DEVELOPMENT OF A SYNCHROPHASOR DATA CONCENTRATOR BASED ON THE IEEE C37.118.2 STANDARD

JUAN CARLOS VARGAS-RUBIO¹, OSCAR DAVID FLÓREZ-CEDIEL² & JULIÁN R. CAMARGO L²

¹IMT Atlantique, Rennes, Francia

²Engineering Faculty, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

ABSTRACT

This document presents the development and validation steps for the design and implementation of a synchrophasor data concentrator based (PDC) on the IEEE C37.118.2 standard. The PDC can send, receive and interpret data from one or more synchrophasorial measurement units. Additionally, it has a graphical user interface that facilitates its configuration and visualization of information from different sources dynamically. It was possible to develop a synchrophasor data concentrator to send, receive, and interpret data from more than one synchrophasorial measurement unit. Additionally, thanks to the graphic user interface designed, the PDC becomes versatile and easy to use the tool. This software allows the user to visualize, add or remove dynamic trend graphs in real-time, see the vectorial behavior of the phasors, review the variations in the frequency of the electrical network, all this for several servers concurrently.

KEYWORDS: IEEE C37.118.2 standard, PDC, PMU, Real-time, Synchrophasor

Received: Oct 06, 2020; **Accepted:** Oct 26, 2020; **Published:** Nov 23, 2020; **Paper Id.:** IJMPERDOCT202067

1. INTRODUCTION

Synchronous networks are systems-oriented to data acquisition, supervision, monitoring and control in real-time of the electrical network. They are used in wide-area electrical networks because each phasor measurement taken from the network is synchronized, through GPS technology, by a given timestamp according to Coordinated Universal Time (CUT). This allows seeing the state of the network in geographically distant places in the same instant of time to identify the exact place and moment in which an event occurred and take control actions to bring the system to normal operating conditions. Some of the applications of synchrophasor technology include wide-area control, validation of the system model, determination of stability margins, maximization of the stable load of the system, recording of disturbances, visualization of the dynamic response of the system, state estimation, among others.

Two main components can be identified within a synchrophasor network, the phasor measurement units (PMU) [1] and the phasor data concentrators (PDC) [2]. PMU are devices that measure and deliver voltage and current, system frequency, rate of frequency change, a timestamp and the metadata needed for the interpretation of the information. The PDCs act as nodes in the communication network where data from synchrophasors from different PMUs or other PDCs is processed and sent as a data stream to a higher level PDC. Some functions of the PDCs can be to perform data quality checks, data storage for later analysis and data visualization, among others.

The storage of synchro-phasor information facilitates a detailed post-failure analysis that helps to determine the sequence of events that caused a network failure, allowing the identification of the causes of the disturbance and

the taking of measures to prevent future reproductions of these events.

2. IEEE C37.118.2 STANDARD

The IEEE standard for the transfer of synchrophasorial data on a power network defines four types of messages as follows [3]. The data message are the measures taken by PMU such as synchrophasors, frequency, rate of change of frequency (ROCOF), analog and digital worlds. The Configuration Message is the device configuration data, amount of PMU included in the data frame, data format, amount of signals and their names, scale factors and other metadata needed to interpret the information. The Header Message is the descriptive information sent by the devices but provided by the user. It is sent as ASCII characters, and the Command Message, which is sent by the devices to control the communication. With this type of message, data transmission is started or ended and configuration and header frames are requested.

3. DEVELOPMENT OF THE CLIENT PROTOTYPE OF THE SYNCHRO-PHASOR TRANSMISSION PROTOCOL

The initial prototype of the PDC was developed entirely in C++ using the Visual Studio Community 2017 development environment. This prototype can exchange information with another process, either in the same device or in a different one, through a TCP/IP socket whose parameters are configurable by the user. The interaction with the user is done through the console application. To simulate the operation of this prototype was used the Hercules software, a client-server communication simulator for the TCP and UDP protocols, and the Npcap library to view the packets transmitted between Hercules and the C37.118.2 Client in Wireshark [4].

Figure 1 shows the Hercules configured as a TCP server since it will act as a PMU, and port 4712 is chosen since it is the default port for synchrophasors according to the standard.

Figure 1: Hercules server

When you click on Listen, Hercules starts receiving customer requests. The advantage of this simulator is that it allows you to send frames in Hexadecimal format by enabling the HEX option shown in the figure. The connection parameters are the IP address, localhost as the Hercules is running on the same device, and the port. Additionally, this PDC

allows the user to determine the number of times to request the retransmission of failed header frames or configuration frames, if necessary, before restarting the protocol. A failed frame is defined as the one for which the client detects incorrect synchronization bytes, inconsistency between the frame length and the FRAMESIZE field or error in the cyclic redundancy check CHK.

When the configuration is finished, the PDC starts communicating with the client, initially sending a command to turn off the data transmission, in case the PMU is trying to send some frame at the moment of connection, and then a command asking for the configuration frame. Figure 2 shows the basic sequence diagram for communication between a PMU and a PDC.

Figure 2: Protocol sequence diagram

The client starts the communication by requesting the configuration frame. The PDC needs to have the PMU configuration frame to be able to interpret the header frame and data frames. The client then requests the header frame, if the user so determines, and then starts the data transmission. The communication ends when the PDC sends the command to turn off the data transmission.

With Wireshark you verify that the protocol works correctly as specified by the standard, the source and the destination are localhosts since the two processes are running on the same computer. Frames from the server were sent manually while frames from the PDC were sent automatically. Figure 3 shows the PDC's interpretation of the header and data frame information, which means that the configuration frame was also correctly interpreted.

```

C:\Users\RD06\Documents\Visual Studio 2017\Projects\PdcPruebas\Debug\PDC.exe
Header Frame Sent by PMU
AA 11 00 18 1E 36 44 85 27 F0 56 07 10 98 50 4D 55 5F 41 58 4F 4E 5B EA

INFO: PMU_AXON

Turn ON Data Transmission
AA 41 00 12 1E 37 59 2D D5 78 00 09 08 80 00 02 F9 45

PMU: Station A
Data Frame sent by PMU
AA 01 00 34 1E 36 44 85 36 00 00 00 41 B1 00 00 39 2B 00 00 E3 6A CE 7C E3 6A 31 83 04 44 00 00 09 C4 00 00 42 C8 00 00 44 7A 00 00 46 1C 40 00 3C 12 D4 3F

Controller has received a new Data from client

MY PMU-----
IDCODE: 7734
SOC: 1149580800
FRACSEC: 0.016817
VA      :133987|_ 0
VB      :134003|_ -120.001
VC      :133995|_ 120.003
I1      :499.874|_ 0
Frequency: 62.5 Hz
ROCOF: 1 Hz
ANALOG1 :100
ANALOG2 :1000
ANALOG3 :10000
Turn Off Data Transmission
AA 41 00 12 1E 37 59 2D D5 85 00 08 98 C0 00 01 93 05

```

Figure 3: Header and data frame interpreted by the Client

4. DEVELOPMENT OF THE GRAPHIC USER INTERFACE FOR THE PDC

The development of the graphical interface was done in C#, creating a proxy class whose function is to pass to the application the data it needs to process. These data are those required to interact with the user as the values of the phasors and their names, frequency, time stamps, among others. For C# to use this data, a conversion must be made in its type since some types of data in C++ do not exist in the same way in C#. For example, C++ vectors were passed as collections to C#. For this purpose, a class was created whose attributes are the data that is expected to be received from C++ but, with data types editable by C#.

The application developed in C# is of the WPF type, which was chosen instead of Windows Forms since it allows better isolation between the design of the graphic interface and the logic behind it, besides offering a better performance since it uses the resources of the graphic card of the computer where it is running. Additionally, WPF has libraries that help create dynamic graphics such as Dynamic Data Display [10], making it easier to plot real-time synchro-phasor information. The language used by WPF to create the graphic interface in XAML.

To take advantage of all the features that WPF offers, the program was implemented based on the MVVM architecture pattern [5]. This model allows for clean isolation between the graphical interface and the code behind it. Under this architecture, each XAML document has a view model (ViewModel), a class that is responsible for updating the interface through a link (Binding) between the values of the items (Label, TextBox, Graphics, etc.) and the variables that they must display [6]. That is when you change the value of a variable during the execution of the application this change is reflected automatically in the interface.

The interface designed for the general protocol execution window is shown in Figure 4. Here you can see the buttons that allow you to start or stop the communication protocol with PMU [7]. Also, general information is presented, such as the name of the PMU, the header information, the ID code, the analog values, and at the bottom, the sent and received synchrophasor messages in hexadecimal format [8].

Figure 4: User graphic interface

The content of the Graphs tab is shown in Figure 5 and the graphs of the phasors chosen by the user and the frequency are displayed.

Figure 5: Interface for dynamic graphics

The image in Figure 6 shows the graphs of all the phasors, four of voltage and one of the current, delivered by the equipment. Since the Dynamic Data Display library self-adjusts the scale so that all the signals can be seen, the graphs of the phasors whose magnitude is almost zero overlaps [9]. In the phase graph, you can see the phase trends of all the phasors. In the graph at the bottom of the image, you can see how the frequency varies between 50.94 Hz and 60.08 Hz. In the vector chart, you can only see the vectors corresponding to V1 (Positive sequence voltage) and VB (B-phase voltage) since their magnitudes are very large compared to the magnitudes of the other vectors. It is seen that the length of V1 is about one-third of the length of VB, this is because the magnitude of V1 is about 7400 V while VB about 22000 V.

Figure 6: Dynamic graphs of all phasors for the SEL 451 Relay

Wireshark will be used to check that the PDC is indeed performing a correct interpretation of the frames delivered by the equipment. The synchrophasor filter of this sniffer captures the frames corresponding to the synchrophasor protocol and decodes them, so you can check that Wireshark and the PDC are interpreting the information in the same way. Figure 7 shows the frames captured from the communication between the PDC and Hercules since these are the example frames of the standard.

No.	Time	Source	Destination	Protocol	Length	Info
82	131.790721	127.0.0.1	127.0.0.1	SYNCHROPHASOR	120	Command Frame, data transmission off
84	131.792645	127.0.0.1	127.0.0.1	SYNCHROPHASOR	120	Command Frame, send CFG-2 frame
106	148.310425	127.0.0.1	127.0.0.1	SYNCHROPHASOR	992	Configuration Frame 2
111	148.323394	127.0.0.1	127.0.0.1	SYNCHROPHASOR	120	Command Frame, data transmission on
401	311.008084	127.0.0.1	127.0.0.1	SYNCHROPHASOR	188	Data Frame
476	338.887259	127.0.0.1	127.0.0.1	SYNCHROPHASOR	188	Data Frame

Figure 7: Frames captured by Wireshark from the communication between the PDC and the Simulator

The results of these tests demonstrate that the synchrophasor data concentrator developed in this project can send, receive, and interpret data from more than one synchrophasor measurement unit. Additionally, thanks to the designed graphical user interface, the PDC becomes a versatile and easy-to-use tool. This software allows the user to visualize, add or remove dynamic trend graphs in real-time, see the vector behavior of the phasors, check the frequency variations of the power grid, all this for several servers concurrently.

5. CONCLUSIONS

A synchrophasor data concentrator allows monitoring, storing and transmitting the information from multiple PMUs in real-time. Due to the multiple applications of synchrophasor technology in the electrical sector, the implementation of synchrophasor networks in Colombia is increasing, making it necessary to develop PDCs capable of handling large amounts of information. At the end of the project, it is expected to have a user-friendly and high-quality application that will allow

taking full advantage of the benefits offered by synchro-phasor technology for the control and supervision of electrical networks.

Future work should develop the PDC as a server. The software must be able to concentrate synchrophasor data from different sources and generate the configuration, header and data messages containing all this information to send them in a single data stream to higher-level PDCs. A database can also be implemented to store synchrophasor data, to use this information for post-failure analysis of electrical systems and to perform software stability tests.

ACKNOWLEDGMENTS

The authors would like to thank the Universidad Distrital Francisco José de Caldas, LASER research group, and Eng. Ivan Dario Claros Gomez, Manager of Technological Development of Axon Group Ltda., has supported the development and testing of the project.

REFERENCES

1. *IEEE Guide for Synchronization, Calibration, Testing, and Installation of Phasor Measurement Units (PMUs) for Power System Protection and Control.* 2012.
2. *IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring.* 2013
3. Brunello, G. y Martin, K. *An overview of the IEEE Standard C37.118.2-Synchrophasor Data Transfer for Power Systems:* 2014 IEEE PES General Meeting Conference & Exposition.
4. Shaoqiang W., DongSheng X. y ShiLiang Y. *Analysis and application of Wireshark in TCP/IP protocol teaching:* 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies.
5. XiaoLong L., DaLiang C., Hui P., XiaoYu Z., YuanXin L. y YaXian Y. *Application of MVVM design pattern in MES:* 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER).
6. Moreno Chuquen, R., & Obando Cerón, J. S. (2018). *Network topological notions for power systems security assessment.* International Review of Electrical Engineering (IREE), 237.
7. IEEE Standards Association: *IEEE Standard for Synchrophasor Data Transfer for Power Systems - IEEE Std C37.118.2, 2011*
8. Prabhakara S., Sureka V., and Murthy M., (2016). *A Novel Technique for Reducing the Fault Current and Overvoltage in Electrical Power Distributed System and Enhancing the Security Through an Active Type SFCL.* International Journal of Electrical and Electronics Engineering Research (IJEEER), Vol. 6, Issue 2, Apr 2016, 13-26
9. "Dynamic Data Display". CodePlex, 2017.