Validation of a CoAP to IEC 61850 Mapping and Benchmarking vs HTTP-REST and WS-SOAP

Markel Iglesias-Urkia*, Diego Casado-Mansilla†, Simon Mayer‡ and Aitor Urbieta*
* *IK4-Ikerlan Technology Research Centre, Information and Communication Technologies Area.*
Pº J. M. Arizmendiarrieta, 2. 20500 Arrasate-Mondragón, Spain.
Email: {miglesias,aurbieta}@ikerlan.es.
orcid.org/0000-0001-7708-3252, orcid.org/0000-0001-5836-4198.
† *Deusto Institute of Technology, DeustoTech-Societal Challenges.*
Av. Universidades, 24. 48007 Bilbao, Spain.
Email: dcasado@deusto.es.
orcid.org/0000-0002-1070-7494.
‡ *Pro2Future GmbH and Graz University of Technology.*
Inffeldgasse 25F. 8010 Graz, Austria.
Email: simon.mayer@pro2future.at.
orcid.org/0000-0001-6367-3454.

***Abstract*—With the advent of Smart Grid systems, the digitalization of electrical grid infrastructures aims to improve energy saving and efficiency. The International Electrotechnical Commission (IEC) is one of the organizations that create and manage norms and standards in areas related to electricity and electronics, such as IEC 61850. In recent years, the research community have proposed mappings of different communication protocols to IEC 61850. However, most of the proposals focused on heavyweight interaction paradigms and protocols such as Common Object Request Broker Architecture (CORBA), Data Distribution Service (DDS), HTTP-REST or Web Services. With the proliferation of the Internet of Things (IoT), new lightweight protocols are appearing opening new perspectives to the standard. Hence, this paper firstly presents a validation of a mapping of the IEC 61850 standard to the Constrained Application Protocol (CoAP) and then compares its performance implementing the IEC's mapping against HTTP-REST and Web Services/SOAP. For comparison purposes, the communication latency, the number of total bytes sent, and the size of the overhead are presented for each of the three approaches. To conclude, future perspectives on the suitability of lightweight protocols to the IEC 61850 are provided.**

***Index Terms*—Benchmarking, CoAP, HTTP-REST, IEC 61850, Internet of Things, IoT, Smart Grids, SOAP, Web Services.**

I. Introduction

The International Electrotechnical Commission (IEC) defined the IEC 61850 [1] standard to model, control and monitor electrical substation automation systems, defining a data model, logging and reporting schemes, events, setting configurations, sample data transfers, control commands and data storage functions. Nowadays, the standard extends beyond its initial use cases and has been applied to new domains related to power generation and transmission, such as Distributed Energy Resources, Wind Power Plants, and Hydroelectric Power Plants.

IEC 61850 includes a communication model along some mappings such as the Manufacturing Message Specification (MMS) [2], serial communication [3] and ISO/IEC 8802-3 Ethernet [4]. Other communication protocols have also been proposed to be used with IEC 61850, for instance CORBA or HTTP-REST. With the increasing adoption of IoT, more lightweight and less resource-demanding network protocols (e.g., in terms of RAM, CPU, and network bandwidth) have been designed [5]. Besides allowing to build cheaper devices as they require less computing capability, the resource-constrained IoT devices suit with the aforementioned lightweight protocols to exchange information in real time with minimal impact on performance [6].

The Constrained Application Protocol (CoAP) [7] and MQ Telemetry Transport (MQTT) [8] are two of the lightweight protocols that are currently gaining momentum in IoT environments [9]. These are being adopted in many different environments and use cases, such as IKEA's Trådfri Lights for CoAP [10] and Facebook's Messenger for MQTT [11], or in our institution's projects with clients that use MQTT for industrial machinery and equipment goods monitoring applications. Although MQTT was released before CoAP and can be considered as a more mature approach, CoAP is also gaining popularity for certain scenarios. For instance, in *control* use cases CoAP fits better thanks to its client-server communication paradigm, whilst for *monitoring* scenarios, MQTT's publish-subscribe model is a more straightforward approach to be applied. Nonetheless, with its `OBSERVE` capability, CoAP includes a publish-subscribe-like communication mode without requiring an intermediary broker like MQTT saving a hop for every message transmission. Taking into account this argument, the authors consider that CoAP is a promising option that demands attention for electrical substation environments.

In the related literature, some lightweight protocols have mapped the IEC 61850. However, to the best of our knowledge, there is not a robust mapping of the standard to CoAP

978-1-5386-7108-5/18/$31.00 ©2018 IEEE

Authorized licensed use limited to: University of St. Gallen. Downloaded on April 29,2023 at 16:00:43 UTC from IEEE Xplore. Restrictions apply.

that covers CoAP's full functionality yet aside from [12]. Hence, this paper presents new results from the implementation and validation of the mapping proposed in [12], providing several metrics to compare the implementation against HTTP-REST and Web Services/SOAP mappings through an internal tool that generates code automatically building from a specific model.

The rest of this document is organized as follows. The next section summarizes the relevant related work, while Section III explains the technical background, including IEC 61850, and contrasts different relevant communication protocols (CoAP, HTTP-REST and SOAP). Section IV describes the implementation of the mapping from [12], followed by a validation. Next, we present the results from a comparison of the different communication protocols (in terms of their latency, total bytes sent and message overhead) along with a discussion. Finally, Section VII summarizes the paper and briefly mentions several potential avenues for future work in this domain.

II. Related Work

IEC 61850 includes sections where different application-layer protocols are proposed to be used for communications, e.g., MMS [2] or ISO/IEC 802-3 [4]. There are other standards based on IEC 61850, e.g., IEC 61400, which include a mapping to SOAP Web Services [13], and different protocols have been proposed to be mapped to IEC 61850 (see Table I).

During the early development of IEC 61850, Sanz et al. [14] proposed to standardize a communication protocol for different manufacturers: at that time, each company had their own proprietary protocol. The standardization of a single protocol for all was built using the mapping of IEC 61850 to CORBA and using the Interface Definition Language (IDL) to model the object interfaces.

Calvo et al. [15] presented a mapping of the IEC 61850 to CORBA and DDS middlewares, using CORBA for the client-server communication model functions and DDS for publish-subscribe. In this way, it was possible to respond to client requests and also to send periodic (sample measured values) and aperiodic (events) notifications. For the upper-level entities – *Servers*, *Logical Devices* and *Logical Nodes* – CORBA objects were used, while lower-level entities – *Data*, *Data Attributes* and *Data Sets* – were mapped to internal objects. DDS was used to map *Generic Substation Events (GSEs)*, *Sampled Measured Values (SMVs)* and reporting and logging services. Calvo et al. measured and reported on the jitter and the latency of the communications. Continuating this work, in [16], Calvo et al. used only DDS, configuring DDS mechanisms to accommodate the needs of both client-server and publish-subscribe models. For modeling the data, in this work the authors also use IDL, inherited from CORBA.

Bi et al. [17] point out that the mappings using MMS, CORBA and Web Services only work over the local network of individual substations not capable of interconnecting substations. They propose JMS and DDS alternatives and focus on a DDS mapping. The authors do not only provide a mapping proposal but also discuss a viability study for the project, a summary of the covered functionality and results from a simple reliability benchmark (the ratio of received to sent packages). However, this work lacks a thorough explanation of the mapping itself.

To the best of our knowledge, the first proposal for using a more generic communication protocol was presented by Pedersen et al. [18], who propose a RESTful (REpresentational State Transfer) approach using HTTP. In their proposal, they map some of the functions of the standard using XML as the format for the data. Both HTTP and XML have been designed for all kinds of environments and devices with rather large resource capacities, and therefore might not be suited for resource-constrained environments – in these scenarios, more lightweight protocols and data formats should be preferred. Along with the mapping, the authors present two case studies: interfacing with a *micro Combined Heat and Power* (μCHP) system and interfacing with an *Electric Vehicle*, but they do not provide enough information to convincingly demonstrate the suitability of their protocol in those scenarios.

In [19], Parra uses HTTP-REST to map the IEC 61850 standard, but uses the more lightweight JSON format for data representation. It maps more services than Pedersen et al. [18], but it still does not include setting, eventing, value transmission, logging and file transfer services. For validating the proposal, the article mentions three applications domains, namely a Smart Home prototype, an Ambient Media prototype and a Health Management prototype.

Hussain et al. [20] model a Distribution STATic COMpensator (DSTATCOM) and then represent the information in XML, to send the XML via XMPP. They map some basic ACSI, Eventing and Sampled Value Transmission services but they do not provide performance metrics of their proposal.

To our knowledge, Shin et al. [21] are the first to map IEC 61850 to a communication protocol specifically designed for IoT environments, in this case CoAP. They propose a simple but robust mapping on the methods. The authors map the basic functions being the main contribution of the paper a set of simulations to diagnose the performance comparing it against SOAP and MQTT implementations. To compare the performance, the authors measured the sent packets per second, the data size, the received traffic, and the network delay. In [12], Iglesias-Urkia et al. state that RESTful principles are however not followed in Shin's proposal arguing that different URIs are used for the same resource instead of making use of CoAP's request methods. Indeed, Iglesias-Urkia et al. propose a more complete mapping of the IEC 61850 to CoAP [12]. All the basic, reporting, logging, settings, eventing, sample value transmission and additional services are included in this mapping, but no validation or performance tests are provided. This paper is a continuation of this work, as it implements the mapping proposed in [12] and tests its performance against other communication protocol implementations.

As previously stated, Table I summarizes the mappings in the existing literature, presenting the protocols, data representation formats, application domains, the measured metrics and the number of mapped functions out of the 61 services

Authorized licensed use limited to: University of St. Gallen. Downloaded on April 29,2023 at 16:00:43 UTC from IEEE Xplore. Restrictions apply.

TABLE I
IEC 61850 MAPPINGS: COMMUNICATION PROTOCOL, DATA FORMAT, APPLICATION DOMAIN, METRICS AND NUMBER OF MAPPED FUNCTIONS.

Authors	Protocol	Data format	Application Domain	Metrics	# functions
Sanz et al. [14]	CORBA	IDL	-	-	-
Calvo et al. [15]	DDS + CORBA	IDL	-	Jitter, latency	-
Calvo et al. [16]	DDS	IDL	-	-	-
Bi et al. [17]	DDS	IDL	-	Reliability (Received/sent)	-
Pedersen et al. [18]	HTTP-REST	XML	μCHP & Electric Vehicle	-	16/61
Parra [19]	HTTP-REST	JSON	Smart Home, Ambient Media & Health Management	-	18/61
Hussain et al. [20]	XMPP	XML	Distribution STATic COMpensator (DSTATCOM)	-	11/61
Shin et al. [21]	CoAP	-	OPNET simulator	Packets/second, data size, traffic, delay	7/61
Iglesias-Urkia et al. [12]	CoAP	-	-	-	54/61
Present work	CoAP	JSON	Smart Elevator	Latency, overhead	23/61

defined in the standard. Despite the disparate number of previous works, only the MMS mapping is included in the standard [2]. Hence, our goal in this work is to implement the proposal presented in [12] to provide a validation and a performance comparison against HTTP-REST and SOAP Web-Services approaches. A direct comparison against other implementations is not completely fair as each individual system's performance would have an important impact on the latency and so would the selected hardware, instead of the mapping or the communication protocol itself.

III. TECHNICAL BACKGROUND

After reviewing the related work, in this section we present the technical background for the project, with an introduction to the IEC 61850 standard and the communication protocols we analyze, focusing on CoAP.

A. The IEC 61850 Standard

The IEC defines an architecture for electrical power systems and IEC 61850 [1] as the standard to model, control and monitor electrical substations. It includes a computational unit called Intelligent Electronic Device (IED), which is the entry point for the system. IEC 61850 has evolved to also support monitoring, control and protection of Smart Grid applications.

IEC 61850 defines the model and communications for electrical substations and it defines an *Information Model*, a set of services to interact with the model and a set of mappings to specific communication protocols [22].

There are two classes for building the *Information Model*, the *Basic Information Model* (BIM) and *Control Blocks for Additional Functions* (CBs). The BIM defines how to organize the elements and their information, while the CBs are specialized classes for setting and managing additional functionality. Fig. 1 presents the elements of both BIM and CBs, with the CBs that have been left out of this implementation in grey.

The elements of the BIM are the servers, which expose the IED to the outside; Logical Devices (LD), virtual representation of real devices; Logical Nodes (LN), virtual abstraction of each functionality, including a special *Logical Node Zero (LLN0)*, which is used to represent the common data of the logical device; Data items to represent information about the physical world; DataAttributes, which are typed information pieces of a Data instance; and the Dataset, a group of *Data*.

The CBs cover Reporting, Logging, Configuration, Eventing and Sampled Value Transmission. In this work, we focus on Reporting CBs, *Buffered (BRCP)* and *Unbuffered (URCB) Report Control Blocks*, as they are the implemented CBs in the IEC 61850 tool presented in Section I. All the elements in the BIM and the CBs have an absolute reference, a unique identifier to identify them in the system.

B. Communication Protocols

In this project, we implemented IEC 61850 over CoAP, and then compared to HTTP-REST and SOAP Web-Services implementations. This section presents these protocols, covering CoAP first and then providing shorter overviews of HTTP-REST and SOAP Web-Services.

1) CoAP: Addressing the need for a RESTful protocol for constrained devices, the IETF CoRE Working Group designed CoAP [7], presenting its standardized version in June 2014 as RFC 7252. Several extensions have been or are being standardized to extend the capabilities of CoAP. In the context of this project, we make use of i) *Constrained RESTful*

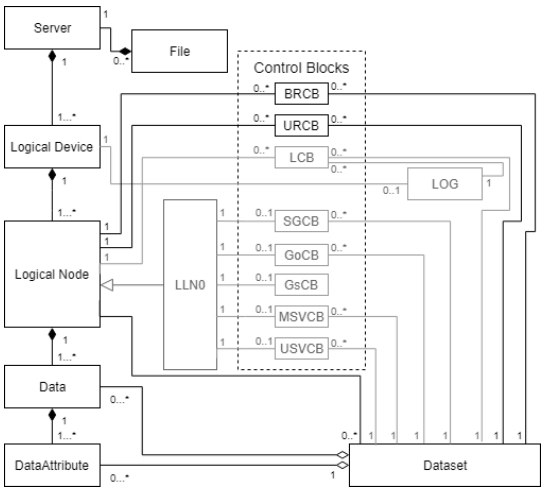


Fig. 1. Basic information model of the IEC 61850 standard.

Authorized licensed use limited to: University of St. Gallen. Downloaded on April 29,2023 at 16:00:43 UTC from IEEE Xplore. Restrictions apply.

Environments (CoRE) Link Format to define content formats[1], ii) *Observing Resources in CoAP* to enable push notifications[2], and iii) *Block-Wise Transfers in the Constrained Application Protocol (CoAP)* to be able to send large payloads[3].

CoAP adopts a client-server interaction model and uses GET, POST, PUT and DELETE requests, on top of UDP. Regarding its response codes, 2.XX signify correct responses, 4.XX indicate client errors and 5.XX are used for server errors. The default UDP port is 5683, and 5684 is used for secure usage using DTLS. CoAP resources are addressed using a URI in the format *coap://host[:port][path][?query]*.

Since UDP does not offer reliability in packet ordering and delivery guarantees, CoAP addresses that issue on the application layer, using Confirmable (CON) and NON-Confirmable (NON) messages. The former requires acknowledgments while the latter does not.

2) HTTP-REST: REST [23] is an architectural style that is most often implemented using the HTTP protocol. It follows the client-server interaction model and is stateless, which implies that every request is self-contained. REST uses different verbs (e.g., GET, POST, PUT, etc. in the case of HTTP) to address resources (identified using URIs), and resource representations can be transmitted in different formats, e.g. as HTML, XML or JSON. Links between resources can be discovered at run time from received responses, enabling clients to navigate resources without a-priori knowledge of the hypermedia structure.

3) SOAP Web-Services: In SOAP Web-Services [24], client requests and service response objects are encapsulated using the Simple Object Access Protocol (SOAP) and transmitted over the network, most often using the HTTP protocol (usually using HTTP POST requests). Services can use the Web Services Description Language (WSDL) to declare their functionality and interfaces for a-priori consumption by clients. Further standards from the WS-* family can be used to define addressing schemes, security parameters, and discovery services. Work has been done to adapt WS-SOAP to the needs of resource-constrained devices, such as the work presented by Priyantha et al. [25].

IV. Implementation of the Mapping

After describing the IEC 61850 standard and the protocols used in this project, this section explains the experimental setup. The servers are generated automatically with an internal proprietary tool in our institution that allows to create data models and then automatically generate code for HTTP-REST and SOAP servers of the IEC 61850 based on the created model. A screenshot of the tool can be seen in Fig 2.

The generated code is created with different layers as it can be seen in Fig. 3. First, there is a kernel that includes the core functions of the IEC 61850. On top of that, there are two libraries: `lib-service-server-rest` for RESTful implementations and `lib-service-server-soap` for SOAP Web-Services. The uppermost layer comprises the servers, i.e. the HTTP-REST server and the Web-Services SOAP server. The former uses the `microhttpd` [26] library, whilst the latter uses `gsoap` [27]. A CoAP server layer has been added for this project, which also builds the entire CoAP resource tree automatically from the defined model. For the reporting services in HTTP-REST and Web-Services SOAP, new servers and clients are developed, with reversed roles. First, the service server stores the reports on a folder. Then the reporting client checks periodically whether there is a report on the folder and if there is, it sends the report to the reporting server. In CoAP this is not necessary due to its *Observe* extension, which allows to send notifications to subscribed clients and enables clients to subscribe to notifications. In our solution, the main application checks whether there is any report in the report folder and if there is, it sends a push notification to the subscribed client.

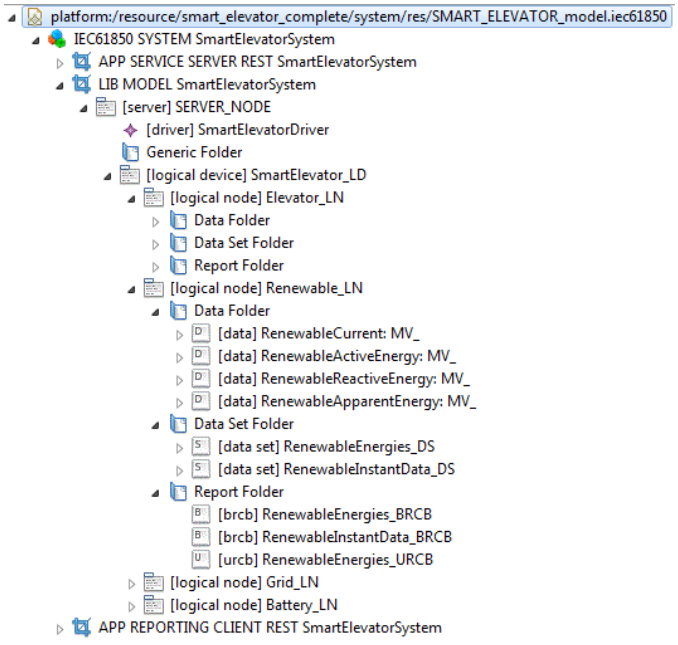


Fig. 2. Screenshot of the IEC 61850 tool.

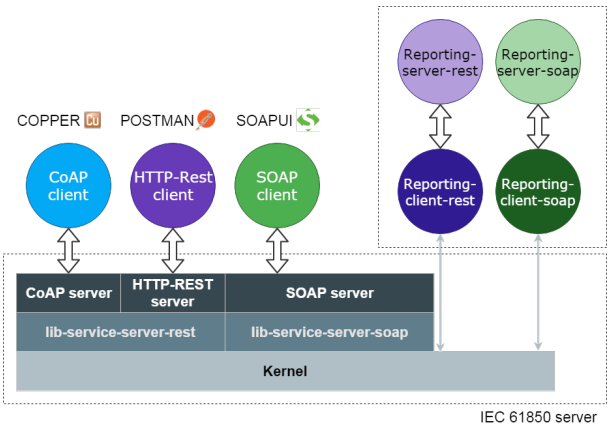


Fig. 3. IEC 61850 tool layers.

[1]https://tools.ietf.org/html/rfc6690
[2]https://tools.ietf.org/html/rfc7641
[3]https://tools.ietf.org/html/rfc7959

Authorized licensed use limited to: University of St. Gallen. Downloaded on April 29,2023 at 16:00:43 UTC from IEEE Xplore. Restrictions apply.

There are many CoAP implementations for different environments [28]. The develop branch version of Feb. 5, 2018 of `libcoap` [29] was selected to implement a subset of the services proposed in [12], based on the services present in the IEC 61850 generating tool.

The services that have been implemented include different types of messages, including getters, setters, notifications and subscriptions. Table II summarizes the implemented services, giving the names of the IEC 61850 functions together with the mapped URI and CoAP request method, along with an identifier for the result table and chart.

Two changes have been considered necessary compared to the mapping presented in [12]: First, *SetFile* has been divided into two different functions depending on whether the file previously existed or not. If it did, a PUT request is sent to the resource pointing to the file, and if it did not, a POST request is sent to */Files*. Second, in *GetDataValues*, a query with the functional constraint is necessary in order to integrate the solution with the existing system.

V. Validation of the Implementation

The server runs on a Raspberry Pi 3 [30], a small single-board computer that is used more and more in industry scenarios. Its CPU is a Quad Core ARM Cortex A53 capable of running at 1.2 GHz and 1 GB of RAM at 900 MHz. The selected Operative System has been Raspbian Stretch installed on a 16 GB class 10 microSD card. Even though the setup for a big scale deployment would not use Raspberry Pi-s, the measured metrics would be proportional for the different protocols making this set-up suitable and appropriate for validation in a laboratory environment.

Fig. 4 represents the model used for the experiments. In purple, the LD, in green the LNs and in orange the CBs. The individual CBs have been left out due to space constraints.

Some of the functions have been tested with different parameters which makes the results to significantly vary due to the different sizes of the sent information.

For running clients, a a Windows 7 PC has been used. The clients are Postman 6.0.10[4] for HTTP-REST, SoapUI 5.4.0[5] for Web-Services SOAP, and Copper 1.0.1 [31] over Firefox 49.0.2 for HTTP-REST for CoAP. The servers and clients that generate the IEC 61850 code generating tool have been used for HTTP-REST and Web-Services SOAP notifications. Wireshark[6] network protocol analyzer has been the selected tool to gather the metrics of the communications.

[4]https://www.getpostman.com/

[5]https://www.soapui.org/

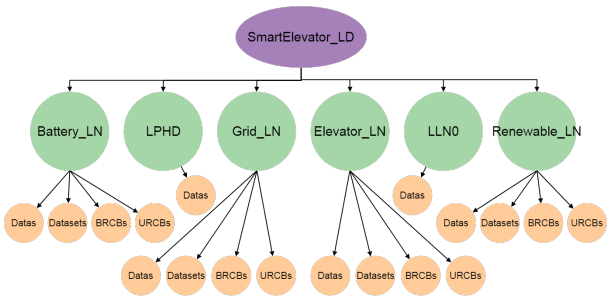


Fig. 4. Model used in the experimentation.

Firstly, we demonstrate that the mapping presented in [12] is suitable with the modifications described in Section IV. Reporting is carried out in an more simple way with CoAP, as the Observe extension can be used to send notifications to subscribed clients. The reporting system in HTTP-REST and Web-Services SOAP is complex as it adds extra components and does not use the publish-subscribe communication pattern, more suitable for this use case.

Secondly, Table III shows the maximum and median latencies in ms for the three protocols. Reporting subscription (functions number 21 and 26) and notifications (functions number 17, 18, 22 and 23) have been left out due to the fact that WS-SOAP and HTTP-REST implementations do not include subscription functions, the clients poll the reports. The table shows that in general, CoAP's median value is the lowest between 6 and 8 ms for the majority of functions, with the exception of some peaks that appear when the block-wise transfer is needed, see functions 8 and 9, where many blocks are use and mean latency goes as high as 4003,6 and 3260,1 ms respectively. The block-wise transfer uses a 'stop-and-wait' method to send a large payload, instead of TCP's sliding window protocol, where more than one segment can be sent without waiting for the previous ACKs. SOAP also performs well in some cases, specially where CoAP does not. REST is in the middle for most cases, with some negative peaks.

Lastly, we measured the bytes exchanged in the communication process and the overhead. It has to be taken into account that TCP requires an open channel to send messages, and we assume that the channel is already open. This means that we do not count TCP handshake nor the session closing exchange. Besides, we also do not count keep-alive messages. Fig. 5 shows the results of data and overhead bytes. Reporting subscription messages have been left out because these messages are not part of the IEC 61850 standard and SOAP and HTTP-REST do not include them as their clients do not subscribe. Release and Abort functions are not included in the WS-SOAP system, a unique function named Logoff is implemented instead. Thus, the measurements are the same in the charts for both Release and Abort.

In this figure, it is clear that XML representation of the objects used in the WS-SOAP implementation is much larger than JSON representation used in HTTP-REST and CoAP. Also, the verbosity of the SOAP protocol adds more overhead than the other options. As both HTTP-REST and CoAP use the same JSON representation, the data parts of the messages are exactly the same. The overhead is generally much smaller with CoAP, but when the Blockwise Transfer extension is used, the differences start to decrease, and if many blocks are needed, CoAP has more overhead than HTTP-REST, see functions 8

[6]https://www.wireshark.org/

Authorized licensed use limited to: University of St. Gallen. Downloaded on April 29,2023 at 16:00:43 UTC from IEEE Xplore. Restrictions apply.

TABLE II
IMPLEMENTED IEC 61850 SERVICES WITH COAP COMMUNICATION PROTOCOL.

Function	URI	Method	Id	Notes
Basic				
Server				
GetServerDirectory	coap://{host}/LDs	GET	1	Logical Devices
GetServerDirectory	coap://{host}/Files	GET	2	Files
Logical Device				
GetLogicalDeviceDirectory	coap://{host}/LDs/{LD}	GET	3	
Logical Node				
GetLogicalNodeDirectory	coap://{host}/LDs/{LDs}/{LN}/{ACSIClass}	GET	4 5 6 7	Datas Datasets BRCBs URCBs
GetAllDataValues	coap://{host}/LDs/{LDs}/{LN}/AllValues[?FC={fc}]	GET	8 9 10	Big payload Medium payload Small payload
Data				
GetDataValues	coap://{host}/LDs/{LDs}/{LN}/Datas/{Data}	GET	11	
SetDataValues	coap://{host}/LDs/{LDs}/{LN}/Datas/{Data}	PUT	12	
GetDataDirectory	coap://{host}/LDs/{LDs}/{LN}/Datas/{Data}/Directory	GET	13	
GetDataDefinition	coap://{host}/LDs/{LDs}/{LN}/Datas/{Data}/Definition	GET	14	
Dataset				
GetDataSetValues	coap://{host}/LDs/{LDs}/{LN}/Datasets/{Dataset}	GET	15	
GetDataSetDirectory	coap://{host}/LDs/{LDs}/{LN}/Datasets/{Dataset}/Directory	GET	16	
Reporting				
BRCB				
Report	coap://{host}/LDs/{LDs}/{LN}/{BRCBs}/BRCB/Reports	Notification	17 18	Big payload Small payload
GetBRCBValues	coap://{host}/LDs/{LDs}/{LN}/{BRCBs}/BRCB	GET	19	
SetBRCBValues	coap://{host}/LDs/{LDs}/{LN}/{BRCBs}/BRCB	PUT	20	
if enable	coap://{host}/LDs/{LDs}/{LN}/{BRCBs}/BRCB/Reports	GET+Obs	21	
URCB				
Report	coap://{host}/LDs/{LDs}/{LN}/{URCBs}/URCB/Report	Notification	22 23	Big payload Small payload
GetURCBValues	coap://{host}/LDs/{LDs}/{LN}/{URCBs}/URCB	GET	24	
SetURCBValues	coap://{host}/LDs/{LDs}/{LN}/{URCBs}/URCB	PUT	25	
if enable	coap://{host}/LDs/{LDs}/{LN}/{URCBs}/URCB/Report	GET+Obs	26	
Additional Services				
Access Control				
Associate	coap://{host}/Associations	POST	27	
Abort	coap://{host}/Associations/{AssID}?action=abort	DELETE	28	
Release	coap://{host}/Associations/{AssID}?action=release	DELETE	29	
File Transfer				
GetFile	coap://{host}/Files/{File}	GET	30 31	Big file Small file
SetFile (update existing)	coap://{host}/Files/{File}	PUT	32	
SetFile (new file)	coap://{host}/Files	POST	33	
DeleteFile	coap://{host}/Files/{File}	DELETE	34	
GetFileAttributeValules	coap://{host}/Files/{File}/Status	GET	35	

and 9. This is due to the fact that each CoAP message needs to have the entire header as it chunks only the payload, while HTTP chunks the entire message, header included.

VI. DISCUSSION

After analyzing the implementation of the IEC 61850 standard and its performance, some conclusions can be provided. On the one hand, CoAP proves to be a good fit for implementing the IEC 61850 as with the Observe extension it is capable of communicating using a publish-subscribe communication pattern, besides its regular client-server. This way, the implementation of the IEC 61850 functions that require this paradigm is much more simple and natural. On the other hand, the bytes required for the communications are much less than WS-SOAP and the overhead is also smaller than the HTTP-REST implementation when not many blocks are needed for a block-wise communication. In terms of latency CoAP performs better in most cases. However, the block-wise transfer penalizes a lot when it is needed, specially when many blocks are required. Reducing the overhead is of vital importance for some Smart Grid systems, where some devices can be very remote and their connectivity very limited. This connectivity issues can make for some parts of the network to run as slow as 1500 bytes/s, and not so big messages can timeout and retransmits begin, amplifying the problem.

The Observe extension has some limitations to perfectly match the IEC 61850 mapping. When setting a BRCB or

Authorized licensed use limited to: University of St. Gallen. Downloaded on April 29,2023 at 16:00:43 UTC from IEEE Xplore. Restrictions apply.

TABLE III
MAX AND MEDIAN LATENCIES IN MS WITH COAP, HTTP-REST AND SOAP WEB-SOCKETS.

Id	Maximum			Median		
	CoAP	HTTP	SOAP	CoAP	HTTP	SOAP
1	22,2	32,3	12,7	5,9	5,4	5,1
2	93,7	69,7	38,2	6,2	7,6	5,8
3	67,4	31,7	212,4	7,0	7,7	8,0
4	38,3	76,8	20,7	12,0	8,2	6,9
5	26,8	226,6	65,4	6,5	5,8	9,1
6	44,9	36,1	103,2	6,8	7,1	8,8
7	67,3	82,0	97,7	6,4	5,2	7,0
8	8354,2	484,9	297,4	4003,6	64,2	75,5
9	5291,4	312,5	304,1	3460,1	256,2	73,5
10	231,8	27,7	25,0	14,4	7,2	8,8
11	25,3	155,5	42,3	6,5	10,0	11,6
12	105,9	299,5	77,6	6,2	209,6	8,7
13	57,6	93,0	28,3	7,4	10,5	7,7
14	597,6	102,9	31,6	390,6	13,1	13,7
15	31,0	46,7	220,8	6,4	6,3	9,1
16	73,7	94,6	23,8	10,5	7,4	6,8
19	29,7	42,4	25,2	9,6	8,5	8,6
20	21,0	677,7	76,4	6,2	209,3	9,5
24	32,7	245,0	24,3	8,2	209,5	9,9
25	22,1	103,4	32,9	5,9	7,7	7,8
27	83,0	103,1	65,0	12,2	8,3	5,4
28	37,7	97,8	23,5	8,8	6,2	5,1
29	30,7	14,5	23,5	10,7	5,5	5,1
30	1196,4	272,9	83,9	956,4	219,3	19,7
31	60,5	26,5	308,3	6,4	6,5	9,2
32	62,9	262,8	66,8	7,2	209,9	8,9
33	58,4	227,4	32,4	7,7	207,6	8,6
34	27,3	157,3	32,8	6,1	8,8	8,4
35	38,4	40,6	17,6	6,4	9,1	7,1

a URCB, the client can not subscribe directly, it needs an extra request, a GET with the Observe option activated. As a future line of work, new subscription mechanisms are being studied, to extend the Observe extension and be able to, for instance, send a PUT request to upload a configuration file and subscribe with that same request. Following with this exact use case, a CoAP Observe subscription response includes an entire representation of the resource, albeit it is not needed when there is still not a report. Thus, extending the Observe extension to be able to respond with a "Successfully subscribed" kind of message is also a point to be taken into account.

Another enhancement for CoAP could be a more effective Block-wise transfer model to send chunked data. In this paper, we concluded that when the number of blocks needed to send a data representation is high, the HTTP-REST implementation outperforms the CoAP implementation in terms of overhead. This is due to the fact that CoAP sends the entire header with each block, thus repeating the header. A path for future work lies there, searching for a more effective way to send large payloads.

Due to the need for integrating the work presented in this paper with the existing IEC 61850 code generating tool, and implementing a CoAP layer outside of the kernel, the representation of the resources has been JSON. An interesting line for future work is to complete the entire IoT stack in the implementation, changing the representation of the resources from JSON to CBOR. This way, the representation of the resources would be lighter, and each of the three implementations (HTTP-REST, WS-SOAP and CoAP) would have its own stack, i.e., TCP+HTTP+JSON for HTTP-REST, TCP+HTTP+XML for WS-SOAP and UDP+HTTP+CBOR for CoAP. This would require to change the kernel of the IEC 61850 generating tool which was out of scope for this work.

The work presented in this paper has been deployed on a lab prototype with one-on-one communication on a dedicated WiFi network. Next steps would be to create a simulation with several nodes to later deploy the implementation in real hardware available on electrical substations, and review the performance in a more congested network.

Finally, a last proposed line of work could be to propose a mapping to other IoT protocols such as MQTT or AMQP, and compare their performance. However, the suitability of these protocols in this use case needs to be studied to check whether the features fit the requirements for the IEC 61850.

VII. CONCLUSION

In this paper, it is presented an implementation of the IEC 61850 standard to CoAP, based on the work presented in [12]. First, the previously existing mappings to different network protocols are briefly reviewed. After that, the IEC 61850 is summarized, along with the CoAP as the protocol used for this work. As the implementation presented in this work is built over an existing proprietary tool that automatically generates code from a model, using HTTP-REST and WS-SOAP, these last two paradigms are also explained. Then, the implementation itself is explained, along with the validation, comparing the performance against HTTP-REST and WS-SOAP in terms of overhead and latency. In the discussion, we conclude that the mapping is effective for most functions. It has to be noted that thanks to the Observe extension and its publish-subscribe paradigm, CoAP proves to be a good fit for a IEC 61850 mapping due to the standard needing both client-server and publish-subscribe communications. Finally, future lines of work are also proposed, specially focusing on enhancing CoAP for better performance on the IEC 61850 functions that do not perform so effectively.

ACKNOWLEDGMENT

This work has been partially supported by the Basque Government through the Elkartek program under the LANA II project (Grant agreement no. KK-2016/00052), as well as the Spanish Government and the H2020 research framework of the European Commission.

REFERENCES

[1] IEC TC-57, "Communication networks and systems in substations – part 7-1: Basic communication structure for substation and feeder equipment – principles and models," 2003.

Authorized licensed use limited to: University of St. Gallen. Downloaded on April 29,2023 at 16:00:43 UTC from IEEE Xplore. Restrictions apply.

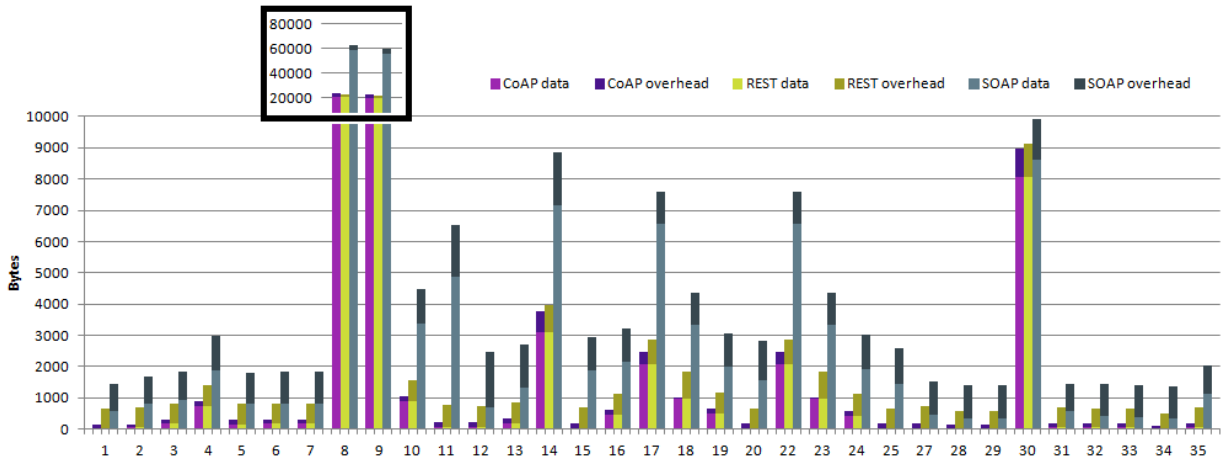


Fig. 5. Data and overhead bytes.

[2] ——, "Communication networks and systems in substations – part 8-1: Specific communication service mapping (scsm) – mappings to mms (iso 9506-1 and iso 9506-2) and to iso/iec 8802-3," 2003.

[3] ——, "Communication networks and systems in substations – part 9-1: Specific communication service mapping (scsm) – sampled values over serial unidirectional multidrop point to point link," 2003.

[4] ——, "Communication networks and systems in substations – part 9-2: Specific communication service mapping (scsm) – sampled values over iso/iec 8802-3," 2003.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[6] M. Iglesias-Urkia, A. Orive, M. Barcelo, A. Moran, J. Bilbao, and A. Urbieta, "Towards a lightweight protocol for industry 4.0: An implementation based benchmark," in *Proceedings of the 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics, ECMSM 2017*, 2017.

[7] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014, https://tools.ietf.org/html/rfc7252.

[8] A. Banks and R. Gupta, "Mqtt version 3.1. 1," *OASIS standard*, 2014.

[9] B. Cabé, "Key trends from the iot developer survey 2018," 2018, https://blog.benjamin-cabe.com/2018/04/17/key-trends-iot-developer-survey-2018.

[10] Pimoroni, "Controlling ikea trådfri lights from your pi," 2018, https://learn.pimoroni.com/tutorial/sandyj/controlling-ikea-tradfri-lights-from-your-pi.

[11] MQTT.org, "Mqtt used by facebook messenger," 2011, https://mqtt.org/2011/08/mqtt-used-by-facebook-messenger.

[12] M. Iglesias-Urkia, A. Urbieta, J. Parra, and D. Casado-Mansilla, "Iec 61850 meets coap: Towards the integration of smart grids and iot standards," in *Proceedings of the Seventh International Conference on the Internet of Things*, ser. IoT '17. New York, NY, USA: ACM, 2017, pp. 3:1–3:9.

[13] IEC TC-88, "Wind energy generation systems - part 25-4: Communications for monitoring and control of wind power plants - mapping to communication profile," 2016.

[14] R. Sanz, J. A. Clavijo, M. J. Segarra, A. de Antonio, and M. Alonso, "CORBA-Based Substation Automation Systems," in *Proceedings of IEEE Conference on Control Applications*, 2001.

[15] I. Calvo, O. García De Albéniz, A. Noguero, and F. Pérez, "Towards a modular and scalable design for the communications of electrical protection relays," *IECON Proceedings (Industrial Electronics Conference)*, pp. 2511–2516, 2009.

[16] I. Calvo, O. Garcia de Albéniz, and F. Pérez, "A communication backbone for Substation Automation Systems based on the OMG DDS standard," in *Przeglad Elektrotechniczny*, vol. 88, no. 1 A, 2012, pp. 146–150.

[17] Y. Bi, L. Jiang, X. J. Wang, and L. Z. Cui, "Mapping of IEC 61850 to Data Distribute Service for digital substation communication," in *IEEE Power and Energy Society General Meeting*, 2013, pp. 1–5.

[18] A. B. Pedersen, E. B. Hauksson, P. B. Andersen, B. Poulsen, C. Træholt, and D. Gantenbein, "Facilitating a Generic Communication Interface to Distributed Energy Resources: Mapping IEC 61850 to RESTful Services," *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 61–66, 2010.

[19] J. Parra, "Restful Framework for Collaborative Internet of Things Based on IEC 61850," Ph.D. dissertation, Universidad del País Vasco - Euskal Herriko Unibertsitatea (UPV/EHU), 2016.

[20] S. M. S. Hussain, M. A. Aftab, and I. Ali, "Iec 61850 modeling of dstatcom and xmpp communication for reactive power management in microgrids," *IEEE Systems Journal*, pp. 1–11, 2018.

[21] I.-J. Shin, B.-K. Song, and D.-S. Eom, "International Electronical Committee (IEC) 61850 Mapping with Constrained Application Protocol (CoAP) in Smart Grids Based European Telecommunications Standard Institute Machine-to-Machine (M2M) Environment," *Energies*, vol. 10, no. 3, p. 393, 2017.

[22] IEC TC-57, "Communication networks and systems in substations – part 7-2: Basic communication structure for substation and feeder equipment – abstract communication service interface (acsi)," 2003.

[23] R. T. Fielding, "REST: architectural styles and the design of network-based software architectures," Doctoral dissertation, University of California, Irvine, 2000.

[24] D. Box, G. Kavivaya, A. Layman, S. Thatte, and D. Winer, "Soap: Simple object access protocol," 1999, https://www.ietf.org/archive/id/draft-box-http-soap-01.txt.

[25] N. Priyantha, A. Kansal, M. Goraczko, and F. Zhao, "Tiny web services: Design and implementation of interoperable and evolvable sensor networks," in *SenSys'08 - Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems*, 2008, pp. 253–266.

[26] F. S. Foundation, "Gnu libmicrohttpd," 2018, https://www.gnu.org/software/libmicrohttpd/.

[27] R. van Engelen, "gsoap 2.8.66 user guide," 2018, https://www.genivia.com/doc/soapdoc2.html.

[28] M. Iglesias-Urkia, A. Orive, A. Urbieta, and D. Casado-Mansilla, "Analysis of coap implementations for industrial internet of things: a survey," *Journal of Ambient Intelligence and Humanized Computing*, Mar 2018. [Online]. Available: https://doi.org/10.1007/s12652-018-0729-z

[29] O. Bergmann, "libcoap: C-implementation of coap," 2017, https://libcoap.net.

[30] Raspberry Pi Foundation, "Raspberry pi," 2018, https://www.raspberrypi.org/.

[31] M. Kovatsch, "Demo abstract: Human–coap interaction with copper," in *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2011)*, Barcelona, Spain, Jun. 2011.

Authorized licensed use limited to: University of St. Gallen. Downloaded on April 29,2023 at 16:00:43 UTC from IEEE Xplore. Restrictions apply.