (OPC-UA servers) employing asymmetric cryptography of Public Key Infrastructure (PKI).

In this model, the MGCC is a computer that will run an OPC-UA client application responsible for reading at regular times each device data (server) to perform an energy flow dispatch optimization problem instance. The variables of interest are:

1. Pmax : maximum power that can be generated at the time of reading by the generating unit.

2. PTO: state variable that indicates the operating condition of the generating unit. If PTO = 1 the unit generates normally. If PTO = 0, the unit failed (in this case Pmax should be considered zero for the optimizer). If PTO = 2 it means that unit is in the loading ramp in the case of batteries or starting ramp (in case of diesel generator or gas micro turbines). In the latter two cases, Pmax should be considered zero by the optimizer.

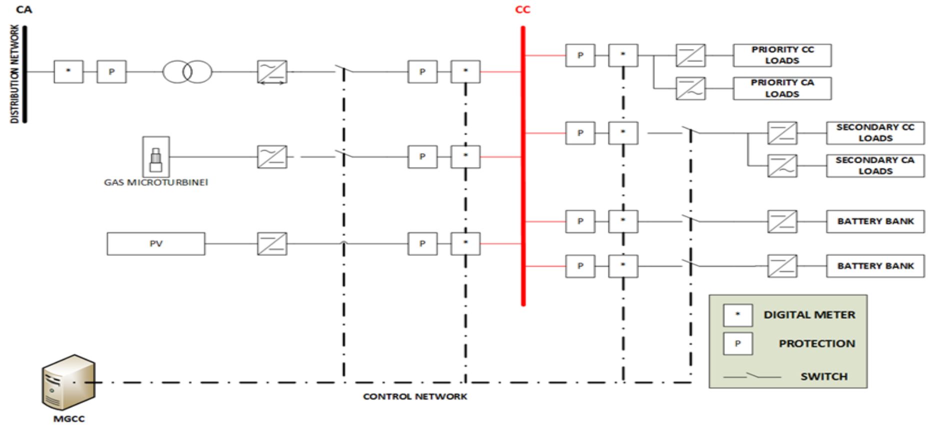

Figure 3 - Microgrid Architecture

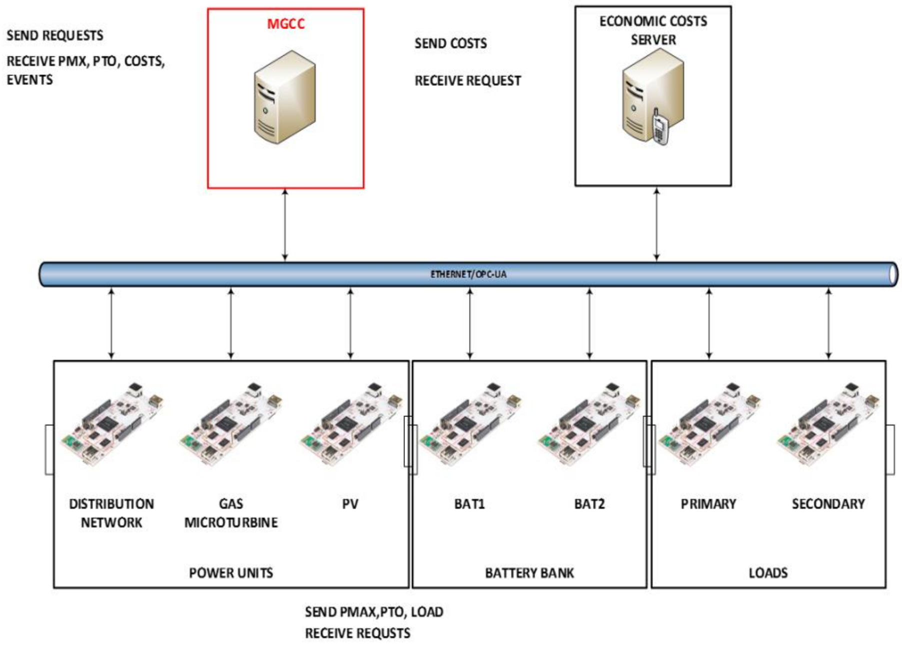

Figure 4 - Microgrid Communication Model

3. C: generation costs of the units. The generation costs will be provided by a specific server that will maintain the seasonal billing tables from the utility, fuel prices, maintenance cost tables of solar panels and battery banks.

4. PL: Power loads. A power meter (server) could provide the current power load demand. When the battery banks are in loading ramp its power is negative, that is, its absolute value must be added to the power consumed by the load group for optimizer purposes.

5. Pref: reference power to configure the inverters and converters after each round of the optimization algorithm. The MGCC subscribes to the OPC-UA event service of each server associated with the generating units, so that the MGCC should be notified at every change in the

6. Value of PTO. In such events, a new reading of Pmax and PTO must be done automatically, regardless of the polling time, and a new optimization instance must be immediately executed.

The MGCC is responsible for the following activities:

1. Reading of data from the server;

2. Registration and monitoring items;

3. Running the optimizer;

4. Control the number of battery banks of load cycles: the MGCC multiplies the cost of the unit over a time interval “T” (to be configured according to the manufacturer's data sheet) at each X load cycle (to be configured according to the manufacturer datasheet). With this, the optimizer will decrease the probability of use and, thus, increase the battery life. After the time “T” the MGCC should reset the counter and return the original cost value (without multiplier). Every time when an event from servers (changing the value of the PTO) is received, the MGCC instantly reads Pmax and the costs of each generator units to run a new instance of the optimizer.
5. Set the inverters and converters with the power values resulting from the optimizer by writing the Pref register (reference power);
6. Generate tabular and charts reports.

The OPC-UA servers will be connected to the power inverters connected to the utility and the diesel generator (or gas micro turbine), inverters connected to the battery bank and the PV generators. They are responsible to send PMax and PTO. The servers can be deployed on embedded devices connected directly to the power inverters and converters. In addition, the servers must provide an OPC-UA event service, every time that the value of the PTO value is changed. The OPC-UA server connected to the power meters sends the power consumed by the loads.

The server applications were implemented in C language using the Embedded Server SDK of MatrikonOPC [9], through the IDE Eclipse Luna C / C ++ toolchain using a cross-compiler for ARMv7 [10] [11]. They run on an embedded system with an Alvinner A20 processor [12] called PCDuino [13]. The main characteristics of PCDuino are:

1. CPU: AllWinner A20 SoC 1GHz ARM Cortex A7 Dual Core;
2. GPU: OpenGL ES2.0, OpenVG 1.1 Mali 400 Dual Core;
3. 1GB DRAM;
4. 4GB intern storage Flash;
5. HDMI Video Output with HDCP Support;
6. Ubuntu 12.04 and Android ICS 4.2 Supported;
7. 0.1" Spaced GPIO Headers;
8. RJ45 Ethernet Connection and On-Board Wi-Fi Module;
9. Two USB 2.0;
10. UART, ADC, PWM, GPIO, I2C, SPI.

The MGCC application was developed in C++ using the SDK Client also by the MatrikonOPC. The MGCC runs on the Linux Ubuntu 14.04 LTS environment. The IDE also uses Eclipse Luna C/C ++.

5. OPC-UA PERFORMANCE EVALUATION

This section shows the performance of the communication between the client and server by measuring the time difference between the OPC-UA read request and read response messages.

The read request was sent by the client application running on a PC with Intel i7 quad core processor, a virtual machine Linux Ubuntu 14.04 LTS, 32-bit, and 3 gigabytes of RAM. The server application sends the read response message, performed on PCDuino. The connection between the client and the server was made through an Ethernet switch at 100 Mbps.

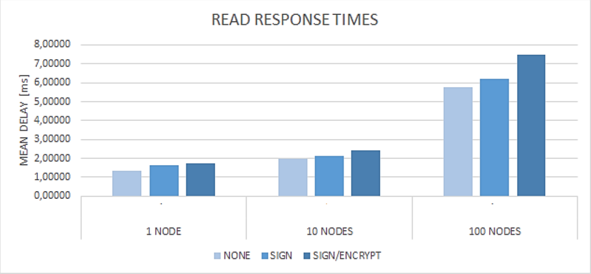

Figure 3 - Response times between read requests and read responses

Three measurement scenarios were experimented: without security support (none), with only authentication security support (sign) and securely support to authentication and encryption (sign & encrypt). Security implementation is based on asymmetric RSA encryption algorithm with 128-bit key. Figure 5 shows the average time between the read request and read response from a 32-bit integer variable in the three described scenarios. The averages were calculated using one variable per request, 10 variables per request and 100 variables per request.

For a given number of variables, independently of the undertaken setting, the time between the read request and the read response varies little, showing a great API efficiency to implement security policy.

Table 1 shows the average time for reading variables and their standard deviations for the 100 samples in each series.

It is seen that the read times are relatively low and there was low variation between the measured values even with encryption enabled (sig & encrypt), indicating that the use of OPC-UA to exchange control messages on microgrids is feasible.

Table 1 - Average read times and their standard deviation

	1 Variable		**10 Variables**		**100 Variables**	
	Mean [ms]	Std Dev	Mean [ms]	Std Dev	Mean [ms]	Std Dev
NONE	1,323	0,011	1,960	0,019	5,754	0,031
SIGN	1,627	0,001	2,114	0,001	6,208	0,008
SIGN & ENCR	1,731	0,012	2,413	0,012	7,461	0,024

5.1 Related Works

In [18] the authors proposed an approach to supervision and control of industrial network using the of Service Oriented Architecture paradigm with DPWS. A DPWS server was developed with Microsoft .NET Micro framework platform in an embedded system based on ARM9 microcontroller, which was acting as a gateway between a DPWS Client and one PLC with MODBUS RTU protocol communication. The SOAP performance in the DPWS is very low, compared to

OPC-UA. Furthermore, this work has not implemented a secure connection.

Table 2 shows the delay of the message exchanges between the client application and the server DPWS.

Table 2 - DPWS exchange message times

Operation	Connection Time [s]	Request/ Response [s]	Total Time [s]
Probe	No connection	1.180	1.180
restart (oneWay)	0.355	0.547	0.902
readHolding-Registers (twoWay)	0.357	0.872	1.220
presetSingle-Register (twoWay)	0.349	0.798	1.147
subscribe	0.361	1.109	1.471
Register-Changed (Eventing)	0.023	0.052	0.211

Table 2 shows that the time for reading a 2-byte variable (Modbus register) is 1.22 seconds (nearly one thousand times greater than the time spent by OPC-UA). However, like OPC-UA, the DPWS supports event service, where the response time is around 200 ms (just under 100 times slower than that of a variable reading process the OPC-UA). This indicates that the use of UA-binary protocol has a great level of compression, optimizing the OPC-UA communication performance.

A performance analysis of OPC-UA security model compared to the standard IPCSEC is presented in [19]. The message transmission delay measurements are made between a client/server application with different packet sizes (this includes data and overhead). The test runs with IPSEC over UDP protocol, where transmission performance is superior. Figure 6 shows the performance of OPC-UA without safety profile (none) and RSA 128-bit encryption.

The variation of the delay with and without 128-bit RSA authentication is not very significant in OPC-UA as seen in Figure 6, similarly to the measurements made in the experiments reported in this work. The measurement time is of the same order of magnitude; however, the authors do not inform the environment of the testbed.

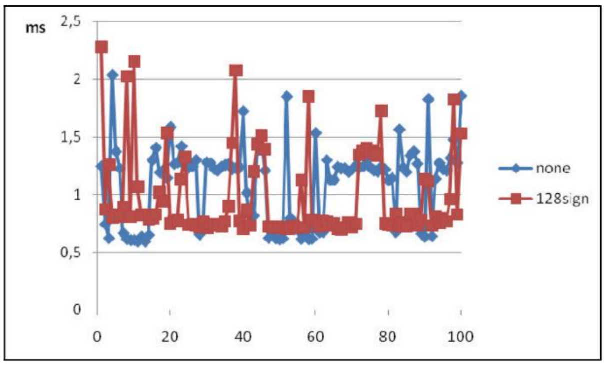

Figure 4 - OPC-UA performance with and without security profile

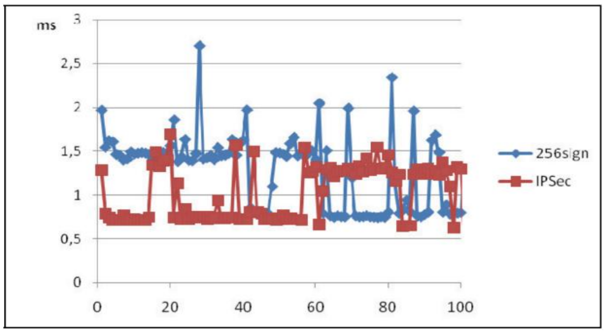

Figure 5 - OPC-UA performance with and without security profile

Figure 7 shows a comparison between the performance of OPC-UA with 256-bit RSA authentication, and IPSEC UDP, as presented in [19].

As can be seen, IPSEC/UDP has better performance. However, IPSEC presents some challenges, such as configuration complexity, complicating its use in embedded systems, as well as not providing end-to-end security, being susceptible to "man in the middle" attack as clarified in [20].

Table 3 summarizes the comparison between the three approaches discussed in this subsection: OPC-UA, DPWS and IPSEC.

Table 3 - Security implementation comparison

	Read Delay	End-to-End Security	Interoperability	Complexity
OPC-UA	Very good	Yes	Very good	Low
DPWS	**Bad**	Yes	Very good	Medium
IPSEC	Excellent over UDP	**Not**	Medium need a complex client)	**High**

As show in table 3, the transmission latency of DPWS has limitations due to the complexity of the SOAP protocol used. The IPSEC besides not providing end-to-end security presents high complexity of implementation, especially in embedded systems.

5.2 OPC-UA Use in Microgrids

In order to assess the feasibility of use of OPC-UA in the microgrids context the response time requirements of the main standards applied to automation of the electricity sector are listed in the following.

Tables 4 and 5 shows, respectively, the transmission requirements for supervisory and control message of the two main standards applied to SmartGrids, the IEEE 1646 which specifies the requirements for computer networks in substations and IEC 61850, related to substation automation.

Table 4 - Time constraints in the IEEE 1646 standard

Information type	Internals to substation	Externs to substation
Protection	4 ms (1/4 wave)	8-12 ms
Monitoring and control	16 ms	1 s
Operation and maintenance	1 s	10 s

Table 5 - Time constraints in the IEC 61850 standards

Message Type	Definitions	Constraints
Type 1	Messages that require immediate action in IEDs	1A: 3 ms or 10 ms. 1B: 20 ms or 100 ms
Type 2	Messages requiring medium transmission speed	100 ms
Type 3	Message requiring slow transmission times	500 ms
Type 4	Continuous data flow between IEDs	3 ms or 10 ms

The results obtained with the implemented OPC-UA with encryption are compatible with the strictest requirements of both standards, even when the amount of transmitted variables is high, which indicates the feasibility and appropriateness of its implementation in microgrids.

6. CONCLUSIONS AND OUTLOOK

This study is focused in an optimization of energy flow dispatch in small microgrids, such as green data centers. It proposes the use of OPC-UA as the middleware for data communication between devices that compose it including inverters, converters and power meters, coordinated by a centralized manager – MGCC. The emphasis of the work was to assess the feasibility of using the OPC-UA for secure and fast management of the microgrid.

The time delay of exchanged messages between OPC-UA client and server applications with security implementation was efficient, consistent with the time requirements of the IEEE 1646 and IEC 61850, for monitoring and control modern electric equipment, which indicates the feasibility of its use for managing microgrids, being superior to similar approaches such as the DPWS platform.

In addition to the efficiency of OPC-UA, being an industrial de facto standard, the adherence to the service orientation paradigm favors its use in non-specific electrical environments, such as the operation of a data center.

The mathematical modeling for the proposed microgrid optimizer and the complete implementation of a testbed to evaluate the performance the energy flow dispatch optimization in a small microgrid will be presented in future work and profit from the proposed communication system.

7. REFERENCES

[1] OPC Foundation Web Site. Unified Architecture. Last visited in June 2015. https://opcfoundation.org/about/opc-technologies/opc-ua/

[2] Cerami, Ethan. Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL. O'Reilly. Feb. 2012.

[3] Mahnke, Wolfgang et all. OPC Unified Architecture. Springer-Verlag Berlin Heidelberg. 2009.

[4] Leitner, Stefan-Helmut, and Wolfgang Mahnke. OPC UA–service-oriented architecture for industrial applications. ABB Corporate Research Center (2006).

[5] OPC Foundation. https://opcfoundation.org/news/opc-foundation-news/update-iec-62541-opc-ua-published/. Lats visited in June 2015.

[6] Post, Olli, Jari Seppälä, and Hannu Koivisto. The Performance of OPC-UA Security Model at Field Device Level. International Conference in Informatics in Control, Automation and Robotics- ICINCO-RA. 2009.

[7] OpenSSL. OpenSSL Project. https://www.openssl.org/. Last visited in June 2015.

[8] Fernbach, Andreas, and Wolfgang Kastner. Certificate management in OPC UA applications: An evaluation of different trust models. Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on. IEEE, 2012. DOI: 10.1109/ETFA.2012.6489675

[9] MatrikonOPC OPC UA Embedded Server Software Development Kit. http://www.matrikonopc.com/opc-ua/embedded/sdk.aspx. Last visited in June 2015.

[10] Fu, Sheng-Yu, Jan-Jan Wu, and Wei-Chung Hsu. Improving SIMD code generation in QEMU. Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition. EDA Consortium, 2015.

[11] Ubuntu Packages. Download link. http://packages.ubuntu.com/precise/devel/gcc-arm-linux-gnueabihf. Last Visited in June 2015.

[12] Allwinner Technology. A20 Information page. http://www.allwinnertech.com/en/clq/processora/A20.html. Last visited in June 2015.

[13] Linksprite Site. http://www.linksprite.com/?page_id=782. Last visited in June 2015.

[14] Ghafurian, Reza. Smart grid: The electric energy system of the future. Proceedings of the IEEE, 2011. DOI: 10.1109/JPROC.2011.2124210

[15] A.T. Ghareeb, A.A. Mohamed and O.A. Mohammed. DC microgrids and distribution systems: An overview. Power and Energy Society General Meeting (PES), 2013 IEEE, pp. 1-5 DOI: 10.1109/PESMG.2013.6672624

[16] Erol-Kantarci, Melike, and Hussein T. Mouftah. Energy-Efficient Information and Communication Infrastructures in the Smart Grid: A Survey on Interactions and Open Issues. Communications Surveys & Tutorials, IEEE17.1 (2015): 179-197. DOI: 10.1109/COMST.2014.2341600

[17] Liaria, Alvaro, et al. Survey on microgrids: unplanned islanding and related inverter control techniques. Renewable Energy 36.8 (2011): 2052-2061.

[18] Dias, Roberto A., Souza Tiago E. Uma Abordagem de Gerenciamento Integrado de Processos Industriais Empregando a Arquitetura Orientada a Serviços. In: Proceedings of I2ts 2010 - 9th International Information and Telecommunication Technologies Symposium. Rio de Janeiro Brazil.

[19] Post, Olli, Jari Seppälä, and Hannu Koivisto. The Performance of OPC-UA Security Model at Field Device Level. ICINCO-RA. 2009.

[20] Wikipedia. Chamada de procedimento remoto. https://pt.wikipedia.org/wiki/Chamada_de_procedimento_remoto. Acessado em 05 de Agosto de 2015.

[21] Lehnhoff, Sebastian, et al. OPC unified architecture: a service-oriented architecture for smart grids. Proceedings of the First International Workshop on Software Engineering Challenges for the Smart Grid. IEEE Press, 2012. DOI: 10.1109/SE4SG.2012.6225723

[22] Sučić, Stjepan. "Optimizing OPC UA middleware performance for energy automation applications." Energy (2014). DOI: 10.1109/ENERGYCON.2014.6850632

[23] Deng, Wei, et al. "Adaptive Micro-Grid Operation Based on IEC 61850", Energies 8.5 (2015): 4455-4475. DOI: 10.3390/en8054455

[24] Quinteiro Pica, Cesare, Marcos Aurelio Izumida Martins, Tania Nalborczyk Leites, and Nilo Rodrigues. "The regulatory challenge of integrating microgrids in the Brazilian context", in Innovative Smart Grid Technologies Latin America (ISGT LATAM), 2015 IEEE PES, pp. 148-153. IEEE, 2015. DOI: 10.1109/ISGT-LA.2015.7381144

[25] Katia Gregio Di Santo, Eduardo Kanashiro, Silvio Giuseppe Di Santo, Marco Antonio Saidel, "A review on smart grids and experiences in Brazil", Renewable and Sustainable Energy Reviews, Volume 52, December 2015, Pages 1072-1082, ISSN 1364-0321, http://dx.doi.org/10.1016/j.rser.2015.07.182. DOI: 10.1016/j.rser.2015.07.182

[26] Sun, Jianfeng; Yue, Xing; Zhang, Zhaolai; Wang, Wei, "Analysis of High Voltage DC Power System Topologies for Use in Datacenters", in Proceedings of 2013 35th International Telecommunications Energy Conference 'Smart Power and Efficiency' (INTELEC), pp.1-5, 2013.

[27] Salato, M.; Zolj, A.; Becker, D.J.; Sonnenberg, B.J., "Power system architectures for 380V DC distribution in telecom datacenters", in IEEE 34th International Telecommunications Energy Conference (INTELEC), pp.1-7, 2012. DOI: 10.1109/INTLEC.2012.6374469

8. AUTHOR PROFILE

R. A. Dias: bachelor's at Electric Engineering from Federal University of Santa Catarina (1988), master's at Mechanical Engineering from Federal University of Santa Catarina (1996) and doctorate at Electric Engineering from Federal University of Santa Catarina (2004). Has experience in Computer Science, focusing on Telecomputing, acting on the following subjects: web services and distributed systems.

G. C. Gomes: technologist in Industrial Mechatronics (2012), Master in Mechanical Engineering from the Federal University of Santa Catarina - UFSC in the concentration area of metrology and instrumentation. Professor at the Technical Course in Industrial Automation at SENAI-SC, Florianópolis unit and as an external researcher at the Group Embedded and Distributed Systems at Federal Institute of Santa Catarina. Has experience in the fields of mechatronics, embedded and distributed systems and automation - with an emphasis on instrumentation and automation tests widely using Labview platform.

M. L. Heldwein: holds a degree (1997) and MA (1999) in Electrical Engineering from UFSC and doctorate (2007) by the ETH Zurich. He is currently assistant professor in the Department of Electrical Engineering of UFSC. From 1999 to 2001 he worked as a research assistant in the Power Electronics Institute (INEP). From 2001 to 2003 he was an engineer of R & D Informat, working on sources of project with Emerson Network Power. From 2003 to 2008 he worked at the ETH Zurich in Power Electronics area. He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE) and a member of the Association of Power Electronics (SOBRAEP).