Simulation of IEC 61850-based substations under OMNeT++

Javier Juárez
Institute for Research in Technology
Comillas Pontifical University
C/ Santa Cruz de Marcenado 26,
28015 Madrid, Spain
javier.juarez@iit.upcomillas.es

Carlos Rodríguez-Morcillo
Institute for Research in Technology
Comillas Pontifical University
C/ Santa Cruz de Marcenado 26,
28015 Madrid, Spain
carlos.rodriguez@iit.upcomillas.es

José Antonio Rodríguez-Mondéjar
Institute for Research in Technology
Comillas Pontifical University
C/ Santa Cruz de Marcenado 26,
28015 Madrid, Spain
mondejar@dea.icai.upcomillas.es

ABSTRACT

Communication systems are essential components within the Smart Grids. Given the complexity of real world communication systems, advanced simulation platforms are required to facilitate the study, analysis, design and evaluation of such systems. The IEC 61850 is one of the most relevant communication standards concerning Smart Grids. Existing simulation platforms for IEC 61850-based systems are mainly focused on the evaluation of the performance and functionality of the communication networks. In this paper, a new IEC 61850 simulation platform is presented. It is based on OMNeT++ and it not only allows to conduct a network performance analysis, but also to carry-out hardware-in-the-loop simulations, or even to evaluate algorithms before implementing them into a real device. For that purpose, a new simulation core using two parallel processes has been developed, and a real IEC 61850 communication stack has been integrated into OMNet++.

Keywords

IEC 61850, OMNeT++, Simulation, Substation Automation System (SAS), Ethernet, Hardware-in-the-loop.

1. INTRODUCTION

Smart Grids are complex in nature and there are many issues to be considered regarding both their development and implementation. Existing power systems need a more decentralized intelligence in order to create automated systems able to achieve all the objectives that the Smart Grid demands. Therefore, communication systems play a major role in the Smart Grid development. These communication networks include an abundance of technologies (wireless, PLC, fiber optic…), protocols and standards, and needless to say the convergence of all of these technologies and protocols is a complicated and crucial issue to solve.

Given the complexity of Smart Grid communication systems, the use of open and flexible simulation platforms facilitates the design of devices, prototypes and network systems. Furthermore, these simulation platforms can also be employed for teaching and training purposes, which could be very useful for the Smart Grid development as well.

One of the main standards in the scope of Smart Grid communications is the IEC 61850 [1]. This standard is focused on the communications between intelligent electronic devices in power utility automation systems. It outlines a virtualized model that represents devices behavior. Its main objective is to achieve interoperability between machines from different vendors. Initially, the scope of this standard was substation automation systems, but recently it has been extended to other areas of the Smart Grid, such as, distribution of energy resources.

IEC 61850 substations are intricate and expensive systems. Thus, having a platform that can be used as a test-bed for developing prototypes, applications or algorithms for such systems could be very advantageous.

It is noteworthy that one type of simulation that is most frequently used in communication networks is the Discrete Event Simulation (DES). This is mainly because this simulation paradigm adapts very well to this type of systems. In general, DES gives an easy, flexible, scalable and highly repeatable way to study the behavior of these systems under different conditions.

Using open-source DES simulators, such as OMNeT++, which can be modified and adapted, not only allows to realize network performance analysis (section 4.1), but also to carry-out hardware-in-the-loop simulation (section 4.2.1). This could be very useful in order to validate prototypes and designs, or to evaluate algorithms (section 4.2.2) before applying them into a real network. In order to achieve this, it is necessary to implement a real IEC 61850 communication stack, and to adapt the simulator for conducting hardware-in-the-loop simulations in a synchronized way.

This paper is structured as follows: section 2 provides an overview on the state of the art for the simulation of IEC 61850-based Substation Automation Systems; section 3 briefly describes the IEC 61850 standard and presents the advanced simulation platform; sections 4 shows some examples about how the simulation platform can be used; and finally, section 5 concludes the paper.

2. STATE OF THE ART

In regards to the literature about Smart Grids there are several works about simulation of IEC 61850-based Substation Automation Systems (SAS). Most of them are simply focused on validating IEC 61850 network architectures. However, validating IEC 61850 devices is an important aspect too, and the new platform is focused on that.

OMNeT++ 2012, March 19-23, Desenzano del Garda, Italy

DOI 10.4108/icst.simutools.2012.247730

Some works about IEC 61850 simulation are detailed below.

Part 5 of the IEC 61850 [2] studies the performance of a switched hub Ethernet network of 10Mbps and 100Mbps with COMNET III simulator, but it does not consider message format and background traffic.

Skeie et al. [3] take advantage of such simulator to demonstrate that Ethernet meets the requirements of the IEC 61850-based systems in terms of maximum delays allowed.

Sidhu and Yin [4] introduce a generic modeling technique of intelligent electronic device (IED) and the setup of a research platform using the OPNET Modeler. They provide a detailed explanation about how to create and setup a platform for the performance study of an IEC 61850-based substation communication system by modeling devices with the specific protocol stack. This work does not use real application messages, so they have to estimate the length and number of messages to exchange.

Thomas and Ali [5] propose a network architecture for IEC 61850 SAS and demonstrate that it is reliable, fast and deterministic. They prove the suitability of Ethernet in SAS applications from the point of view of network performance. In order to do so, the network nodes are modeled according to Sidhu and Yin [4] using the OPNET simulator.

Kanabar and Sidhu [6] present the performance evaluation of the IEC 61850-9-2 process bus for a typical 345/230 kV substation by using the OPNET simulation tool. The paper proposes a sampled value estimation algorithm, which can be implemented into a protection and control IED as one of the corrective measures for sampled value lost or delay. This algorithm has been developed with the help of PSCAD/EMTDC and MATLAB simulation tools incorporating various sampled values lost and delayed scenarios obtained from OPNET.

Finally, Ozansoy et al. [7] present a detailed design and layout of implementation of a real-time publisher/subscriber communication model to satisfy the communication needs of the IEC 61850 protocol, and evaluates the performance of the model using OPNET simulator, but it does not give details about how the simulation has been made.

The papers presented above are focused on the performance of IEC 61850-based communication networks, and on the IEC 61850 requirements. However, the simulation results are very sensitive to the estimated number of exchanged messages, or to the size of the messages. Thus, it is very useful and challenging to develop a simulator able to evaluate specific situations in a more realistic way. For that reason, this paper presents a new simulation platform that will not only allow to analyze network performance, but also to implement a test-bed that facilitates the development of prototypes, algorithms and designs.

As explained previously, OPNET is the most used simulator in this area. It is a well-known simulator and it has probably the largest protocol model library among the existing simulation tools. However, for the proposed simulation platform we used OMNeT++. Apart from economic aspects, the OMNeT++ simulator is used in order to achieve the same performance as that of OPNET simulator, but with the advantage of being completely open-source. This means it can be easily modified and extended in order to improve its behavior.

3. ADVANCED SIMULATION PLATFORM

This section describes the advanced simulation platform developed in this work. For that purpose, section 3.1 briefly presents the IEC 61850 standard. Then, the description of the advanced simulation platform is exposed by addressing two main components: the simulation models, and the simulation core.

3.1 The IEC 61850 standard

IEC 61850 provides a layout of a virtualized model for data and services that represents the behavior of the devices in charge of controlling electric facilities. This data model is object-oriented and it is represented with a set of tables provided in IEC 61850 part 7. The standard has three key points. Firstly, it defines the requirements of the IEC 61850-based communication systems; secondly, it establishes a solid data model and communication service; and, finally, it specifies a configuration language which facilitates the configuration of IEC 61850 devices.

The standard separates the data model from communication services and the protocol stack. It has some advantages, such as the possibility that a new communication technology can be integrated in the model without changing the standard model.

The elements contained in electric facilities are modeled in the IEC 61850 as Logical Nodes (LN). For example, the LN defined in the standard for representing switches with short circuit breaking capability is called XCBR. The LNs are contained in Logical Devices (LD), which in turn are contained in the IEC 61850 devices connected in the SAS. For instance, an LD can represent a bay unit, i.e. a device that controls a bay of the substation. A bay is a set of pieces of conducting equipment within a substation that are grouped according to functional reasons.

Summarizing, the IEC 61850 standard establishes:

- Functions and information visible within the network. How it is named and how it is described.
- How this information can be accessed and exchanged.
- How the devices are connected to the communication network.

Other important issues to be considered in IEC 61850 standard are the Generic Object Oriented Substation Event (GOOSE) messages. This type of messages is used to exchange information about any change of state in one device, and the time it happened. This information could be, for example, a detected fault by a protection relay.

Due to the fact that services and objects defined in IEC 61850 are abstract, it is necessary to map them into specific protocols. IEC 61850 can be mapped into any protocol, although it is a very difficult process given the complexity of the data model. For that reason, the IEC 61850-8-1 standardizes the mapping of the abstract object and services to the Manufacturing Message Specification (MMS) protocol defined in ISO 9506. It is the only public protocol that easily supports the complex naming and service models of IEC 61850. This protocol was designed to transfer real time data and supervise control information between industrial devices.

The protocol stack used by IEC 61850 is shown in Fig. 1. Sampled Values (transmission messages for analog measurements) and GOOSEs should be fast and reliable, and they have strict time requirements, thus they are based on Ethernet technology. MMS messages have not so strict time requirements and they are implemented over TCP/IP. MMS protocol is based on the Open Systems Interconnection (OSI) model, and it is necessary to use an intermediate layer (RFC 1006 layer) in order to emulate OSI services over TCP/IP. MMS application protocol is specified in Abstract Syntax Notation Number One (ASN.1) format that is a notation for describing data structures.

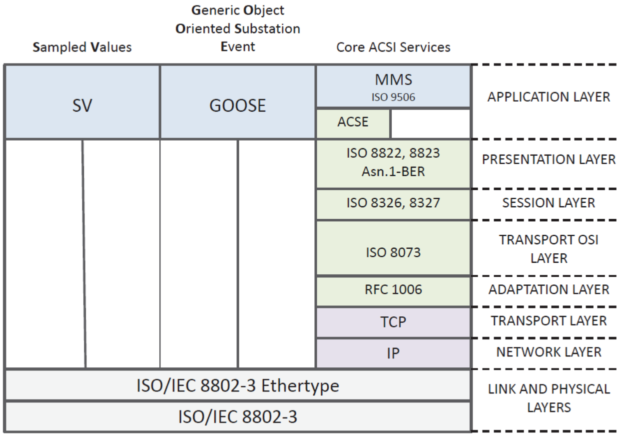

Fig. 1. IEC 61850 protocol stack.

3.2 IEC 61850 simulation models

The simulation models describe the behavior of all the components of the network, i.e.: its devices, protocols and applications. In addition, they simulate the ways in which these components interact within the network. In the case of IEC 61850, the simulation models depict the protocol stack, the message format, and the behavior of the devices.

In this work, ISO/IEC 8802-3 (with some modifications in order to support IEEE 802.1Q standard), TCP and IP layers were imported from INETMANET framework, an open-source communication networks simulation package. Moreover, a real MMS stack (from RFC 1006 layer to MMS application layer) has been integrated into OMNeT++ using an external library that has been developed with open-source resources. This library has been validated through an interoperability test using an MMS stack designed by SISCO Company.

Two types of devices have been developed:

- The Merging Unit (MU), which is a device that functions as an interface between electronic measurement transformers and control and protection devices, by merging the sampled data of the measured values.
- A generic device for protection and control functions. The modules that it uses are: Ethernet interface, network layer, TCP layer, Sniffer module and Application module. The purpose of the Sniffer module is to monitor all the traffic that passes through it and to record the messages in a PCAP file, in order to analyze them with Wireshark. It is an open-source network protocol analyzer. This module allows to check the correctly formatted messages and to evaluate in detail what is happening in the simulation.

Both models can be freely downloaded [9].

With the aim of validating the simulation models presented in this paper, the tests defined in Sidhu and Yin [4] were reproduced. They select a 220 kV substation of type D2-1 with two transformer bays, and simulates two basic architectures (star and ring). With respect to the data flow, the following scenario is assumed by the authors: the metering devices are sending Sampled Values at specified sampling rate (960, 1920 and 4800 samples/s); the protection and control units send GOOSEs messages continuously to its corresponding breakers; and both the protection and control devices and the breakers send MMS messages with their status to the station server at a rate of 20 Hz.

The tests reproduced here only evaluate the star architecture in the 100 Mbps LAN case. The results are shown in Table I along with the error regarding the results of Sidhu and Yin [4].

Table I. Simulation results

		No priority tagged	No priority tagged	Priority tagged	Priority tagged
	Sampling Rate [Samples/s]	**Max. [ms]**	**Error [ms]**	**Max. [ms]**	**Error [ms]**
Sampled Values: 98 bytes	960	0.023	0	0.023	0
	1920	0.023	0	0.023	0
	4800	0.023	0	0.023	0
Sampled Values: 52 bytes	960	0.016	-	0.016	-
	1920	0.016	-	0.016	-
	4800	0.016	-	0.016	-
GOOSEs intrabay: 16 bytes	960	0.014	-0.001	0.014	0
	1920	0.014	-0.001	0.014	0
	4800	0.014	-0.001	0.014	0
GOOSEs interbay: 16 bytes	960	0.035	0	0.035	0.001
	1920	0.035	0	0.035	0.001
	4800	0.035	0	0.035	0.001

Considering that switches have a latency delay from 1μs to 10μs approximately, depending on the manufacturer and its characteristics, and that the maximum error obtained in the simulation with regards to the results of Sidhu and Yin [4] is about ±1μs, the results of the tests can be considered satisfactory and the models are successfully validated.

3.3 Simulation core

The simulation core describes how the simulator uses the simulation models in order to carry-out the simulation. It consists of two elements: an event list, where the events are stored; and a scheduler that selects the next event in the event list to be executed.

OMNeT++ has three schedulers defined, and allows the implementation of new ones. The basic scheduler defined is the cSequentialScheduler, whose behavior is simple: it works in a loop in which it first selects the first element in the event list, then it executes the routine of this event, and finally it deletes it from the event list. The second scheduler is cRealTimeScheduler. It uses wait calls that synchronize the simulation time to the system clock. However, there are a limit for this synchronization that is given by the ratio between simulation time and real time, whose value needs to be close to 1. (Fig. 2 illustrates this concept).

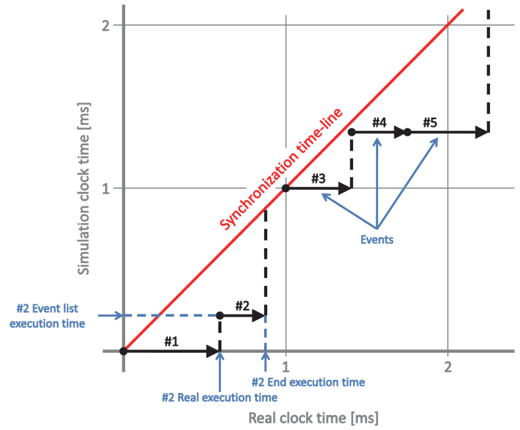

Fig. 2. Simulation time vs real time.

The Synchronization time-line represents when simulation time and real time are completely synchronized; but this is unrealistic since OMNeT++ is a discrete event simulator, and sometimes there will be a delay between simulation and real time because a CPU only allows a limited number of events to be handled in real-time.

The third scheduler is cSocketRTScheduler. Its behavior is similar to cRealTimeScheduler, but during wait calls the scheduler waits for incoming messages from an external device using raw sockets.

For hardware-in-the-loop simulations, since these schedulers are sequential, there may be synchronization problems between the simulator and the real network devices, which cause delays in the transmission of messages and therefore could make the results unrealistic. Anew simulation core has been created in order to solve the above mentioned problems. Furthermore, it can be easily modified for adapting its behavior to the real requirements of systems. This simulation core uses two parallel processes, which means that a computer with two cores is required. That being said, it can be extended for using more cores. The first process manages the main simulation, in other words, the event list and the main scheduler. The second process manages the connection with the real network. Fig. 3 shows the structure of this simulation core.

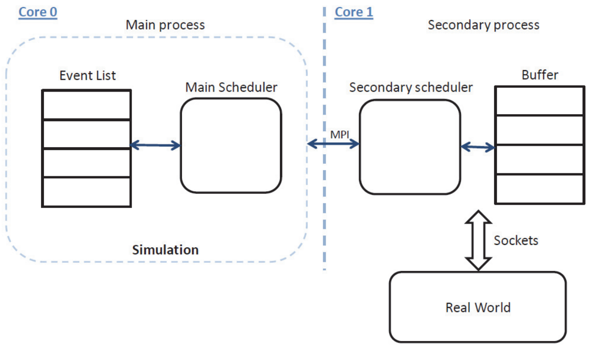

Fig. 3. Simulation core structure.

The communication between cores has been made with the Message Passing Interface (MPI) that allows processes with different cores to communicate with one another by sending and receiving messages.

The main scheduler behavior is similar to that of the cRealTimeScheduler, but it also monitors the messages that have to be exchanged with a real network. Thus, when an event is inserted in the event list and its destination is a real network it is extracted from the list and passed to the secondary process by an MPI send call. In addition, if a new message arrives from the secondary process (by an MPI receive call), it is inserted in the event list and executed at specific simulation time.

The secondary scheduler monitors the source (a raw sockets or an MPI) of the received messages, and continues to process it. When a message arrives from real network, it is sent to the main process. If a message arrives from the main process to be sent to a real network, the scheduler looks if the execution time is the current time and proceeds to send it; if not, the message is stored in a buffer, and it is sent at the accurate time without delay. Therefore the problems explained above are solved.

The main restriction in order to ensure the synchronization between real world and simulation is that the creation time of an event (time at which the event is inserted in the event list) have to be less or equal than the real time to be scheduled.

In conclusion, in the existing OMNeT++ infrastructure, the communication with external devices (through raw sockets) is realized only in wait calls, so synchronization problems could easily appear, and hardware-in-the-loop simulations could be invalidated. The new scheduler described here, solves these problems by using two processes working in parallel. In section 4.2.1, a case study about a hardware-in-the-loop simulation, using the new simulation core, is presented.

4. EXAMPLES

This platform, since it uses the real IEC 61850 stack and it permits the communication with external devices in a synchronized way, could be very useful in different areas, such as training and teaching, developing and testing new algorithms, prototype designs, analysis of Substation Automation System designs, optimal device configuration, or fast interoperability studies.

Two examples are detailed in this section. The first is a network performance simulation in which the communication between the control center and substations has been analyzed. The second is about how to use the simulation platform as a test-bed with the aim of conducting hardware-in-the-loop simulations and developing real network algorithms.

4.1 Network performance: communication between control center and substations

With this example, it is expected that, despite the fact that a real IEC 61850 stack has been integrated in the simulator, the scalability of the simulation will be still acceptable.

The IEC Technical Committee 57, responsible for IEC 61850 standard, has planned to extend the IEC 61850 application areas to new ones, such as feeder automation domain, or control center communication (upcoming IEC 61850-90-2). Regarding the last one, some aspects should be taken into account:

- Harmonization of IEC 61850 modeling with the IEC Common Information Model (CIM, IEC 61968/61970). Some contributions focused on such harmonization are detailed in Santodomingo et al. [10] and Kostic et al. [11].

- Communication between substations and control centers (communication requirements, communication architecture, and other issues).

Regarding the communications, it is necessary to take into account issues such as: the data concentration, the security and the information to be exchanged. Thus, it is very useful to develop a simulation model that evaluates these communication issues.

The possible communication architectures for communications between substations and control center are:

- Communication based on IEC 60870-5-101/104 protocols.
- Communication using a Proxy server.
- Communication without using a Proxy server.

The first solution is used in IEC 61850-80-1, and will not be evaluated here. Communication between control center and substations without using a Proxy server could be applicable for small utilities, but for large networks, with multiple nodes, it is unrealistic.

The substation under study uses a Proxy server for communications with the Control Center. This substation is a 220kV substation of type D2-1 (see grey lines in Fig. 4), with two transformer bays (T1, T2), six feeder bays (F1-F6), and one section bay (S).

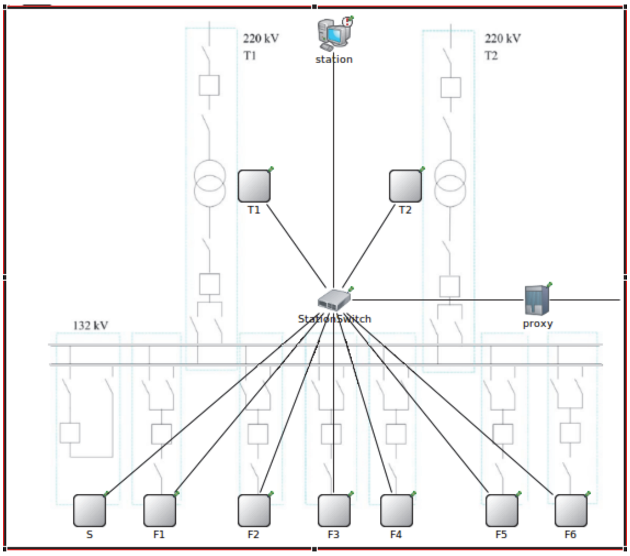

Fig. 4. Substation to simulate.

The Proxy server has two network interfaces, one for the substation, and the other for the WAN. The main function of the Proxy server is to mirror the Logical Devices located in the IEC 61850 devices of the substation (see Fig. 5). Since the simulation platform uses real format messages, it is possible to design the Proxy in a realistic way.

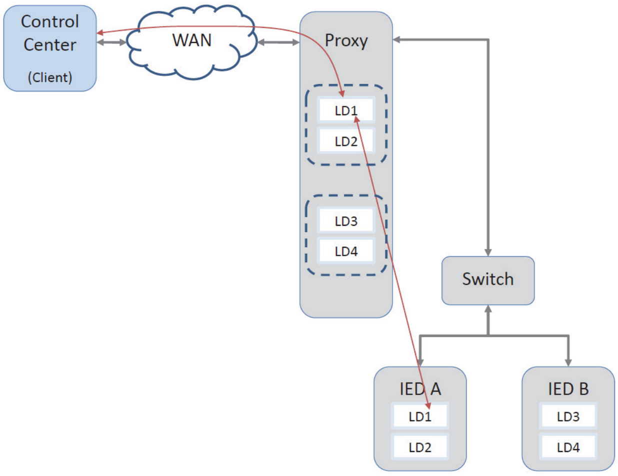

Fig. 5. Communications with Proxy server.

4.1.1 Simulation and results

The simulation has been carried-out for a wide area network with six substations and one control center, such as shown in Fig. 6.

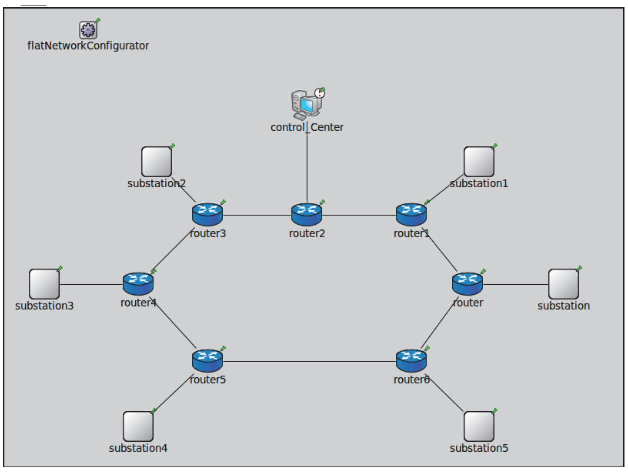

Fig. 6. Wide Area Network Topology.

The Control Center sends messages to substations each 50 ms. When a message arrives to a Proxy from the Control Center, the Proxy sends a request message to the corresponding device, receives the response, updates their status and proceeds to send the response to the Control Center.

The behavior of each substation is similar to example provided by Sidhu and Yin [4]. Thirty seconds of operation has been simulated, and the number of nodes in the simulation is about 300.

The simulation results in terms of performance are shown in Table II. This simulation has been carried-out in an Intel Core Quad CPU (Q9650) of 3.00GHz, with 4 GB of RAM. These results are considered acceptable regarding the scalability.

Table II. Performance results for 30 seconds of real operation

Sampling Rate [Samples/s]	**960**	**1920**	**4800**
Execution time	4’ 52’’	8’ 30’’	19’ 18’’

4.2 Using the simulator platform as a test-bed

The simulation platform can be used as a test-bed from two different points of view. It can be used to conduct hardware-in-the-loop simulation in which it carries-out different tests on real devices; or it can be used to develop network algorithms in the simulator, which afterwards can be implemented into real devices. This section explains these two points of view in detail.

4.2.1 Hardware-in-the-loop

In hardware-in-the-loop simulation, some components are real hardware, and other ones are simulated, normally because they are not available. This increases the realism of the simulation and permits evaluation of real developments in a more detailed way.

A simplified hardware-in-the-loop simulation is analyzed here in order to demonstrate the power of the platform, and to give an idea about how this technique can be used.

A fault case will be presented from the GOOSEs exchange point of view. This case does not include all the IEC 61850 functionality. Thus, functions like automatic reclosing, breakers failure or alarms, for example, are left out of the case.

The substation under study (Fig. 7) will be composed by two transformers working in a parallel manner (T1 & T2) feeding three loads (F1, F2 & F3); two with high priority and one with low priority, which means that in an anomalous condition of operation (one transformer inoperative for example), the low priority load could be disconnected.

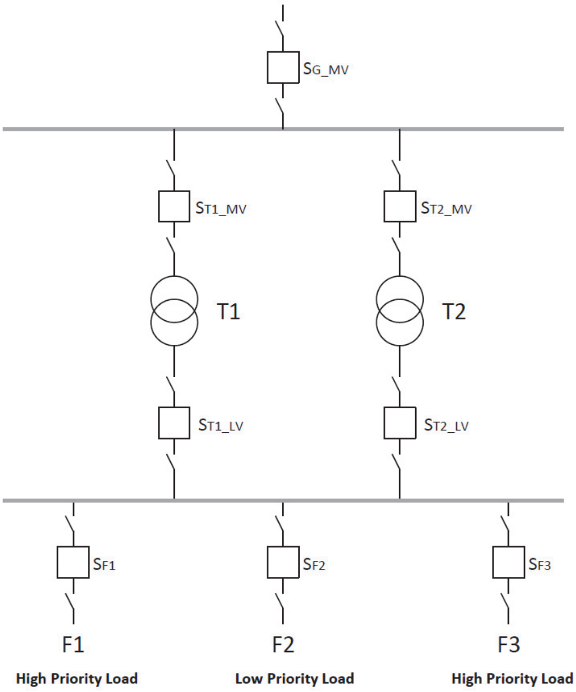

Fig. 7. Substation to simulate.

Due to flexibility and adaptability that IEC 61850 provides, there are several solutions for the same problems from the LN configuration point of view. Moreover, choosing one option or another depends on the functionality expected, the physical constraints and other variables with respect to design issues. In this study, only some LNs involved in this example are exposed in order to better understand how the platform can be used.

Fig. 8 and Fig. 9 show the LN configuration of devices in the feeder and the transformer bay.

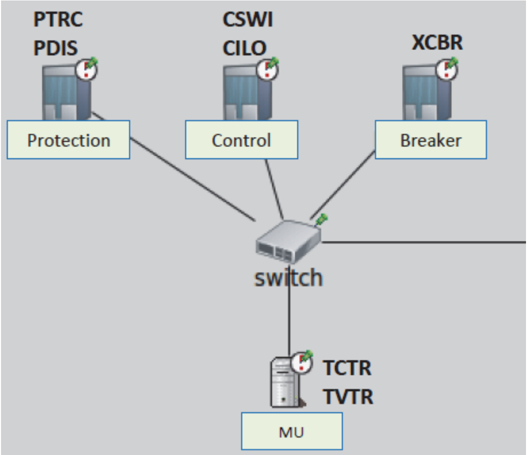

Fig. 8. LN configuration for a feeder bay (F1, F2 & F3).

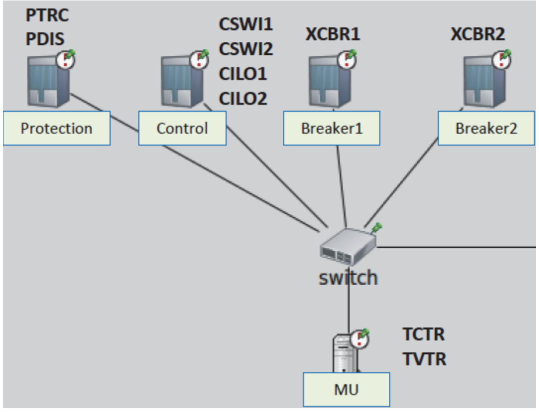

Fig. 9. LN configuration for a transformer bay (T1 & T2 bays).

The LNs employed in the tests are: TVTR (Current transformer); TCTR (Voltage transformer); XCBR (Circuit breaker); PTRC (Protection trip conditioning); PDIS (Distance protection); CILO (Interlocking); CSWI (Switch controller).

With the above configuration, one fault case will be analyzed attending to the switches that are open and the messages that have been exchanged.

Switch SF2 (Breaker in Fig. 8) is a real device connected to the simulator platform through an Ethernet connection. From the point of view of this device, it sees the substation in a transparent way, so it does not know if the substation is simulated or real. Therefore, the behavior of this device can be tested in any situation, and the reaction of other devices in response to its actions can be evaluated as well.

The fault is located between T1 and ST1_MV, and causes that switches ST1_MV and ST1_LV open in order to isolate the fault. Only transformer T2 will be operative to ensure the continuity of supply. In these conditions, SF2 will be open to disconnect the low priority load.

Initially, in T1 bay, the Merging Unit (MU) is sending Sampled Values (SVs) (at sampling rate of 960 samples/s) to Control & Protection devices with the measurements of the current and voltage transformers. When a fault occurs, PDIS detects it and PTRC generates a trip message (GOOSE message) that is sent to XCBR1 in Breaker1 and XCBR2 in Breaker2. The breakers open and their status information changes from ON to OFF. Two new GOOSEs per breaker are sent to Control and Protection devices with the aim of notifying them the new states of the breakers. Next, Control sends one GOOSE to Control device in F2 bay in order to indicate that the breaker should be open (to disconnect the low priority load). In F2 bay, CSWI, in control device, generates a GOOSE message, and sends it to XCBR in Breaker device (the real device). The breaker status information changes from ON to OFF. This new state is sent (by a GOOSE message) to Control and Protection devices with the aim of notifying them of this change.

The simulation platform is composed by one PC (Intel Core 2 Quad Processor Q8300) with four cores of 2.50 GHz connected with a real device through an Ethernet switch. This external device simulates the functions of a breaker in a substation, and it is based on IEC 61850 standard.

From the point of view of the results, the messages exchanged with the real device have been carried-out synchronously without any delay and at correct times.

The maximum delay for messages exchanged has been analyzed as well, and determined to be 0.03ms (GOOSE of 32 bytes) in the worst case. The IEC 61850 requirements are fulfilled.

4.2.2 From simulation to real device

The simulation platform uses the real format of IEC 61850 messages, so applications developed inside this simulation platform can be easily integrated into a real device.

In order to design real applications, it is important to understand how OMNeT++ uses Finite State Machines (FSM), a useful tool for the design of algorithms. It will be very useful in order to follow a methodology.

FSM in OMNeT++ uses two types of states: steady states and transient states. Each time that a routine is executed, the FSM state leaves the actual steady state and arrives to another steady state. During the execution of the routine, it is possible to go from one transient state to another, but the final state will always be a steady state. It is possible to assign a code to handle the entering and leaving of a state.

The application has been developed in C/C++ programming language, under the Linux operating system.

The design of a generic control device will be analyzed from the point of view of its messages processing algorithms in order to give us an idea about how to implement real algorithms and develop real code.

In summary, the device has to manage three stages. First is the initialization and connection; second the main process; and finally the disconnection. There are three types of messages to process: SV, GOOSE and MMS. Taking this into account, a possible FSM algorithm is shown in Fig. 10.

The dotted circles are transient states, and the non-dotted circles are steady states. In each state and transition, real code can be implemented, thus with minimal changes this algorithm can be easily integrated in a real device after being tested in simulation. It is important to note that the real device should use the same MMS library as the one implemented here.

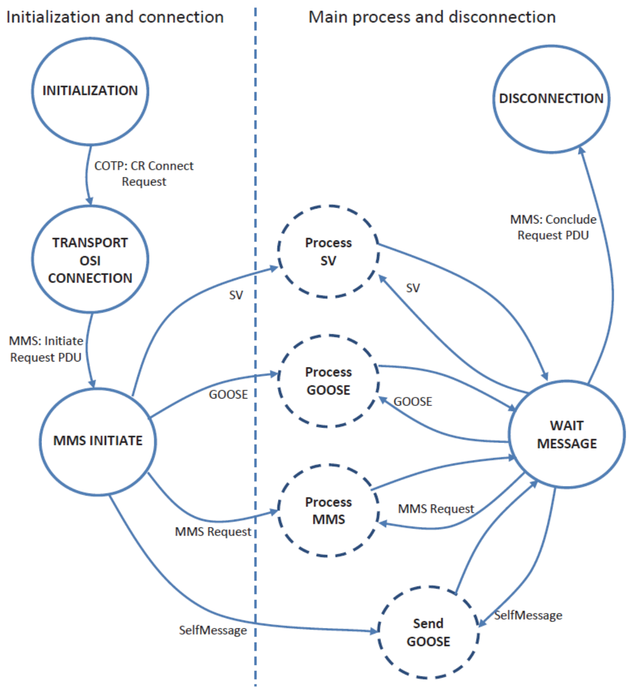

Fig. 10. Message processing algorithm.

Since a real IEC 61850 stack is used, developing applications in this way will permit, first of all, the evaluation of the application behavior in the simulator, secondly the implementation of this in a real device, and finally the testing of such device with a hardware-in-the-loop simulation.

5. CONCLUSIONS

Most of the previous works about IEC 61850 simulations are focused on the evaluation of the performance of an IEC 61850-based communication network, and furthermore on proving that the IEC 61850 requirements are fulfilled. However, the simulation results are very sensitive to the estimated number of exchanged messages among other things; needless to say it could be difficult to evaluate some specific situations in a realistic manner.

In this paper, an advanced simulation platform for IEC 61850-based substation system has been presented. With this simulation platform, it will be possible to carry-out more advanced simulations (such as hardware-in-the-loop simulations) or simple and fast simulations (such as network performance analysis), depending on what the user wants and what the situation requires. This provides flexibility.

On a final note, the two fundamental points of this platform are the newly developed simulation core, which uses two processes working in a parallel manner, and lastly the implementation of the real IEC 61850 stack.

6. REFERENCES

[1] IEC 61850, "Communication networks and systems in substations", 2003.

[2] IEC 61850-5, "Communication networks and systems in substations - Part 5: Communication requirements for functions and device models", 2002.

[3] T. Skeie, S. Johannessen, and C. Brunner, "Ethernet in substation automation", *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 43–51, 2002.

[4] T. S. Sidhu and Yujie Yin, "Modelling and Simulation for Performance Evaluation of IEC61850-Based Substation Communication Systems", *IEEE Transactions on Power Delivery*, vol. 22, no. 3, pp. 1482–1489, 2007.

[5] M. S. Thomas and I. Ali, "Reliable, Fast, and Deterministic Substation Communication Network Architecture and its Performance Simulation", *IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2364–2370, 2010.

[6] M. G. Kanabar and T. S. Sidhu, "Performance of IEC 61850-9-2 Process Bus and Corrective Measure for Digital Relaying", *IEEE Transactions on Power Delivery*, vol. PP, no. 99, p. 1, 2010.

[7] C. R. Ozansoy, A. Zayegh, and A. Kalam, "The Real-Time Publisher/Subscriber Communication Model for Distributed Substation Systems", *IEEE Transactions on Power Delivery*, vol. 22, no. 3, pp. 1411–1423, 2007.

[8] Zhi Zhang, Zhonghai Lu, Qiang Chen, Xiaolang Yan, and Li-Rong Zheng, "COSMO: CO-Simulation with MATLAB and OMNeT++ for Indoor Wireless Networks", in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, 2010, pp. 1–6.

[9] J. Juárez, "Webpage of Javier Juárez Montojo". [Online]. Available: http://www.iit.upcomillas.es/jjuarez/.

[10] T. Kostic, O. Preiss, and C. Frei, "Towards the formal integration of two upcoming standards: IEC 61970 and IEC 61850", in *2003 Large Engineering Systems Conference on Power Engineering*, 2003, pp. 24–29.

[11] R. Santodomingo, J. A. Rodríguez-Mondéjar, and M. A. Sanz-Bobi, "Ontology Matching Approach to the Harmonization of CIM and IEC 61850 Standards", in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010, pp. 55–60.