UNIVERSITY POLITEHNICA OF BUCHAREST

FACULTY OF POWER ENGINEERING

THESIS SUMMARY

Real-Time Data Acquisition, Storage and Transmission from a SMU Prototype towards the Data Correlation Centre

research on improving
the observability in WAMPAC networks

Author: **Ing. Mihail POPA**
Scientific advisor: **Prof. Dr. Ing. Răzvan MĂGUREANU**, MMAE department

Bucharest
2014

Foreword

The thesis entitled „Real-Time Data Acquisition, Storage and Transmission from a SMU Prototype towards the Data Correlation Centre”, written by Ing. Mihail Popa belongs to the modern research area of power engineering, having as well interdisciplinary characteristics by implementing concepts, principles, technologies and solutions from other adjacent fields like telecommunications, data security and computer science.

Dealing with a subject of a great interest nowadays – the observability in WAMPAC and WAMS networks, this thesis belongs to the research area of smart grids as well. The document has the structure of a fully qualified scientific endeavour, following the typical guidelines of exposing a series of issues noticed during the author’s researches; analysing the current limitations of the concepts and methods of data transmissions in today’s PMU’s and proposing new approaches and interdisciplinary concepts, testing them at the same time by means of a pilot environment.

This summary is a synthesis which pinpoints the solution propositions and results of the thesis. The simplified structure of the document is outlined below:

Chapter 1

Necessity and Opportunity of the Research

1.1. Introduction

The techniques of modern computing, telecommunications and data security have evolved substantially in recent years, offering people plenty of new functionalities and significantly simplifying their mobility and communication in the new globalization context.

The monitoring and control processes in power engineering, especially in the research area of smart grids, can benefit from many points of view from innovations which are already well established in other technology areas (such as telecommunications, cryptography/security and computer science), changing as well some classic concepts and approaches about monitoring, observability, estimation, forecast and control.

For example, modern digital communication technologies, which have replaced over time the old phone lines and which have been the base of development in voice over IP solutions, offer today an already well-established infrastructure for the real-time transport of audio signals in international phone conversations, audio and video conferences, including the so-called "tele-presence" (involving a very high number of simultaneous audio and video streams between the participants).

Using the same telecommunication infrastructure, not only for media applications, but also for monitoring and completely observing the smart grid by transmitting the complete[1] set of power signal waveforms in a real-time manner, is one of the solutions proposed in this thesis, perhaps also as a future foundation of a new smart-grid monitoring and control strategy, in the context of permanent changing requirements, such as: the geographic scale at which the monitoring takes place, the ever growing data volume required for a 360 degrees monitoring and the consistently shorter time the data has to be available in. [see chapter 2.6.1 of the thesis]

During the development of the proposed solution, the author analyses and presents, how the modern telecommunication technologies significantly contributes to the performance growth of the WAMPAC (*Wide Area Monitoring, Protection and Control*) and

[1] The completeness of the waveform refers to a very high resolution and sampling rate (≥12 bit and 44KHz), in order for the energy phenomena to be deeper and more accurately analyzed (over time spans as short as 22 µs)

WAMS (*Wide Area Measurement Systems*) networks, through a more precise and detailed real-time monitoring and observation, by embracing interdisciplinary principles and techniques (like for instance: re-using some of the well-known, proven and already perfected protocols, or adapting some of the most stable compression and decompression algorithms to the nature of the power signals which flow through the lines of the power grid). [chapters 1.2, 2.5 and 2.6]

In addition, this thesis discusses also the technical and system requirements which have to be fulfilled in order to sustainably achieve the desired performance of the existing WAMPAC networks, raising at the same time some warnings about the consequences of not fulfilling these requirements. The necessity of this thesis occurs, as result, out of the significant differences between various aspects of the WAMPAC networks and PMU/PDC equipment highlighted below by the author:

- The implementation in power engineering of a quite rudimentary protocol (the C37.118), while the telecom uses for generations well known and already tested protocols for real-time data transport [solution detailed in chapter 2.6 of the thesis];
- Lack of data security transferred between the PMU's and the rest of WAMPAC network they belong to, while the wold of Internet (which the WAMPAC network relies on) is frequently shaken by cyber-attacks, to which the researchers in telecom and computer science areas have successfully responded by using solutions to encrypt the data using asymmetric and pre-validated SSL [chapter 1.2.2 of the thesis];
- C37.118 connectivity limit between the PMU and the PDC to only one permitted initiation way (that is from the PDC to the PMU), which narrows the interoperability between different branches of the WAMPAC network (belonging to different partner companies/operators), while the classic TCP/IP networking already uses proven solutions of virtualising the data networks (using the so-called VPN tunnels) [chapter 1.2.2.2 of the thesis];
- Low time resolution of the power signal waveform offered by the PMU equipment via reconstruction out of the synchrophasors (of approximately 25 points per period, equivalent to around 800 µs), whereas the requirements for studies and researches in power engineering specify the need for more detailed waveforms (ability to view waveform samples as close to each other as 20 ~ 40 µs). Nowadays the sampling of a signal during data acquisition can be easily achieved at a higher rate, such as 44100 Hz using just a usual personal computer equipped with a sound card. [chapter 2.3]

The first chapters present the synthesis of the author's studies regarding the current implementation of the monitoring solutions for power grids using three families of monitoring equipment available nowadays: data-logger, PMU and PDC, taking their advantages and disadvantages into analysis. In addition, in the next chapters the author defines, presents and studies the hybrid concept of the SU equipment, proposed as a solution for combining all advantages and eliminating the limitations of the above three equipment families. This way the current thesis opens new research directions (detailed within the chapter 4.3) regarding:

- Optimization of the compression/decompression algorithms against the nature and characteristics of the power signals, with the purpose of achieving a higher data compression rate, ensuring at the same time the integrity of the information on the receiver side after decompression;
- Auto-management mechanisms of the SMU equipment and its inside monitoring processes and self-containing diagnosis and automation;
- Implementation of the SMU concept on FPGA support with the purpose of achieving an industrial autonomous equipment, capable of self-management and remote administration without the need of local human intervention;
- Extension of the PDC functionalities in order to accommodate the high resolution waveforms provided by the SMU among the Phasor measurements provided by the classic PMU's and correlate their time stamps

1.2. Interdisciplinary concepts applied in power engineering

This chapter briefs the interdisciplinary concepts, which represent the foundation of the cross-technology integration of this power engineering thesis with all the other adjacent research fields, where various principles, methods and concepts have been inspired from, with the purpose of building the solution proposed by the author in this thesis. At the same time, all the cross-technology concepts used in this chapter have been classified based on their origin area, as well as on their degree of relevance to this thesis.

In section 1.2.1 are highlighted the need and the basic principles of data transmission protocols in the network used and implemented in the prototype developed by the author SMU. According to the needs defined by WAMPAC systems for monitoring electrical networks, data communication protocols must not only ensure their transport from the transmitter (measurement equipment) to the receiver (the management and collection centre), but to ensure, on one hand that integral data has arrived and, on the other hand, the fact that the transport time required is minimal.

Therefore, the author highlights in section 1.2.1 the differences of the same methods of transportation data set by different protocols in terms of interconnection and data security capabilities. The latter aspect is then furthermore discussed in section 1.2.2, where the author highlights the modern techniques of ensuring the data security used in designing the SMU prototype and the importance of taking them into considerations when developing a WAMPAC system for the stability of a given smart-grid. Modern solutions used nowadays in adjacent technology areas are also analysed, having a higher weight on the two approaches of achieving the bases connectivity between the key components of the WAMPAC network (that is PMU, respectively PDC): active PMU versus passive PMU (current state of implementations) as it's being illustrated in figura 1.

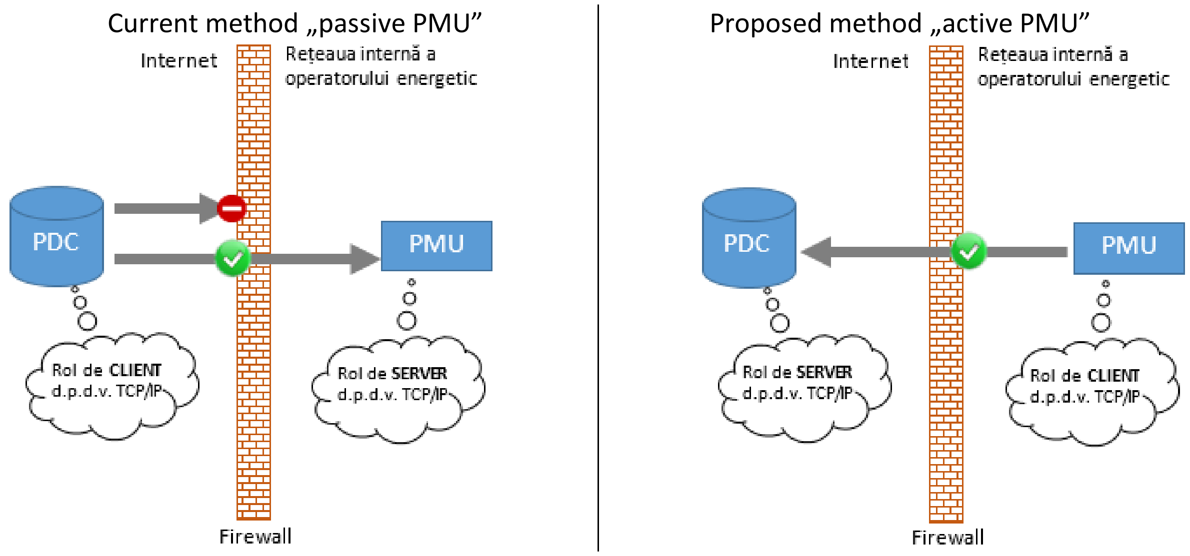

Figura 1. Comparison between the two implementation options for the communication topology between a PMU and a PDC within a WAMPAC system (source: the author)

The two implementation options for the communication topology between o PMU and a PDC are based mainly on the concepts of *client* and *server* used intensively in telecom areas. Together with the tightly correlated concepts of firewall, encryption, data compression they are discussed in detail and analysed in the sub-sections 1.2.2.1, 1.2.2.2 and 1.2.2.3 around the basic principle of data security (as illustrated in figura 2) used thereafter for the definition of the system and technical requirements of the SMU prototype design.

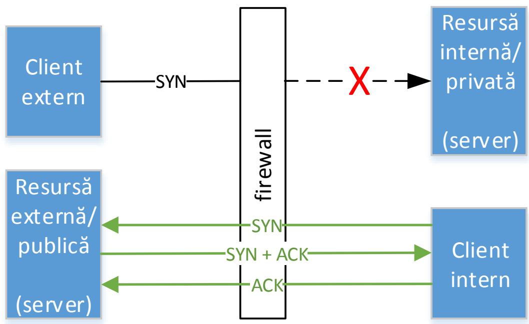

Figura 2. Difference between the connection initiated by from outside (blocked) and the inside one (permitted) (source: the author)

Firewall devices are spread in all network environments due to its effectiveness. They are efficient at blocking the access of unauthorized external clients. Practices like identity theft, interposing between an authorized client and a server protected by the firewall, and interception of their data, cannot be fully eliminated or blocked, unless modern encryption algorithms based on public and private keys are used.[1] In the sections after 1.2.2.3 within the thesis, the author starts a complex study on current security practices in the WAMPAC network, identifying the limitations of the current power systems security concepts and proposing principles from the today's cryptography and testing them afterwards by means of the SMU prototype.

Chapter 2

Real-time Data Acquisition, Storage and Transmission

This chapter contains a tight integration of both study topics on last generation of PMU solutions and implementations in WAMS and WAMPAC networks, done by the author during intensive researches and granular tests, as well as design, prototyping and development (hardware and software) topics, done entirely by the author and described thereafter in detail.

2.1. Grid's Macro-Observability throughout the Current WAMPAC systems

In section 2.1. of the thesis, the author performs a detailed study on the observability issues of the WAMPAC networks, paying special attention to implementations in communication and security point of views specified by the most modern standards (like: C37.118.2-2011 [15], IEC 61850, IEC 60255-118-1 și IEC 62351-1) and implemented in last generation WAMPAC networks in Japan and United States. The authors of [16] define the WAMPAC as being a set of equipment, integrated into a distributes system (on a very large geographic scale), capable of delivering measurements "off the field" in real-time and correlate data from many other such nodes, with the purpose of creating an overall image of the entire network and finally take automatic control, in case that the stability of the entire monitored power system starts to face any issue. The same study defines the four states of a

[1] Key pair used with different purposes – oe key for encryption and the other one for decryption

power system, pron the perspective os the WAMPAC monitoring and creates a diagram of those states and their transitions among each other (figura 3)

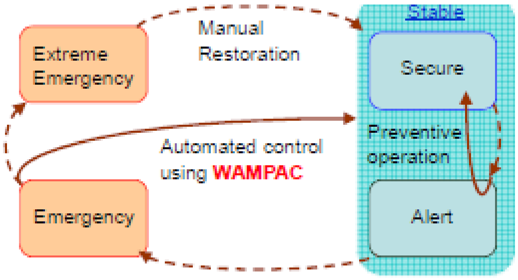

Figura 3. The state chart of a WAMPAC system (source: [16])

According to the graphic representation in figura 3, the new generation of WAMPAC systems overtakes a new role compared to the previous generation of WAMS: (i) in case that small deviations from the steady state are noticed, the WAMPAC system generates some warnings or informative messages, while also taking some automatic preventive operations and (ii) in case that a moderate instability phenomena emerges, the WAMPAC system initiates automatic control measures (either proactive, or reactive). The border between a moderate instability and a high or very high instability is very thin and difficult to determine precisely; it is most of the times dictated by economic factors.

In contrast to the previous generations of WAMS infrastructures, where the system was not assigned any automatic control [29], but just monitoring and alerting, the today's WAMPAC system can intervene automatically in restoring the balance, only in some clearly defined conditions presented by the author of the current thesis in chapter 2.1., when the data consolidated by the super-PDC's confirm what degree of complexity the disturbance has.

Such a WAMPAC structure which integrates the concept of super-PDChas been deployed in the United States as the EIPP (*Eastern Interconnection Phasor Projects*) [14, 34, 17]. It integrates independent operators and regional organizations for power transportation and distribution to consumers, together with the modules of the WAMPAC system and data visualisation and reporting solutions from field-measurements, like:

- Visualizing the local frequencies on regions and key node level (at interconnections or certain grid segments)
- Visualizing the phase differences one many monitoring areas, as well as pre-alarming in case certain thresholds are reached
- Visualizing the voltage variations

The role of a PMU equipment in the wide context of a WAMPAC network is afterwards studies by the author in sections 2.1.1 and 2.1.2, from the point of view of both functionalities and data flows between internal components, instantaneous frequency and phase estimation algorithms and modalities of transmitting the acquired data.

According to the latest version of the standard C37.118.2 published in 2011 [15], a PMU is defined officially as being a piece of equipment which produces estimations of the instantaneous frequency and the rate of change of frequency (*RoCoF*) based on the power signals of voltage and current on the specific power grid node and a time synchronization signal. It finally computes and delivers the so-called synchrophasors. The standards does not define though whether the PMU must be constructed on an integrated hardware platform (using microcontrollers and/or DSP's) or on a software platform or even as a virtual instrument in a virtualization ecosystem.

In the paper [5] published by the author of the current thesis in 2011 during the IEEE SMFG international conference, all the requirements imposed to a PMU equipment are outlined based on economic, robustness, availability, security and data consistency criteria. As result of the study presented in this paper, it has been concluded that a classic implementation takin into account only the specifications of C37.118.2 documentation does not guarantee the following:

- Data security
  The Data transmitted from a PMU to a PDC are not secure; as result there is a high risk of loss, theft of measurement data alteration
  As chapter 3.1 in the thesis presents, the interception of the C37.118.2 data transmission between a PMU and a PDC is possible using common unsophisticated devices available to anyone;
- Lack PMU to PDC or PDC to PMU mutual authentication
  Due to this lack of authentication there is no certainty that the participants at a data conversation PMU – PDC are exactly who they pretend to be; there is high risk that a PMU in one node of the grid is mistaken by another PMU in the same or a different grid segment, leading to a useless measurement

Despite these limitations in functionality and implementation, the PMU equipment studied by the author have quite high purchase and integration costs, as depicted by the author in chapter 3.3 of the thesis. This is also the reason why studies like [10, 11] try to find the best placement of the PMU's in strategic nodes of the power grid, in order to minimize the necessary number of PMU's and maximize the observability of the entire grid, offering as well fault tolerance, if one or more PMU's fail.

2.2. The Concept and Characteristics of the SMU Prototype

In many fields, where the observability and understanding a wide scale phenomenon is necessary, there are specific requirements to correlate the measurement obtained from a multitude of manifestation points of the phenomenon (these are usually geographic distributed at great distances one from the others), in order for the entire global image of the phenomenon to be studies/observed/understood correctly based on all the measurements from all the nodes captured at a given point in time (such an "image freeze" in the time lapse of the phenomenon) [26]. Such phenomena include the electric flows in a geographically distributed power system, given their rapid development and spreading in the recent decades. [3].

Due to their geographically distributed nature, the measurements node cannot be always placed at equal distances from the central point of collecting the data and the data transmission lines, connecting all these nodes to the central point of data collection, cannot have the same signal propagation characteristics [11], thus creating delays impossible to estimate or compensate in a software manner.

As such, the measured data in different geographically distributed points arrive at the central correlation node at different points in time (some have greater delays than others), although the actual measurements took place at the same time. This fact prevents the correlation logic from using the point in time when the measurements arrived at the correlation node and thus scientists have identified the necessity, that the data generated on-field is already time stamped with a common time reference, even before leaving the remote measurement devices on the field.

As result, the correlation logic can then use the time stamps when the data has been measured (as stamped by each individual device), instead of using the time stamp when the data has been received by the correlation node.

In order for the time references of all on-field measurement devices to coincide, it is required that each equipment has its internal clock synchronized with all the other pieces of equipment from other remote locations, thus building the concept of the *synchronized measurement unit* (or abbreviated: *SMU*)[1] for simplicity and analogy with the well-known PMU equipment. In chapter 2.2 of the thesis the author introduces the concept of SMU as a hybrid concept between the working and building principles of PMU equipment and data-logger equipment. One can say about the PMU devices, that their entire processing power is

[1] In literature, the term SMU has not been yet used as is, in order to indicate the equipment which performs this type of waveform recording, but only as a general concept about the act of "synchronized measurement" or "synchronized-measurement technology". The author has used this term for the first time based on the analogy with the term PMU (Phasor Measurement Unit), however adapted to a totally different ways of measureing which does not necessarily involve a "steady-state" of the power grid.

invested in running the computational algorithms for determining the instantaneous phase and frequency.

In contrast to a PMU device, the entire processing power of a SMU device is focused not towards the local computation of the angle and frequency, but towards the integrated execution of the following three processes (depicted in figura 4), according to the system requirements studied by the author in section 2.6.1 of the thesis:

- Recording the waveforms of the voltages and currents on all three phases[1] at the highest possible resolution
- Efficient data compression of the recorded data, without any loss of information
- Real-time transmission of the compressed data as streams and also the management of the streamed/stored data

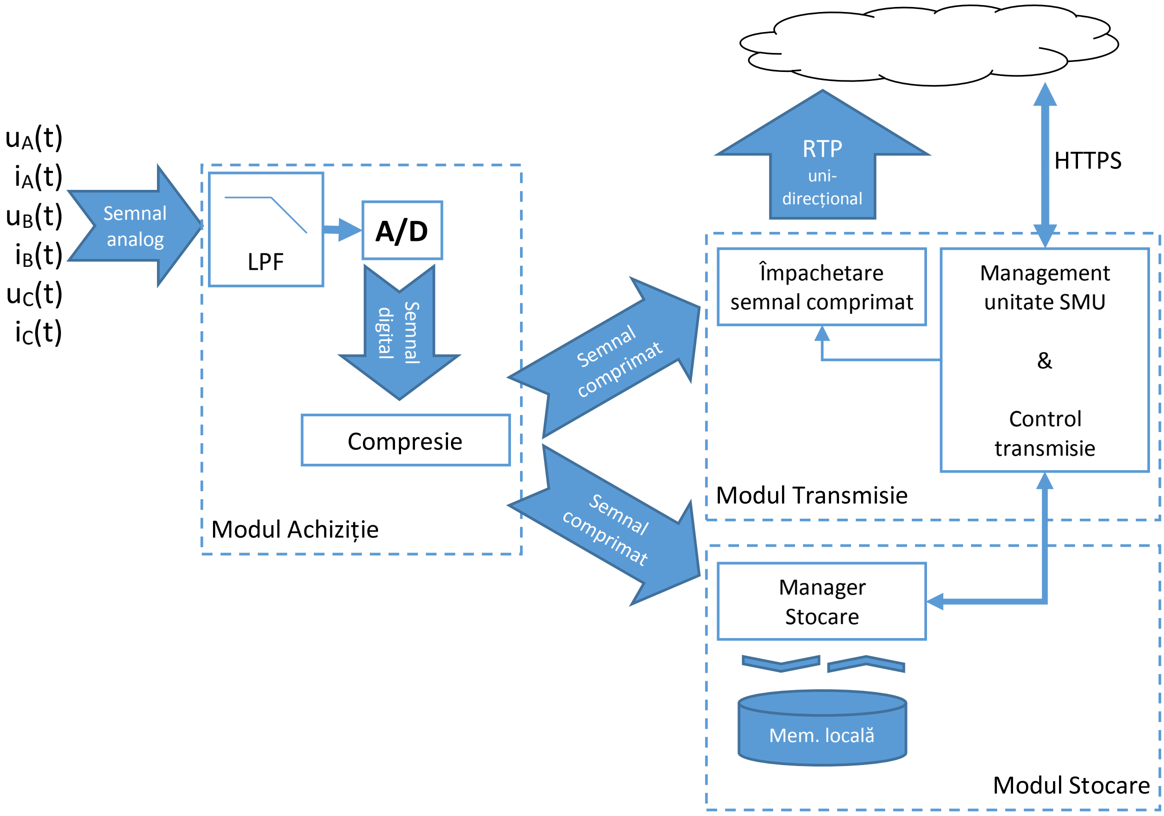

Figura 4. Processes run by the SMU equipment
(source: the author)

Another interesting category of measurement devices used intensively in power engineering are the *data-loggers*. The term resembles its permanent recording capacity

[1] The idea of this type of recording has been already taken into consideration and then implemented by the manufacturers of the so-called „data-logger" as described by [36], however a SMU differs significantly as described in section 2.2 of the thesis

similar to and airplane's "black box", but for all the waveforms within the grid node where the equipment is deployed.

The base similarity between the SMU device and a data-logger is that both devices focus on the high fidelity recording of the power signal waveforms, not on the local computation of the power parameters like frequency, RoCoF, angle, magnitude and Phasors.

Opposite to a data-logger, a SMU device performs the high definition recording of the power signal waveforms on more channels than a data-logger (for comparison the UfE equipment used, studied and tested by the author [36] only records on 4 channels). In addition, the SMU device ensures that the waveforms are time stamped using the same GPS technology as the PMU does. At the same time the SMU device ensures the real-time streaming of the data, not only the local storage as the data-logger does.

The fundamental difference between the measurement principle of a PMU and the one employed by the SMU consists in the "outsourcing" done by the SMU of the computation of the instantaneous phase and frequency, its processing power being aimed towards the real-time high definition recording, compression and streaming of the waveforms. The computations are done in turn by the centralized data collection node.

As a consequence of shifting the place where the computation is performed, a new interesting discussion arises about the advantages and disadvantages of centralized versus decentralised computation of power grid parameters. They have been analysed by the author in table 2 at page 36 in the thesis. This shifting involves changing the way data is transmitted, in order to ensure that the waveforms do not suffer any alteration during the transport and also that none of the waveform segments are lost. In addition the way the waveforms are received by the correlation centre also changes, including the way the decompressed data is used further for the centralized computation.

Because the computation of these parameters is not a local one, but rather a centralized one after having performed the shifting as mentioned above, the whole centralized processing power required to do the computation is much higher, however due to the new computing technologies (similar to *big-data* used in *cloud computing*), this is no longer impossible to achieve. Figura 5 illustrates the concept of *cloud computing* used in power grid parameter computation and its distribution of computing tasks on multiple tiers.

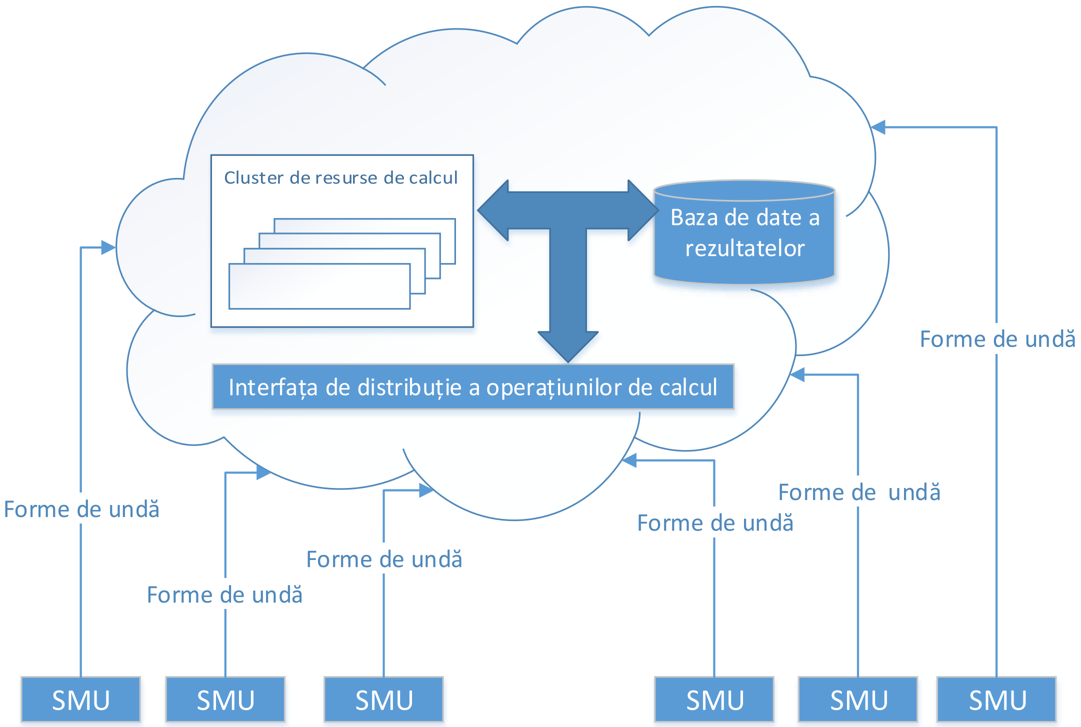

Figura 5. Concept of *cloud computing* used in the centralized computation of the power grip parameters based on the high definition waveforms delivered by SMU's (source: author)

Regarded as a niche equipment, borrowing elements and principles from two major categories of power grid measurement devices, the SMU prototype developed by the author during his research has a high potential for opening new research possibilities and new development areas for improving the power grid state estimation and forecast algorithms based on synchronized investigations of the high definition waveforms collected from all around the power system in a time synchronized fashion.

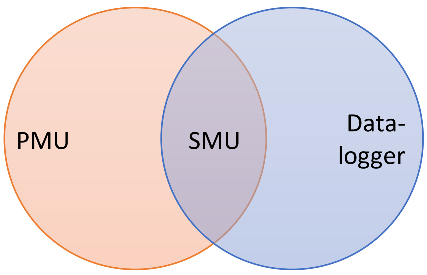

Figura 6. Positioning the SMU devices as a niche equipment (source: the author)

This prototype has the status of a hybrid (comparison in table 1) between the classic PMU devices and data-logger units: performs high definition waveform recordings (like the data-logger) for 3 x voltages and 3 x currents and, in addition, similar to PMU devices, it time stamps each waveform and streams it in real-time.

Description	PMU	Data-Logger	SMU
Conversion of the analogue signal (voltage and/or current) into digital signal on each channel of the power grid node	✓	✓	✓
Number of input measurements channels: ▪ 3 canale (de cele mai multe ori numai u_A, u_B și u_C) ▪ 4 canale (u_A, u_B, u_C și i_{total}) ▪ 6 canale (u_A, i_A, u_B, i_B, u_C, i_C)	 ✗ ✗ ✓	 ✓ ✓ ✗	 ✗ ✗ ✓
Time synchronization of the internal system clock with an exact time reference (e.g.: GPS)	✓	✗[1]	✓
Time stamping all measurements produced by the equipment	✓	✗	✓
Compression of the waveform for subsequent lossless reconstruction	✗	✓	✓
Recording / Storing the high definition power waveform into internal local memory	✗	✓	✓
Usage of the digital signal from the out put of the ADC for producing the estimation of the instanteneous phase angle, frequency and RoCoF (*Rate of Change of Frequency*)	✓	✗	✗[2]
Basic management of the recording inside the storage drive	✗	✓	✓
Memory Management of the recorded waveforms: ▪ Retransmission of the incomplete/missing ▪ Deletion of the transferred and confirmed samples ▪ Reusage of the freed memory sectors	 ✗ ✗ ✗	 ✗ ✗ ✗	 ✓ ✓ ✓
Real-time streaming of data	✓	✗	✓
Details about the data being sent/streamed in real-time: ▪ High definition power grid signals (in compressed manner) ▪ Voltage and currents synchrophasors ▪ Frequencies	 ✗[3] ✓ ✓	 ✗ ✗ ✗	 ✓ ✗[4] ✗[4]
Data transmission method: ▪ Data collection centre initiates TCP or UDP connection towards the measurement device (device is the server) ▪ The measurement device initiates TCp or UDP connections towards the data collection centre (the device is the client)	 ✓[5] ✗	 ✗ ✗	 ✗ ✓
Gestiunea echipamentului de la distanță	✓	✓	✓

Tabelul 1. Feature comparison of the three types of measurement equipment studied (source: the author)

[1] Data-Logger equipment from Schneider Electric and UfE - Umweltfreundliche Energieanlagen GmbH, studied by the author of this thesis do not offer components for time synchronization from GPS reference. Today's generations of „Data-Logger" equipment (also called „Power Meters") from Schneider Electric apparently have this type of time sync component, however it is not completely and officially documented whether this time sync is actually used for the actual timestamping of the recorded data.

[2] In case of using the SMU devices, this calculation is done in a centralized manner having the waveform data from all the SMU's around the powergrid as their inputs.

[3] In case of PMU's, the waveform could be reconstructed out of the transmitted synchrophasors, however the resolution of the resulting waveform is reduced.

[4] Although the calculation of the frequency and phase angle is not done by the SMU itfels, they are still computed by the central correlation centre.

[5] During the tests performed by the author of this thesis on the PMU's manufactured by Artbiter Systems, the author has identified the need of an additional VPN connection initiated from within the power operator's network towards the PDC, in order fot the PDC to initiate later the actual C37.118 connection towards the device located at the power operator's premises via the VPN tunnel.

2.3. Real-Time Data Acquisition in the SMU Prototype

The concept of *real-time* is a current subject, debated in areas where modern computation techniques and data processing apply. In opinion of many researchers like [20, 8], the real time is in fact, a constraint on the deadlines which a computing system has to fulfil when delivering the results of its operations/tasks upon the input data. Regarding the data acquisition process the concept of real time refers to the capacity of the system to guarantee that the acquired signal does not miss samples from the original analogue signal and, in addition, that the acquired signal transitions from state A to state B (for instance from minimum to maximum), within the same tie interval the original analogue signal does.

Currently [8], the whole balance point of the acquisition process lies within the main element of converting the analogue signal into its digital equivalent. This element is called in the literature *analogue-digital converter*. The requirements in various areas where ADC's apply have stimulated intense researches [8, 12, 21] with the purpose of optimizing these ADC modules, in terms of increasing the sampling frequency and bit depth for signal quantization, as well as the increase in the data throughput from the ADC towards the processing units, having a less power consumption. Regardless whether the computation of the power grid parameters is performed by the PMU internally or by the correlation centre (in a centralized manner), the basis of implementing such a precise measurement for the frequency is actually the accuracy of the input signal sampling. This is offered mainly by the quality of the quartz crystal used in the oscillator which generates the sampling clock.

In the past, in order to ensure a high accuracy, a developer for digital measurement systems had to take into consideration very high costs for their DAQ modules. The higher frequency the input signal had and the more precise the studies on the input signal have to be, the higher the performance of the DAQ and as such the costs are.

For the purpose of building the SMU prototype, the author takes into consideration (as described in sections 2.3.1 and 2.3.2. of the thesis) the perspective of using a DAQ based on a regular computer sound card. Due to the modern requirements in personal computing areas [4], such peripherals have been significantly developed and are now capable of delivering the same accuracy as the professional DAQ modules at much lower costs.

The authors of [25, 27, 24] have concluded as well that using the sound cards in nowadays personal computers for measuring the signal frequencies is a feasible alternative to classic DQA modules, as long as an integrated correction of the acquisition offset caused by the quartz oscillator is performed.

Currently, most of the high performance sound cards, already have built-in algorithms to compensate the deviations caused by the quartz crystal imperfections, either integrated in their chips, or built into the software offered by the sound card manufacturer and integrated within the operating system kernel.

The latter option of having the correction built into the software driver, requires an external discipline of the internal computer clock in order to achieve the correction. More specifically, the OS component called kernel uses an internal driver for adjusting on the fly the system clock (including the one used by the sound card driver) according to an external reference, like for instance a high precision GPS receiver.

In many applications where it is expected that the acquired signals have a wider spectrum, with maximum frequencies above 150 kHz (such as the medicine), it is necessary that dedicated high precision FPGA bases equipment is used for data acquisition. In the study [9] done in medical research areas, the external time reference is tightly integrates directly into the DAQ module. In smart-grids research area, the expected frequency domain is however much lower, which makes it possible to use a DQA component such as a sound card without direct hardware integration with the external GPS time reference.

As such, the implementation of the acquisition for the analogue signals ($u_A(t)$, $i_A(t)$, $u_B(t)$, $i_B(t)$, $u_C(t)$ și $i_C(t)$) in a hybrid SMU equipment can be achieved without any compromise on precision, by using a regular A/D conversion component based on a computer sound card, thus offering much lower costs. The SMU prototype developed by the author is based on a Maya 44 sound card for the acquisition of the analogue power grid-specific signals and converting them into digital signals, in order to subsequently transmit them in a compressed and time stamped manner towards the centralized correlation and processing.

The authors of [23] use a sound card as well for their project, which consists in building a virtual oscilloscope. In contrast to the author's SMU prototype, their project [23] aims to just visualize the mono- or bi-channel test waveforms, do minor measurements locally and comparisons of two simultaneous signals. Both prototypes use attenuation circuits in order to protect the physical inputs of the sound cards, since the power grid signals have much higher amplitudes than what the sound cards supports (in our case, Maya 44 supports only ±48V inputs)

2.4. SMU's Internal Clock Synchronisation with GPS Reference

Theory of synchronized measurements assumes that all equipment involved in data acquisition synchronously have a common time reference synchronized via a precision clock. By *timestamp* one understands the the internal clocks of each of the equipment pieces „count" the time units using the same clock frequency, in the sam rithm and starting form the same common origin and that the currently counted time units is the same for all synchronized devices.

Today an accepted solution from economic point of view and relatively easy to implement technically is the usage of GPS receivers as precise reference clocks, which besides the NMEA data (like geographic position, motion and UTC) received from the satellite, can provide a timely constant and steady synchronization signal, which actually marks the beginning and the end of a leap second, with a high accuracy.

PMU equipment currently used in the electric network integrated WAMPAC uses GPS timing to achieve a sinusoidal reference within the DSP controllers. Great PMU equipment manufacturers such as Arbiter Systems, Schneider Electric and SEL integrate into each produced and sold unit such a GPS receiver for time synchronization.

An alternative, more economically accessible could be the use of a common NTP server (*Network Time Protocol*) for a s small group of devices, which delivers the time reference via network communication.

In section 2.4.1, 2.4.2, 2.4.3 and 2.4.4 of the thesis, the author combines detailed studies about the current state of time synchronization solutions using either NTP protocol or protocol PTPv2 comparing them one by one from functionality and integration possibilities with the SMU prototype, by attempting to implement their principles into the source code of the SMU modules.

The infrastructure to achieve the timestamp reference in the SMU prototype is a typical communication stack in a modular operating system (like Linux), as illustrated in the below figura 7.

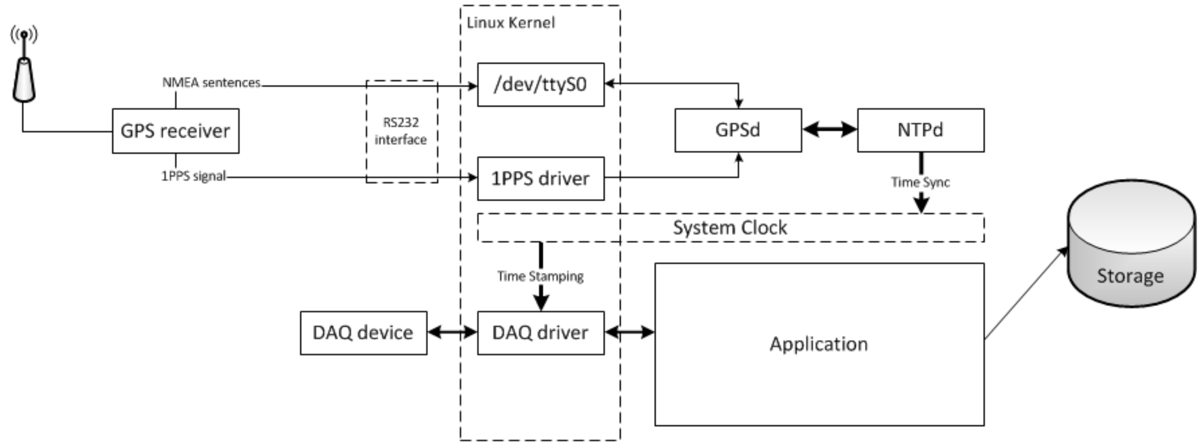

Figura 7. Typical architecture of a GPS integration into a Linux operating system for synchronizing the internal clock (source: the author)

It can be seen in Figure 7 that the interface between the system clock of the SMU and the GPS receiver is accomplished by the local NTP service which communicates tightly with the *GPSd* driver responsible with the correlation of the NMEA messages with the 1PPS pulses received via the RS232 interface. The $GPGGA NMEA messages indicate the number of the current second counted by the GPS receiver and transmitted to the Linux kernel via GPSd driver and, whereas the 1PPS pulses indicate exactly when the time interval of the current second begins and ends. Finally the whole suite of applications run on top of the Linux OS (like the data acquisition module) receive automatically the time reference set by the kernel.

In case of a group of devices the author proposes the designation of a dedicated time server, responsible with keeping the time reference of the other devices up-to-date with the GPS reference, all the other devices not being directly connected to the GPS receiver, but using either NTP or PTPv2 to obtain their time stamps. In section 2.4.2 the author states that the two protocols can complement each other, because the NTP protocols provides the correct time to all equipment, while the PTPv2 provides the same time to all equipment regardless what this might be.

In section 2.4.4. the author describes step by step the implementation of the time synchronization solution of the SMU prototype with the GPS time reference by means of NTP component, building thus a so-called „NTP stratum-1”. The SMU prototype developed by the author is based on an embedded Linux platform, on which the NTP component is installed and integrated with the SHM, PPS and GPS_NMEA drivers illustrated in table 4 in the thesis. During the building of the SMU prototype the author uses a GPS module made by Quectel and connected to the main hardware platform via RS 232. [35].

The assembly of terminals RXD, TXD, VCC and GND represent the logical serial interface where the GPS receiver sends the NMEA statements, like those depicted in figura 8, where one can see that the GPS sends the whole set of position, time, speed, and satellite visibility.

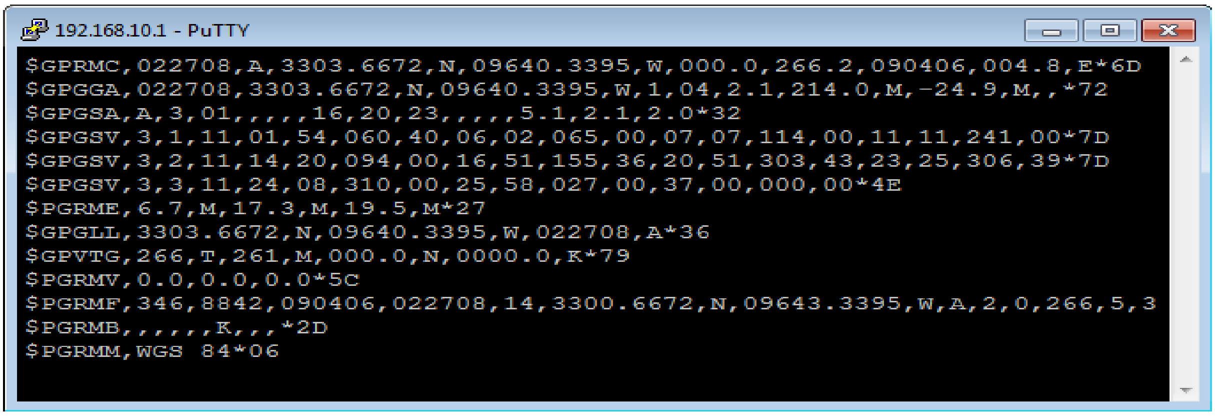

Figura 8. NMEA parameters received by the SMU prototype from GPS (source: [5])

For the purpose of obtaining only the time reference from the GPS receiver, the GPGLL, GPRMC, GPVTG, GPGGA, GPGSA and GPGSV parameters have been disabled from within the GPS receiver’s microcontroller, using the following set of commands: `echo -n -e "$PMTK314,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0*29\r\n" > /dev/ttyS0` as shown in figura 9 as well.

```
192.168.3.118 - PuTTY
login as: root
root@192.168.3.118's password:


BusyBox v1.19.4 (2013-03-06 20:07:44 UTC) built-in shell (ash)
Enter 'help' for a list of built-in commands.

############################################################

Prototip SMU pentru aplicatie in domeniul Inginerie Elctrica
(masurarea, achizitia, stocarea si transmisia in timp real
a formelor de unda din nodul de retea energetica considerat)

Prototip bazat pe OpenWRT linux 12.09

############################################################

root@SMU:~# stty ispeed 9600 ospeed 9600 -F /dev/ttyS0
root@SMU:~# echo -n -e "$PMTK314,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0*29\r\n" > /
dev/ttyS0
root@SMU:~#
```

Figura 9. The modification of the GPS parameters via GPS receiver's microcontroler (source: the author)

The purpose of running this command to modify the GPS parameters is actually to minimize the quantity of data delivered by the GPS receiver through the serial interface at each iteration (once every second). This way the delays produces by the serial interface communication is significantly reduced. As one can see in figura 9, at the end of the command $PMTK314 the number 29 is entered. This represents a verification hash for the integrity of the command and it is calculated using the in figura 10.

```
function updateChecksum(cmd){
  // Calculeaza hash prin XOR pe toate caracterele
  var checksum = 0;
  for(var i = 0; i < cmd.length; i++) {
    checksum = checksum ^ cmd.charCodeAt(i);}

  // Converteste in hexa (baza 16, majuscule, grupul cel mai semnificatif primul)
  var hexsum = Number(checksum).toString(16).toUpperCase();
  if (hexsum.length < 2) {
    hexsum = ("00" + hexsum).slice(-2);}

  return hexsum;
}
```

Figura 10. The source code of the CRC calculation function used at the end of the GPS configuration commands (source: the author, adapting from [30])

The assembly of terminals DCD and GND, which also belong to the same RS 232 physical interface belong however to a different logical interface, which ensures the 1PPS signal stream, the two interfaces being correlated in order to achieve the high precision time synchronization with the GPS reference clock. In case of implementation with SHM (*Shared Memory*), the two logical interfaces are listed inside the Linux OS as a single virtual file in the RAM `/dev/shm`.

This way of implementing the time sync requires the usage of the GPSd component, as well as the NTPd component, the first one being responsible with reading the data from the two interfaces and combingin hem as a single source for NTPd. The NTPd configuration in this case for the particular case of the SMU prototype is illustrated in figura 11.

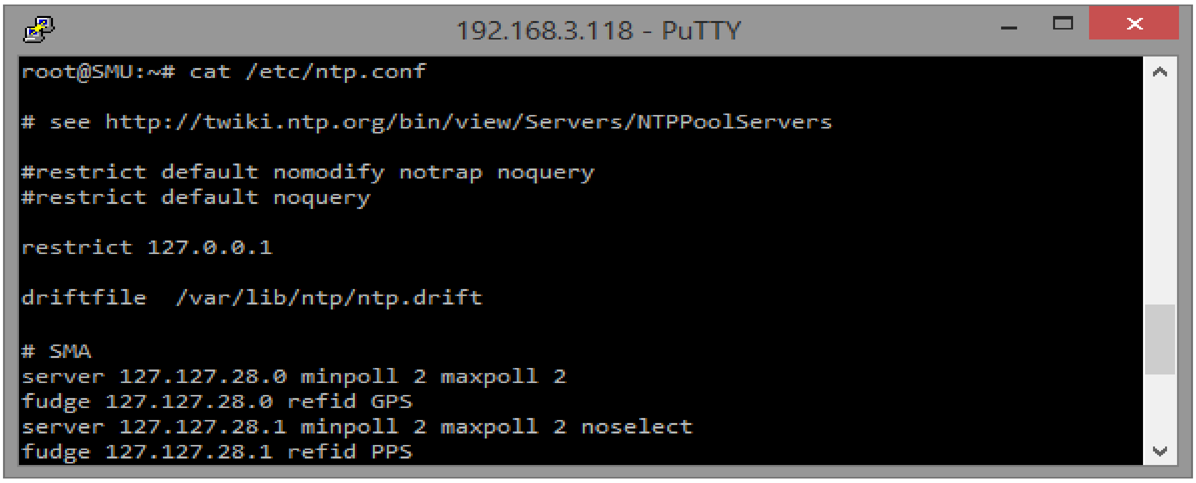

Figura 11. The NTPd configuration within the SMU protorype
(source: the author)

One can note that the NTP uses two pseudo-IP addresses such as „127.127.X.Y" to indicate that the time reference is localy connected and not via network. When writing „127.127.X.Y", the variables X ans Y represent the following:

- Variable X represents the driver used by NTP (according to the list in table 4 in thesis)
- Variable Y represents the index of the interface of the driver X

This way of implementing the time sync using *Shared Memory* has the advantage of a wider compatibility between the GPS receiver and the NTP component via GPSd module.

The second method of implementing the time sync uses GPS_NMEA and PPS as two different entities, collecting two separate streams of data towards the NTP component which interprets them separately. This requires that a special Linux kernel module is enabled for applying the time discipline proposed by the NTP component based on the GPS reference clock and 1PPS pulse flow. Using this approach inside the SMU prototype running on the Embedded Linux involves recompiling the Linux Kernel with the parameters `Config_Serial_8250`, `Config_PPS` și `Config_PPS_Client_8250`

The existence of the Kernel driver required for accepting the 1PPS pulse flow after recompilation is verified by running the command `dmesg | grep "pps" | "8250"` shown in figura 12 and in figura 13.

```
192.168.10.1 - PuTTY
drivers/pps/procfs.c: procfs enabled
drivers/pps/pps.c: LinuxPPS API ver. 2.431-b2 registered
Serial: 8250/16550 driver $Revision: 1.90 $4 ports, IRQ
sharing disabled
serial8250: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
serial8250: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A
```

Figura 12. Staus of the Kernel module responsible with PPS (source: [5])

```
192.168.10.1 - PuTTY
"pps_core":"pps"[82]: New message from PID 15846 (flags 0)
"pps_core":"pps"[189]: PPS_FETCH: source 0
"pps_core":"kapi"[158]: capture clear seq #2652 for source 0
"8250":"8250"[1273]: serial8250: PPS clear event at 25520476
"pps_core":"kapi"[147]: capture assert seq #2653 for source 0
"8250":"8250"[1269]: serial8250: PPS assert event at 25521276
"pps_core":"kapi"[158]: capture clear seq #2654 for source 0
"8250":"8250"[1273]: serial8250: PPS clear event at 25521476
"pps_core":"kapi"[147]: capture assert seq #2655 for source 0
"8250":"8250"[1269]: serial8250: PPS assert event at 25522276
"pps_core":"pps"[82]: New message from PID 15846 (flags 0)
"pps_core":"pps"[189]: PPS_FETCH: source 0
"pps_core":"kapi"[158]: capture clear seq #2655 for source 0
"8250":"8250"[1273]: serial8250: PPS clear event at 25522476
"pps_core":"kapi"[147]: capture assert seq #2656 for source 0
"8250":"8250"[1269]: serial8250: PPS assert event at 25523277
"pps_core":"kapi"[158]: capture clear seq #2657 for source 0
"8250":"8250"[1273]: serial8250: PPS clear event at 25523477
```

Figura 13. Events generated by the 1PPS pulse flow (source: [5])

After validating that the PPS kernel module is functional (according to the previous tests) the NTP component within the SMU prototype is configured to communicate with both logical interfaces at the same time according to the configuration file written by the author and illustrated in figura 14.

```
192.168.3.118 - PuTTY
root@SMU:~# cat /etc/ntp.conf

# see http://twiki.ntp.org/bin/view/Servers/NTPPoolServers

#restrict default nomodify notrap noquery
#restrict default noquery

restrict 127.0.0.1

driftfile  /var/lib/ntp/ntp.drift

# GPS(NMEA)+PPS
server 127.127.20.0 mode 16 prefer
fudge 127.127.20.0 flag1 1 flag2 0 flag3 0
```

Figura 14. NTP Configuration parameters for GPS_NMEA and PPS (source: the author)

While performing the implementation steps for the time sync of the SMU prototype with the external precise GPS reference, a compatibility issue occurred between the classic Linux console, associated during boot time with the serial port used at the same time by the PGS receiver /dev/ttyS0. The conflict occurred through this incompatibility caused the the GPS receiver's NMEA commands to be wrongly interpreted by the NTP component. This issue was successfully solved by the author of this thesis by modifying the kernel boot parameters `console=tty0` and `console=ttyS0,38400n8` from the boot sequence.

2.5. Data Storage within the SMU Prototype

Through the SMU prototype, as a hybrid device between the PMU and data-logger concepts, it is expected that the electric phenomena in those power grid nodes where the instability is the highest or has the highest chances to occur are studies/analysed/understood. Therefore the SMU equipment itself has to have a robust construction of both hardware and software components and thus show a maximum tolerance to instability phenomena (for instance: repeated voltage outages, surges, frequency fluctuations, or other disturbance of the voltage or current sine waveform) in that particular node.

This section describes in detail the organization, construction and the effective implementation of the software component within the SMU prototype, which besides performing a reliable waveform recording, ensures the management of the recorded data (verification, retransmission where applicable, overwriting and deletion of old and confirmed data).

The final permanent storage of the measured data (that is the high definition power signal waveforms) out of the considered power grid node is done actually by the PDC (or data correlation centre which receives the data from all the other SMU's). The SMU device itself has an important role by storing temporarily the waveform data in its own internal memory. Whenever one speaks about the real-time transmission of huge amounts of data via a public network such as the Internet, then the concept of „*bottleneck*” has to be taken into consideration, being caused by instable and unpredictable transfer speeds (sometimes high enough, but sometimes insufficient) through the considered network[1], as shown in figura 15.

[1] More about these aspects and the studied and proposed solutions are described in chapter 2.6.3 in the thesis

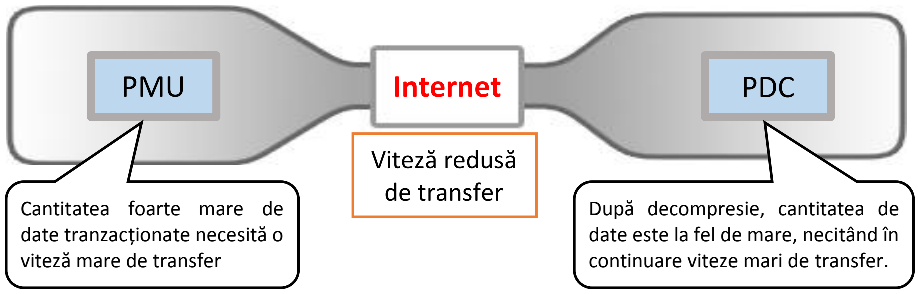

Figura 15. The bottleneck problem, in case of systems with very high data flows. (source: the author)

The logic of the prototype module used for the management of the memory (as part of the embedded system built by the author) is also presented in this section of the thesis. Because the data volume is very high and the storage capacity is limited, the SMU prototype decides when the old data is deleted and overwritten in the internal memory. This section highlights the factors on which the storage module decides whether the data is safe to be deleted or not.

In contrast to the way the classic PMU's studied by the author so far store their data, the SMU prototype uses an internal storage with a much higher capacity (approximately 250 ~ 500 GB) as a set of hard-drives configured in a RAID1[1], on which the SMU stores temporarily the waveform data. Although this data has been sent to the PDC, this data still remain on the SMU's memory for a limited period of time, until the PDC confirms the receiving and the integrity of the data.

Once the PDC confirms this the storage module within the SMU prototype marks that particular segment of the waveform as being safe to be deleted or overwritten, when the capacity reaches its limit.

The first part of this chapter describes the construction of the embedded SMU prorotype software, which is vital for the functioning and device management in "unfriendly" powergrid conditions, as those described at the previous page. This robust construction has been achieved by the author on a platform with x86 architecture processor starting from the OpenWRT embedded Linux version 12.09 [33]. Due to their process and real-time execution oriented nature, the embedded systems based on Linux are the preferred ones by most of the measurement equipment manufacturers. To this category belong some of the PMU devices as well, which are now part of the complex WAMPAC networks. Manufacturers like Schneider Electric și General Electric chose modified Linux versions for integrating with their DSP equipped PMU's. On the other side, the manufacturer Arbitter Systems integrates a

[1] The term RAID represents „*redundant array of inexpensive disks*" and refers to a concept of writing the data on several hard-disks in a redundant manner, in order to protect the written data against the power outages.

proprietary operating system into their 1133A model, about which no information has been currently disclosed.

The SMU prototype has been built on a *Single Board Computer* AAEON PFM-540I format PC/104 with AMD Geode LV800 processor having an x86 architecture and a „NAND flash" Compact Flash Type 1 memory module for the OS (figura 16). It is interfaced with an Ethernet communication module (Intel 82551xER 10/100 MBit) on the PCI bus, one Wireless IEEE 802.11 a/b/g/n module (Atheros) on the serial bus, one GPS/GLONASS (Quectel) receiver on the RS232 bus, including 1PPS signal in real-time and one sound card (Maya 44) cu 6 synchronous channels on the serial bus.

Figura 16. The SBC (single board computer) module used for the SMU prototype (source: http://www.aaeonusa.com/download/datasheet/PFM-540I-RevB.pdf)

In contrast to the method chosen by the author of this thesis to construct the SMU prototype, other manufacturers such as National Instruments offer, for example, dedicated platforms for developing and integrating embedded software for their CompactRIO controllers [32] based mostly on FPGA technology. This method involved however much higher costs, compared to the total costs of the solution proposed by the author based on *embedded Linux*.

In the section 2.5.2 of the thesis, the author details the construction of the embedded operating system used to create the storage of the SMU prototype, which is divided into three sections of the storage:

- The „*read-only*" zone similar to a PROM memory, programmed/compiled according to the steps described in section 2.5.3 of the thesis

- The zone with permanent storage, used by the embedded system to store the configuration parameters, such as IP address for the communication, Wireless parameters, network authentication data and other dynamic data delivered by the PDC.
- The zone with temporary storage, which overlaps as a structure with the „*read-only*” memory and does not contribute directly to the actual storage of the waveform data, but only plays a vital role in buffering the waveform streams into the network

The profile used for the compilation of the embedded Linux system and generating the first two storage zones for the SMU prototype is described in figura 17. One can identify there the definition of the dependent module list, the reference name, and the evaluation function.


```
define Profile/SMU
  NAME:=SMU
  PACKAGES:=grub jshn kernel kmod-3c59x kmod-8139too \
        kmod-e100 kmod-e1000 kmod-input-core kmod-ipt-conntrack \
        kmod-ipt-core kmod-ipt-nat kmod-lib-crc-ccitt kmod-libphy \
        kmod-natsemi kmod-ne2k-pci kmod-nls-base kmod-pcnet32 \
        kmod-r8169 kmod-sis900 kmod-sound-core kmod-tg3 kmod-usb-audio \
        kmod-usb-core kmod-via-rhine kmod-via-velocity libblobmsg-json \
        libip4tc libiwinfo libiwinfo-lua libjson liblua libnl-tiny \
        libubox libubus libubus-lua libuci libuci-lua libxtables lua \
        luci-i18n-english luci-lib-core luci-lib-ipkg luci-lib-nixio \
        luci-lib-sys luci-lib-web luci-mod-admin-core \
        luci-mod-admin-full luci-proto-core luci-proto-ppp luci-sgi-cgi \
        luci-theme-base luci-theme-openwrt ubus ubusd uhttpd usbutils \
        libusb-1.0 libpthread librt zlib kmod-crypto-core \
        kmod-crypto-arc4 kmod-crypto-aes wpad kmod-cfg80211 \
        kmod-mac80211 kmod-lib-crc-itu-t kmod-rt2x00-lib \
        kmod-rt2x00-usb kmod-rt2800-lib kmod-rt2800-usb alsa-utils \
        alsa-lib libncurses terminfo hostapd-mini kmod-input-evdev \
        kmod-hid kmod-usb-hid cmus libncursesw libmad libvorbisidec \
        libflac
endef

define Profile/SMU/Description
        Profil modificat pentru SMU embedded
endef
$(eval $(call Profile,SMU))
```

Figura 17. The definition of the compilation profile for the embedded operating system used for the SMU platform (source: the author)

The complete list of the modules included in the compilation process of the read-only memory image is represented in table 6 within the thesis. Some of the module groups have the pure role of debugging the system and are used only in pilot and testing phases. Therefore they can be safely excluded when the final compilation of the SMU device is performed.

The last steps performed by the author before the actual compilation of the SMU’s read-only memory is defining the list of default configuration files. In the source directory the author defines a library of these files, and the references to it are then used within the compilation process. The structure of these files is shown in figura 18.

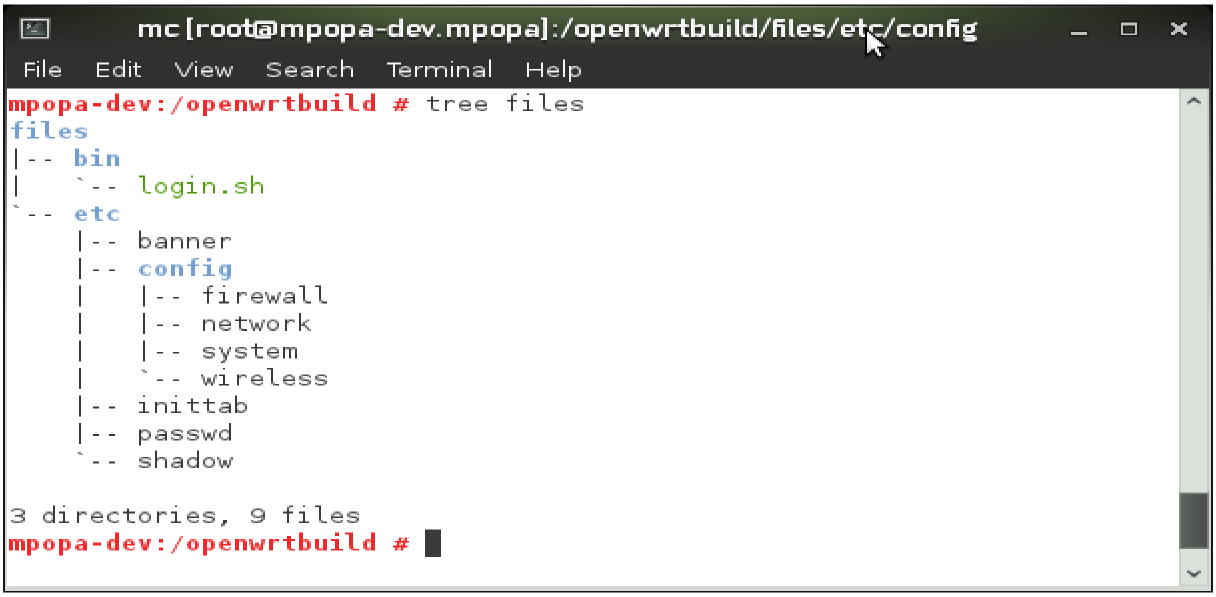

Figura 18. The structure of the default configuration files compiled into the read-only memory of the embedded system (source: the author)

The binary image of the SMU#s read-only memory is finally compressed using the LZMA algorithms, resulting a „*SquashFS*” partition of a total size of 8MB. The author used the syntax:
`make image PROFILE=SMU PACKAGES=”-dnsmasq -ppp -ppp-mod-pppoe -kmod-ipt-nathelper” FILES=files/`

The two phases of constructing the image triggered by the above command are presented in more detail in the section 2.5.3 of the thesis. The binary image of the embedded system is then written at a physical level on the Compact Flash support using the command `dd if=openwrt-generic-x86- of=/dev/sdb` as shown in figura 19.

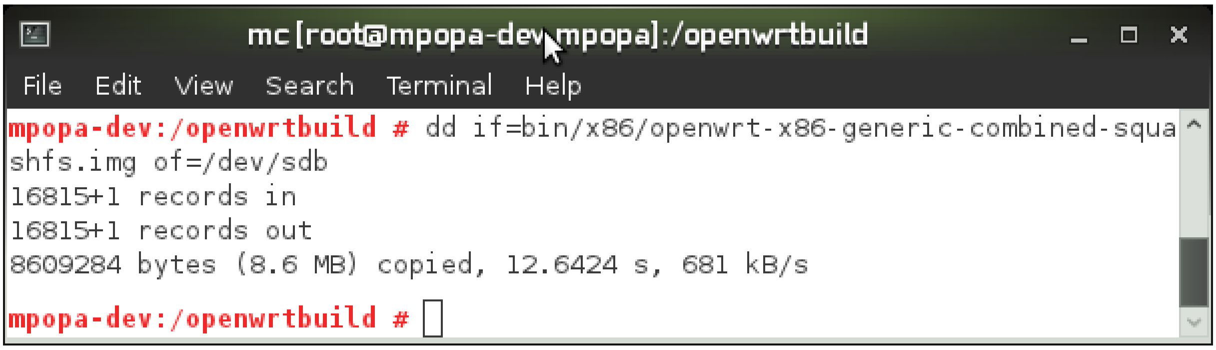

Figura 19. The result of the embedded image compilation (source: the author)

According to the aspects presented in chapter 2.2.1 from the thesis, one of the main differences between the PMU equipment and the SMU prototype is the shifting of the place where the data transfer occurs. The succession of the blocks A, B and D in the case of a SMU prototype is shown in figura 20.

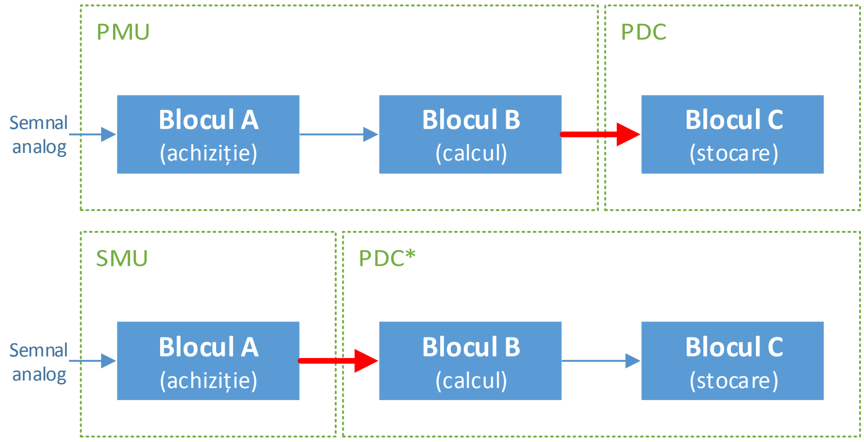

Figura 20. The shift of the place where the data transfer takes place / difference between the PMU (up) and SMU (down) (source: the author)

This shifting implies the increase of the data quantity which needs to be transported from the SMU to the PDC, which also implies a more careful management of the data transmission, because the risk of inconsistency also increases once the data quantity increases. Therefore the need that the SMU prototype stores the data for a much longer time span arises after they have been sent to the PDC, with the purpose of being able to retransmit it in case connectivity problems occurred.

The second consequence of the shifting illustrated in figura 20 and which results out of the necessity to store the waveform measurements for a longer period of time, is the fact that the storage media responsible for the data retention has to be protected against power outages itself, by using an algorithm to minimize the data write operations on this storage media and a hardware UPS-like module.

In chapter 2.5.4. of the thesis the prototype of the storage module is also presented in detail. This module implements a simple logic to enable/disable some functions dependingon the state of the power supply of the SMU. Their sequence is shown in figura 21.

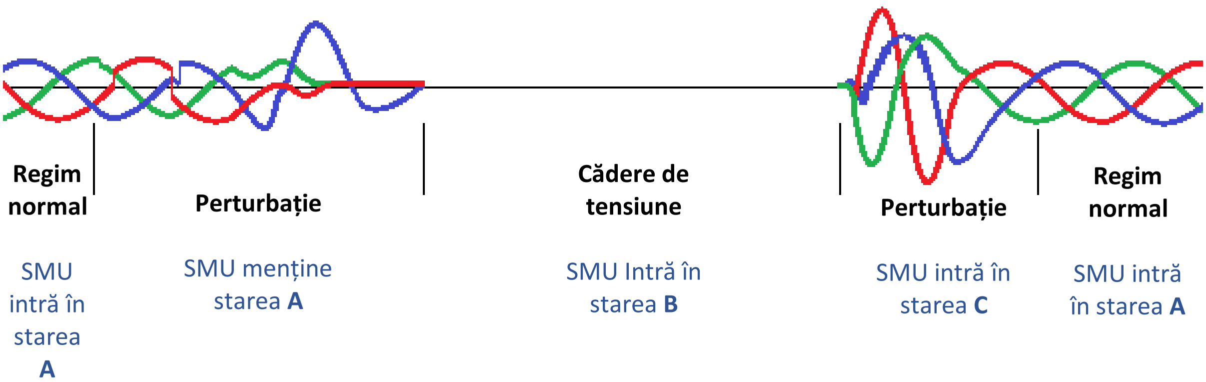

Figura 21. The sequence of the SMU storage management states (source the author)

Based on the above sequence, the following state chart results, as shown in figura 22, where the transitions, conditions and the characteristics of each state are illustrated.

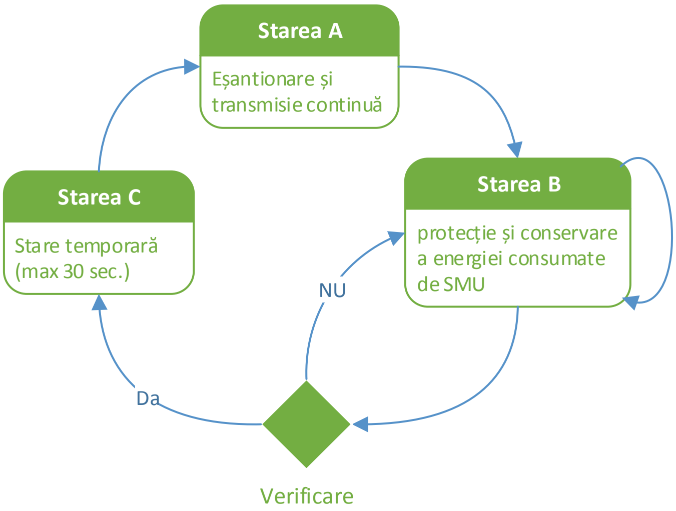

Figura 22. The state chart of the storage management module within the SMU prototype (source: the author)

State **C** (transitive state) is an intermediary phase between the state B and state A, where the storage management module stays for a short period (approximately 30 seconds). This state resembles much the state A. It differs only by the fact that the storage of the waveform samples during the recovery of the voltage is done temporarily on the Compact Flash storage media, following to be moved permanently to the hard-drive media once the module enters back the state A (figura 22)

The goal of the algorithm proposed by the author is to maximize the amount of waveform records, despite the voltage drop affecting the SMU; to ensure that the waveform corresponding to the voltage recovery time are also recorded; minimize energy consumption requirements from UPS, while the voltage of the node network is down and ensuring sustainable storage and transmission of the waveforms recorded during the sag after returning voltage.

Studies and tests conducted by the author in this research and the observations described by the author in the preceding chapters are unique at the moment in literature, because no other integrated measurement/acquisition system used in power engineering has this type of logic of managing the recorded data, transmitting it and conforming the reception before takin other decision in regard to the stored data. In addition no other embedded system offers this ratio between the multitude of functions and the degree of fault tolerance and while the voltage is affected by disturbances.

2.6. Real-Time Data Transmission Made by the SMU Prototype

Data transmission is a key point in high precision instrumentation. In power engineering, more specifically, in research areas of smart grids, where the measurement of the parameters needs to be achieved simultaneously in as many points as possible for a complete evaluation of the grid state, it is necessary that the acquired data from such devices place on-field is done in real-time. This requirement imposes the usage of a protocol adequate to the nature of the data being transmitted. For example, in the case of WAMPAC networks where the PMU's are the primary element of data acquisition the protocol C37.118 has been developed. [15].

The studies [11, 19, 16] from the last years on the WAMPAC networks have identified and optimized the data transport through this protocol. The last version of C37.118 published in 2011 [15] specifies slight modifications in the architecture of the data packets in order address the new requirements for security and data integrity. However when it comes to transporting large amounts of data (like the high definition power signal waveforms), this protocol starts losing its compatibility with the new requirements described in sections 2.6.1 and 2.6.2 in the thesis.

Currently the balance point in monitoring the power grids via WAMS and WAMPAC networks lays within the computation of the exact angle and instantaneous frequency by the PMU devices, placed in a strategic manner in certain grid nodes, as well as on the delivery of the computation results to the PDC's in the shortest time possible. [1, 2]

The integration between the C37.118 protocol and the IEC61850-90-5 standard is achieved on multiple layers within the simplified OSI model[1] and is shown further in figura 23.

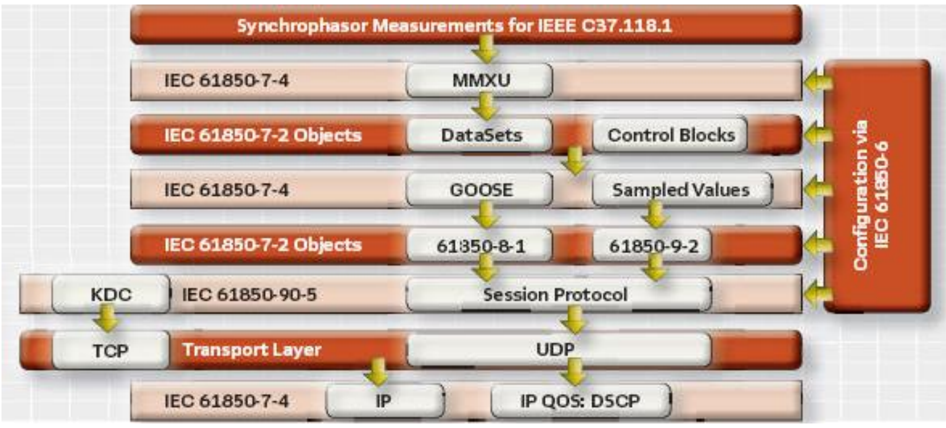

Figura 23. Integration between the C37.118 and IEC61850-90-5 (source: [31])

[1] As discussed in the thesis in section 1.2. at page 7, the OSI model helps the implementation of the communication between two or more network entities, encapsulating the packets from a certain class into packets from a different class, with the purpose of ensuring the compatibility of data.

The current literature, where different initiatives like [18, 13, 22, 7] of implementing WAMPAC networks are described, also mentions about the lack of data encryption of the measurement packets, their structure being visible to any random observer placed on one or more communication lines and capable of intercepting the data traffic.

Based on the data provided by the PMU's, there are various algorithms implemented at a centralised level for estimating the grid state and based on which scientists try to achieve a higher degree of observability, in order for the decision (either automatic or manual) about the state of the power grid to be taken as accurate as possible. Such an estimation is sensitive to the quality of the data offered by the PMU's. There are high risks that the results of the estimation (including decisions to be taken) are wrong if the data transmission from the PMU's has been the target of cyber-attacks, with or without false data injection.

Some manufacturers use open protocols (like IEEE C37.118, IEC 61850-90-5, IEEE 1344), while others implement their own proprietary protocols (like F-NET, SEL Fast Message, Macrodyne și BPA PDCstream), whose documentations are not publicly accessible, hoping that this way they may minimize the risk of false data injection.

At the bottom line, when it comes to the detailed study of a bad phenomenon inside the power grid, one starts to investigate even the smallest details of the waveform from the point in time when the event happened, as well as the moments shortly before and after the event. The Manufacturers of the current monitoring solutions for WAMPAC networks propose the currently the synchrophasors provided by their equipment as solutions to reconstruct the waveform from the time span of the event. However this approach is limited to only the number of frames per second the PMU is able to transmit and it is also vulnerable to data loss over the data transmission lines, if there is no retransmission logic in plce.

This method of reconstructing the waveform based on the synchrophasors has the disadvantage that the resulting waveform is only limited to the numbe of frames per second the PMU can calculate the phasors at[1] and in case that one or more frames does not get transmitted, the reconstruction of the waveform fails completely[2].

On the other side, the transmission of the power signal waveform itself, in a compressed manner as a single TCP packet, has the advantage that all the samples are in one place and are transmitted together, eliminating entirely the risk of incomplete signal; ensuring that the signal sampling is independent of the transmission rate and ensuting that once the TCP packet

[1] Currently, the PMU equipment studied by the author can provided a maximum of 50 frames per seconds, which is equivalent to a very low sampling rate.

[2] This issue has been frequently observed during the data communications between the PMU's and the PDC's in the mico-WAMS network built by the author at MicroDER Lab (tested both UDP and TCP)

arrived at the PDC, the whole power signal wave form is available, including all its high resolution details, just like the original signal.

This approach change in terms of data transmission involves changing the nature of the data which is transmitted. This way, the quantity of data which needs to be transmitted increases significantly creating the need to compress the data prior to sending it, based on the state-of-the-art techniques used in other adjacent technologies, like telecommunications where concept of *codec* (Compression – decompression) is used for data streaming.

As result of author's deep analysis and studies on modern telecommunication solutions for compressing the data in a real-time and lossless manner, described in sections 2.6.3.2 and 2.6.3.3 in the thesis, the author proposes one of the highest performing codecs today, as a solution to compress the power grid signal waveforms. Developed under the open source license, the *FLAC* (*Free Lossless Audio Codec*) is the one proposed by the author taking the following aspects into consideration:

- The codec's ability to achieve high compression rates without data loss (this means that the waveform at the output of a compression and decompression cycle is identical with the initial waveform); this is also the reason for the name *lossless*;
- Time and computing power (for example: number of processor registers used, quantity of RAM, number of machine cycles) necessary for the compression and decompression are relatively low, which indicates that the compression algorithms are well optimized;
- High integrability with other modules which build up the SMU prototype;
- The possibility of seeking the compressed waveform in search for a given time span (start timestamp and duration), which makes it very useful when retrieving lost waveform segment out of the SMU's temporary storage, without having to perform a prior decompression.

The integration of this codec in the SMU prototype has been achieved with the help of the open source libraries: `gstreamer`, `libgstreamer`, `libgrtsp`, `libgstinterfaces`, `libgstnetbuffer`, `libgstdataprotocol`, `libgstcontroller`, `gst-mod-flac`, `gst-mod-alsa`, `gst-mod-tcp`, `gst-mod-tls`, under PL (General Public License) on different Linux platforms, including the embedded OpenWRT.

Approaching the implementation using open source libraries contributes significantly to reducing the development costs, offering as well a vast applicability in case of other similar signals which require real-time transmission. Integrated into the Linux OpenWRT embedded operating system, the GStreamer open source libraries build up the base of developing many other existing solutions nowadays in Internet for conversions, streaming, editing and media analysis, because they offer the programmer a rich and complex *framework* made up from codecs, interfacing elements, filters, sources and sinks, and data buses (as shown in figura 24)

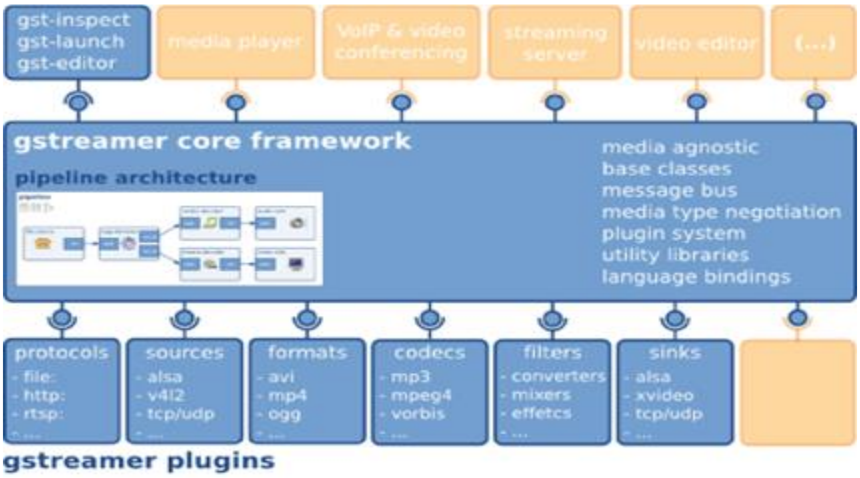

Figura 24. Graphic representation of the GStreamer libraries used in the SMU's transmission module (source: [28])

The main transmission software module of the SMU prototype is shown in figura 25. As the very first step of the application, three main object types are defined by the author: the event of the connection („*on_pad_added*") which is called whenever the interface (or the *pad*) of the block gets connected; the main bus which interconnects all the modules (also called *pipeline* by GStreamer documentation), named „bus" (of type *GstBus*) inside the code and the main control loop of the pipeline, initialized by „`loop = g_main_loop_new (NULL, FLASE)`".

```
#include <gstreamer-0.10/gst/gst.h>
#include <glib-2.0/glib.h>


static gboolean
bus_call (GstBus     *bus,
          GstMessage *msg,
          gpointer    data)
{...29 lines }


static void
on_pad_added (GstElement *element,
              GstPad     *pad,
              gpointer    data)
{
  GstPad *sinkpad;
  GstElement *decoder = (GstElement *) data;

  /* We can now link this pad with the vorbis-decoder sink pad */
  g_print ("Dynamic pad created, linking demuxer/decoder\n");

```

```c
    sinkpad = gst_element_get_static_pad (decoder, "sink");

    gst_pad_link (pad, sinkpad);

    gst_object_unref (sinkpad);
  }

int
main (int    argc,
      char *argv[])
{
  GMainLoop *loop;

  GstElement *pipeline, *source, *demuxer, *decoder, *conv, *sink;
  GstBus *bus;
  guint bus_watch_id;

  /* Initialisation */
  gst_init (&argc, &argv);

  loop = g_main_loop_new (NULL, FALSE);


  /* Check input arguments */
  if (argc != 2) {
```

bus_call

Figura 25. Main transmission module in the SMU prototype
(source: the author)

After compiling the pilot module shown previously, the result of the execution is returned into the main console (figura 26).

mc [root@mpopa-dev.mpopa]:/home/mp/Documents

```
mpopa-dev:/home/mp/Documents # ssh root@192.168.3.118
root@192.168.3.118's password:
BusyBox v1.19.4 built-in shell (ash)
##############################################################
Prototip SMU pentru aplicatie in domeniul Inginerie Elctrica
(masurarea, achizitia, stocarea si transmisia in timp real
a formelor de unda din nodul de retea energetica considerat)
Prototip bazat pe OpenWRT linux 12.09
##############################################################
root@SMU:~# gst-launch -e -v alsasrc ! audio/x-raw-int, endianness="(int)
1234", signed="(boolean)true", rate="(int)16000", channels="(int)4", widt
h="(int)8", depth="(int)8" ! queue ! audiorate ! flacenc ! filesink locat
ion=test.flac
/GstPipeline:pipeline0/GstAlsaSrc:alsasrc0: actual-buffer-time = 191812
/GstPipeline:pipeline0/GstAlsaSrc:alsasrc0: actual-latency-time = 21312
/GstPipeline:pipeline0/GstAlsaSrc:alsasrc0.GstPad:src: caps = audio/x-raw
-int, endianness=(int)1234, signed=(boolean)true, rate=(int)16000, channe
ls=(int)4, width=(int)8, depth=(int)8
Pipeline is live and does not need PREROLL ...
New clock: GstAudioSrcClock
/GstPipeline:pipeline0/GstCapsFilter:capsfilter0.GstPad:src: caps = audio
```

```
/x-raw-int, endianness=(int)1234, signed=(boolean)true, rate=(int)16000,
channels=(int)4, width=(int)8, depth=(int)8
/GstPipeline:pipeline0/GstFlacEnc:flacenc0.GstPad:src: caps = audio/x-fla
c, channels=(int)4, rate=(int)16000
/GstPipeline:pipeline0/GstFlacEnc:flacenc0.GstPad:sink: caps = audio/x-ra
w-int, endianness=(int)1234, signed=(boolean)true, rate=(int)16000, chann
els=(int)4, channels=(int)4, rate=(int)16000
/GstPipeline:pipeline0/GstFileSink:filesink0.GstPad:sink: caps = audio/x-
flac, channels=(int)4, rate=(int)16000, streamheader=(buffer)< 7f464c4143
01000002664c614300000002212001200000000000000003e80670000000000000000000000
000000000000000000, 840000282000000007265666572656e6365206c6962464c414320
312e322e312032303037303931370000000 >
EOS on shutdown enabled -- Forcing EOS on the pipeline
Waiting for EOS...
Got EOS from element "pipeline0".
EOS received - stopping pipeline...
Execution ended after 2918615937 ns.
Setting pipeline to PAUSED ...
/GstPipeline:pipeline0/GstFileSink:filesink0.GstPad:sink: caps = NULL
/GstPipeline:pipeline0/GstFlacEnc:flacenc0.GstPad:src: caps = NULL
/GstPipeline:pipeline0/GstAudioRate:audiorate0.GstPad:sink: caps = NULL
/GstPipeline:pipeline0/GstAlsaSrc:alsasrc0.GstPad:src: caps = NULL
```

Figura 26. The execution result of the main transmission module within the SMU prototype (source: the author)

The main data transmission module within the SMU prorotype, whose execution result is shown in figura 26, has been developed by the author of this thesis in C++ using the GStreamer libraries. On the other hand the same libraries have been also used for testing the child modules individually, debugging and troubleshooting all the emerging issues. These operations have been conducted using the precompiled `gst-launch` executable offered by the GStreamer libraries builtin. The modules tested with the help of this tool, have been contactenated using the symbol „!" (exclamation mark)

For example the syntax shown in in figura 26 represents a concatenation of the main acquisition module with the following ones:

```
gst-launch alsasrc ! audio/x-raw-int,rate=22000,channels=4,width=16
```

The secondary module is the one which controls the output type of the first one (like sampling rate, number of channels and bandwidth). These parameters are not the ones used when defining the compression specifications. They only have effect on the raw acquisition of the power grid-specific signals. As result of specifying these parameters, the main module enters the sampling loop, for ensuring that there are no lost samples from the original signal, delivering all the sampled values directly into the SBC's memory via DMA (*Direct Memory Access*).

Chapter 3

Pilot for Testing and Simulating the SMU Prototype

This chapter shows how the SMU prototype functions within the pilot infrastructure developed by the author for testing purposes, as well as the results of the test, experiments and simulations of the interdisciplinary solutions integrated by the author into the SMU prototype. Due to the multi-disciplinary character of the SMU prototype, the phases of the pilot are presented hierarchically, starting from the high level phases of the functions, until the final technical details combined with principles, methods and solutions from other research areas. In addition there are also operational details offered about the step by step construction of the prototype and its structured testing, as well as about the gradual state of the built-in modules once these have been integrated within the rest of the prototype.

Finally there is also a short economic evaluation of the prototype from costs and profitability perspectives, taking also the usage scenarios into account, as well as the benefits of a possible integration in a classic micro-WAMPAC used for experimental purposes. The contents, configurability, flexibility and benefits of such an approach have been also presented by the author in paper [6], not yet published at the time of writing, however accepted for presentation and publishing in the proceedings of AMPS 2014 (indexed by ISI web of knowledge).

3.1. The Test Environment of the Micro-WAMPAC Infrastructure

For studying the behaviour of the SMU prototype developed by the author, two parallel test environments have been constructed by the author:

- The first environment used for testing is also a semi-productive one and it is based on the current standards and protocols C37.118 and IEC61850-90-5. This environment consists in 5 physical PMU devices dynamically installed[1] in key points of the national power grid/ substations and interconnected from data transmission perspective with

[1] The dynamic character of installing this PMU equipment is caused by the fact that the interconnection to both power grid and data network, as well as the PMUs' configuration for each new installation is done in a significantly shorter time frame, compared to the classic WAMPAC and WAMS networks; in addition it shows a high degree of programmability and flexibility.

the openPDC system installed on top a virtual machine (whose details are shown in figura 27) in the central location within UPB (EB105).

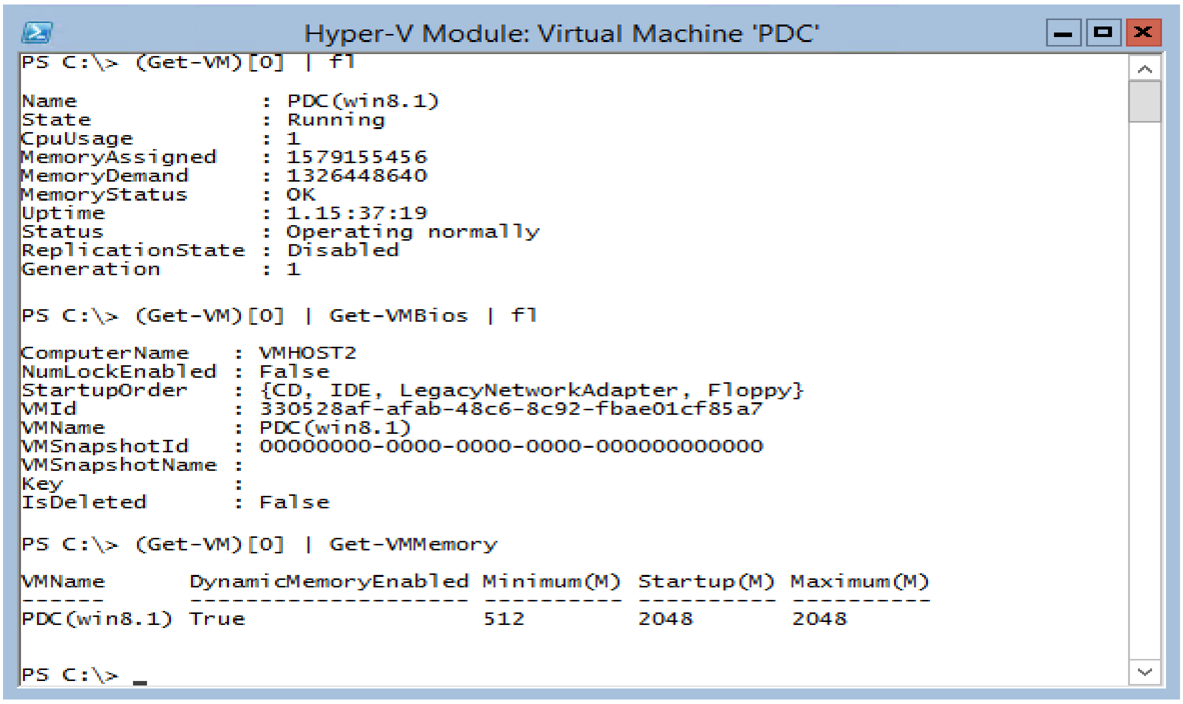

Figura 27. The infrastructure of the virtual openPDC system (source: the author)

The infrastructure of this first comparative test environment consists in a virtual communication base similar to a VPN network (*Virtual Private Network*) via the public network of Internet, between the regional zones of placing the PMU's and the central location where the PDc is placed. This VPN infrastructure is shown in figura 28.

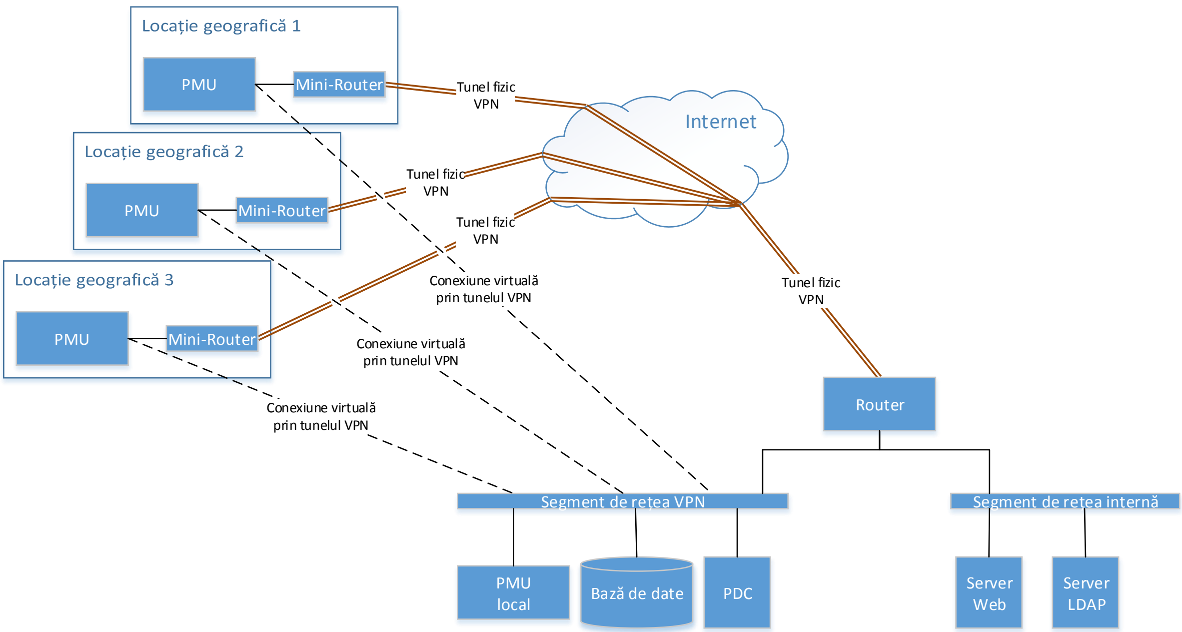

Figura 28. The data communication infrastructure of the first test environment for the SMU prototype (source: the author)

The necessity of integrating a VPN solution comes from the nature of the C37.118 protocol, according to the facts highlighted in chapter 1.2.2 from page 13 in the thesis, whereby the PMU (according to the current implementations) plays the role of a server from TCP/IP point of view.

Although this is not a sustainable solution[1], using the VPN represents the only solution at this time to interconnect the PDC with one or more PMU's locates in the internal networks of different power grid providers/operators (most of the times secured by firewalls according to the state-of-the-art network security principles shown in section 1.2.2.2 of the thesis).

- The second test environment is also a mixed one (consisting in both physical elements and virtual machines). In comprises in the actual SMU prototype, a personal computer having the role of an experimental signal generator for testing, on top of which a minimalistic LabView applications runs in order to simulate the power grid-specific signals and finally a virtual machine with the role of the real-time waveform stream receiver.

  The infrastructure of this test environment is shown in figura 29, where one can note that the first phase of the pilot, the SMU prototype does not acquire the analogue signals directly from the power grid lines, but first (for experimenting purposes) it acquires the analogue signals from the LabView signal generator, as if it took the signals from the power grid.

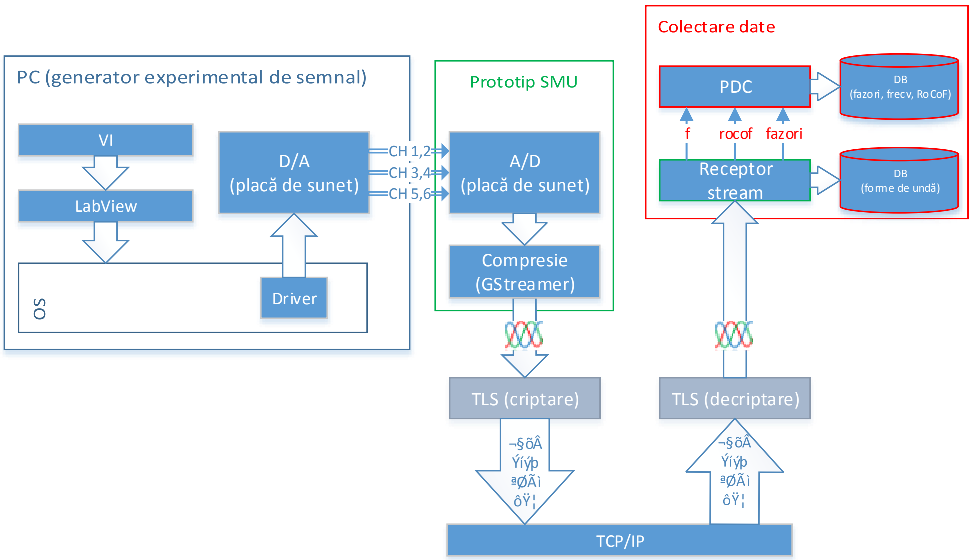

Figura 29. Infrastructura pilot de testare a funcționalităților prototipului SMU (sursa: autorul)

[1] The statement that the VPN is not a sustainable solution in this context, relies on the fact that the TCP/IP packets sent by the PMU, need to be taken through an additional level of processing (encryption, decryption, segmentations, etc.) performed by the VPN solution, which introduces additional delays in the transmission, which is expected to be a real-time one.

The elements marked with green (the SMU prototype, including all its subcomponents and the receiver of the SMU's data stream) represent the scope of the research presented in this thesis and tested during this pilot phase. The elements marked with red are the subject of future related researches proposed by the author for integration with existing PDC systems. The element of experimental signal generation, represented on the left side of figura 29, has the sole purpose of testing the data acquisition temporarily on the platform of a regular computer built ba the author in LabView.

As result of building this temporariy experimental signal generation module, the following waveform (figura 30) is generated at the output of the analogue ports, taking also into account the LabView functions to simulate the power grid disturbances. This output waveform is the unsed by the SMU prototype as DAQ input to otain the results described in section 3.2 of the thesis and synthetized in this summary as well.

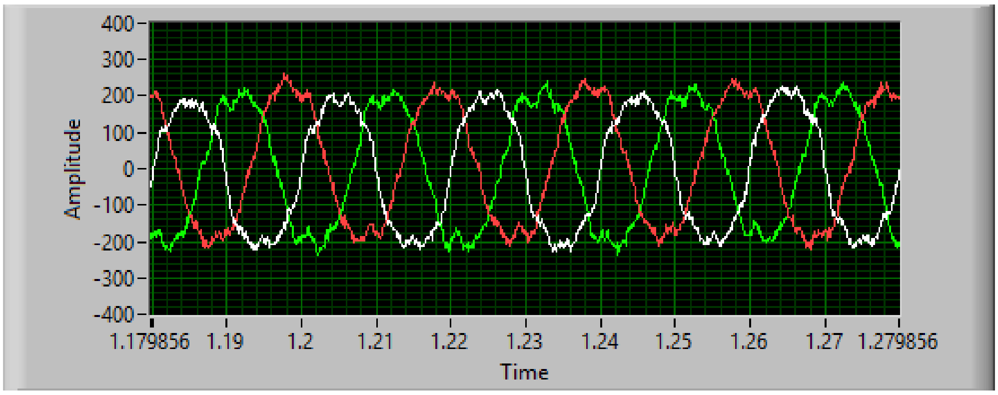

Figura 30. The signal generated by the LabView module developed by the author for testing purposes (sursa: autorul)

When applying this signal to the input of the SMU prototype's acquisition module and starting the debugging steps of the GStreamer libraries (whose structure and functionality has been previously described in section 2.6.3 in the thesis), the author obtains the results presented further in this summary. This results have been also validated also from the point of vire of the criteria and system requirements defined and discussed in section 2.6.1 of the thesis, as well as via comparison with the micro-WAMS experimental network developed partially in the MicroDER Lab and paritall on-field.

For the post-pilot phase the plan is to replace this signal generator module, having actual power grid voltage and current signals at the input of the SMU prototype, via power electronics elements to protect the DQA inputs of the SMU, from the high voltages of the power grid.

3.2. First Results of the Pilot

This chapter describes in detail the routines performed by the software modules of the SMU prototype for achieving the real-time transmission according to the solution proposed by the author in the previous chapters. Three situations can be distinguished in regard to the module interoperability:

- The acquisition of the signal for the temporary storage on the non-volatile memory support of the SMU prototype. This is done by the author by running the command:

  ```
  gst-launch -e -v alsasrc ! audio/x-raw-int, endianness="(int)1234", signed="(boolean)true", rate="(int)16000", channels="(int)4", width="(int)8", depth="(int)8" ! queue ! audiorate ! flacenc ! filesink location=test.flac
  ```

  One can note the usage of the element „*queue*” with the purpose of building up a buffer for the write process on the memory support, where the bandwidth, including the speed is lower than the acquisition. The results of this routine are shown in figura 31.

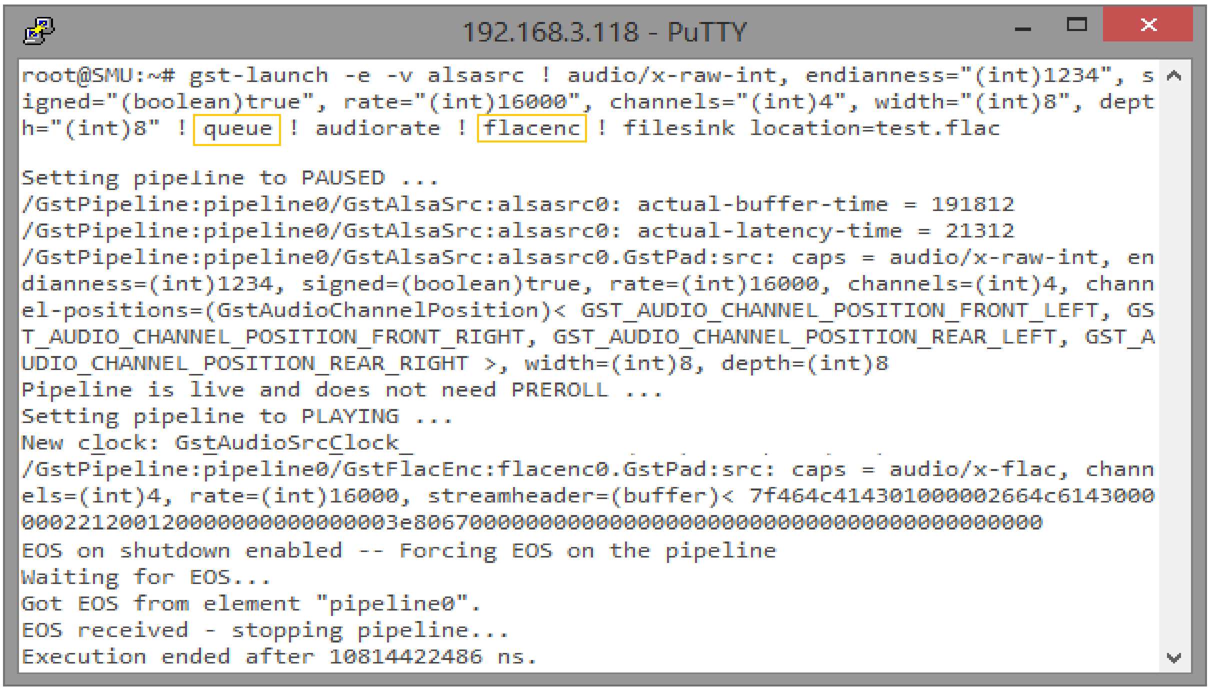

Figura 31. The results of the temporary storage routie for the simulated power grid waveform (source: the author, using the GStreamer libraries)

- The signal acquisition for the temporary storage of the simulated power grid signal as multiple sequential files on the non-volatile memory. This is achieved by the authos using the command:

  ```
  gst-launch -e -v alsasrc ! audio/x-raw-int, endianness="(int)1234", signed="(boolean)true", rate="(int)16000", channels="(int)4", width="(int)8", depth="(int)8" ! queue ! audiorate ! flacenc ! multifilesink location=rec%05d.flac next-file=1
  ```

  The result of this routine is also shown further in figura 32:

```
192.168.3.118 - PuTTY
root@SMU:~# gst-launch -e -v alsasrc ! audio/x-raw-int, endianness="(int)1234", s
igned="(boolean)true", rate="(int)16000", channels="(int)4", width="(int)8", dept
h="(int)8" ! queue ! audiorate ! flacenc ! multifilesink location=rec%05d.flac ne
xt-file=1

Setting pipeline to PAUSED ...
/GstPipeline:pipeline0/GstAlsaSrc:alsasrc0: actual-buffer-time = 191812
/GstPipeline:pipeline0/GstAlsaSrc:alsasrc0: actual-latency-time = 21312
/GstPipeline:pipeline0/GstAlsaSrc:alsasrc0.GstPad:src: caps = audio/x-raw-int, en
dianness=(int)1234, signed=(boolean)true, rate=(int)16000, channels=(int)4, chann
el-positions=(GstAudioChannelPosition)< GST_AUDIO_CHANNEL_POSITION_FRONT_LEFT, GS
T_AUDIO_CHANNEL_POSITION_FRONT_RIGHT, GST_AUDIO_CHANNEL_POSITION_REAR_LEFT, GST_A
UDIO_CHANNEL_POSITION_REAR_RIGHT >, width=(int)8, depth=(int)8
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstAudioSrcClock
/GstPipeline:pipeline0/GstCapsFilter:capsfilter0.GstPad:src: caps = audio/x-raw-i
nt, endianness=(int)1234, signed=(boolean)true, rate=(int)16000, channels=(int)4,
 channel-positions=(GstAudioChannelPosition)< GST_AUDIO_CHANNEL_POSITION_FRONT_LE
FT, GST_AUDIO_CHANNEL_POSITION_FRONT_RIGHT, GST_AUDIO_CHANNEL_POSITION_REAR_LEFT,
 GST_AUDIO_CHANNEL_POSITION_REAR_RIGHT >, width=(int)8, depth=(int)8
/GstPipeline:pipeline0/GstCapsFilter:capsfilter0.GstPad:sink: caps = audio/x-raw-
int, endianness=(int)1234, signed=(boolean)true, rate=(int)16000, channels=(int)4
, channel-positions=(GstAudioChannelPosition)< GST_AUDIO_CHANNEL_POSITION_FRONT_L
EFT, GST_AUDIO_CHANNEL_POSITION_FRONT_RIGHT, GST_AUDIO_CHANNEL_POSITION_REAR_LEFT
, GST_AUDIO_CHANNEL_POSITION_REAR_RIGHT >, width=(int)8, depth=(int)8
/GstPipeline:pipeline0/GstQueue:queue0.GstPad:sink: caps = audio/x-raw-int, endia
nness=(int)1234, signed=(boolean)true, rate=(int)16000, channels=(int)4, channel-
positions=(GstAudioChannelPosition)< GST_AUDIO_CHANNEL_POSITION_FRONT_LEFT, GST_A
UDIO_CHANNEL_POSITION_FRONT_RIGHT, GST_AUDIO_CHANNEL_POSITION_REAR_LEFT, GST_AUDI
O_CHANNEL_POSITION_REAR_RIGHT >, width=(int)8, depth=(int)8

/GstPipeline:pipeline0/GstMultiFileSink:multifilesink0.GstPad:sink: caps = audio/
x-flac, channels=(int)4, rate=(int)16000, streamheader=(buffer)< 7f464c4143010000
02664c614300000022120012000000000000003e806700000000000000000000000000000000000
00000, 84000028200000007265666572656e6365206c6962464c414320312e322e3120323030373
0393137000000000 >

EOS on shutdown enabled -- Forcing EOS on the pipeline
Waiting for EOS...
Got EOS from element "pipeline0".
EOS received - stopping pipeline...
Execution ended after 67985202930 ns.
```

Figura 32. The results of the temporary storage routine of the simulated power grid waveform signal on multiple distributed files (source: the author, using the GStreamer libraries)

- The signal acquisition for the temporary storage of the simulated power grid waveform on the non-volatile memory support, as well as for the real-time transmission as FLAC compressed waveform towards the experimental receiver via TCP/IP. The two routines are executed simultaneously, because of the integration of the "fork" module integration, by which the application initializes two different processor *threads*[1]. This routine is tested using the command below:

[1] Noțiunea de *thread* este folosită în domeniul ingineriei calculatoarelor pentru a defini un și neîntrerupt de operații în cadrul unei subrutine din procesorul unui echipament de calcul. Capacitatea procesoarelor moderne de a executa mai multe astfel de șiruri cu scopul de a eficientiza utilizarea resurselor interne este referențiată deseori sub numele de *multi-threading*.

```
gst-launch -e -v alsasrc ! audio/x-raw-int, endianness="(int)1234", signed="(boolean)true", rate="(int)16000", channels="(int)4", width="(int)8", depth="(int)8" ! queue ! audiorate ! flacenc ! tee name=fork ! queue ! multifilesink location=rec%05d.flac next-file=1 fork. ! queue ! tcpclientsink host=192.168.5.10 port=443
```

Figura 33 illustrates the execution of this routine, as well as its results.

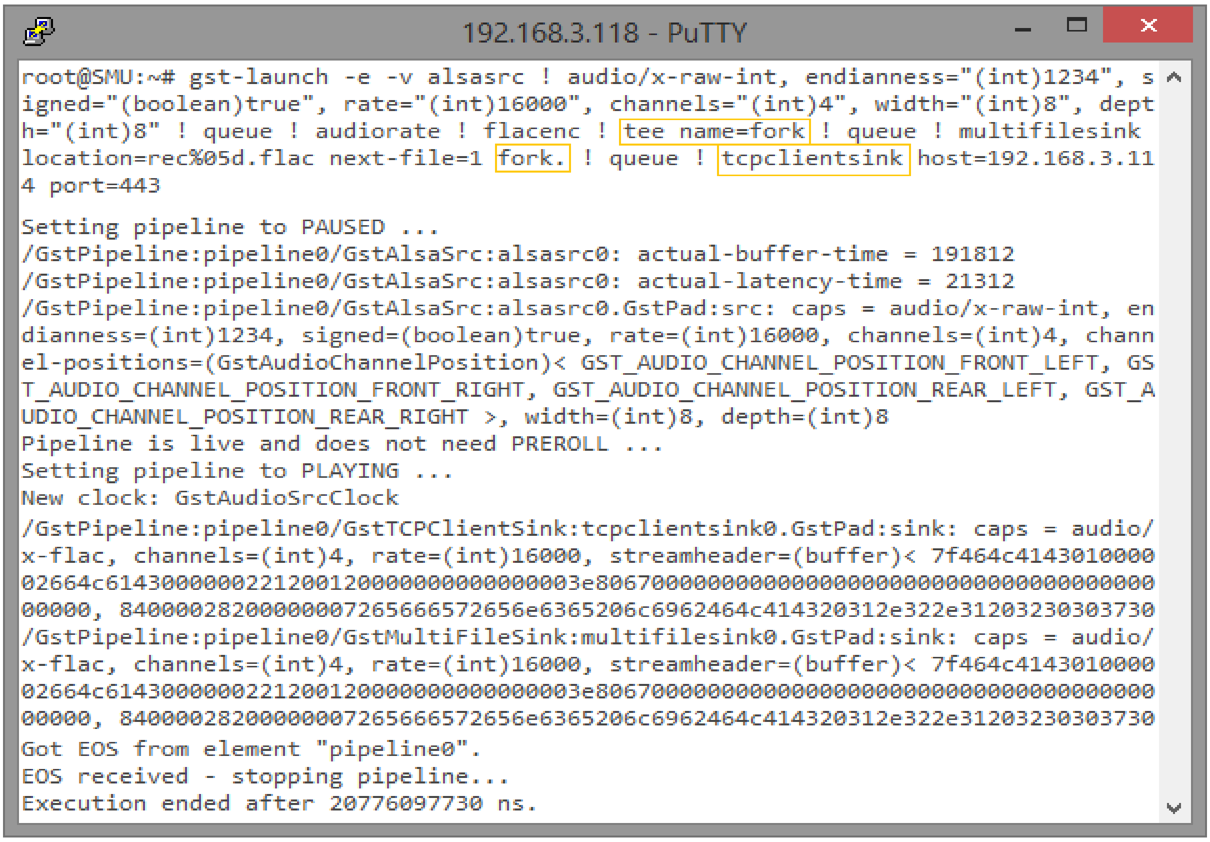

Figura 33. The results of the temporary storage routine of the simulated power grid waveform signal, including the transmission routine (source: the author, using the GStreamer libraries)

Since any data transmission over a communication network requires both a client and a server, the above transmission routine has an equivalent command on the receiving side where the virtual machine runs. This virtual machine plays the role of an experimental PDC.

At the end of the tests one can notice that the receiver of the waveform dumps a local file on its hard drive as result of the stream reception; similarily the SMU prototype generates a file on its memeory support as result of the storage routine, in order to be able to retrieve a lost waveform if the transmission fails for any reason (according to the requirements described in estion 2.5.4. of the thesis).

The data integrity criteria, defined analyzed and presented by the author in section 2.6.1.3 requires that the above two files (the one generated temporarily by the SMU itself and the one generated by the stream receiver upon reception of the waveforms) are identical, or, in other words, have identical MD5 or SHA1 file signatures.

The below Figura 34 shows the fact that as result of the validation tests, the MD5 signatures of the two files are indeed identical, which confirms without any doubts that the data transmission initiated by the SMU prototype is indeed efficient and capable of transmissing large amount of waveform data in a compressed manner without any data loss.

Receptor Prototip SMU (emițător)

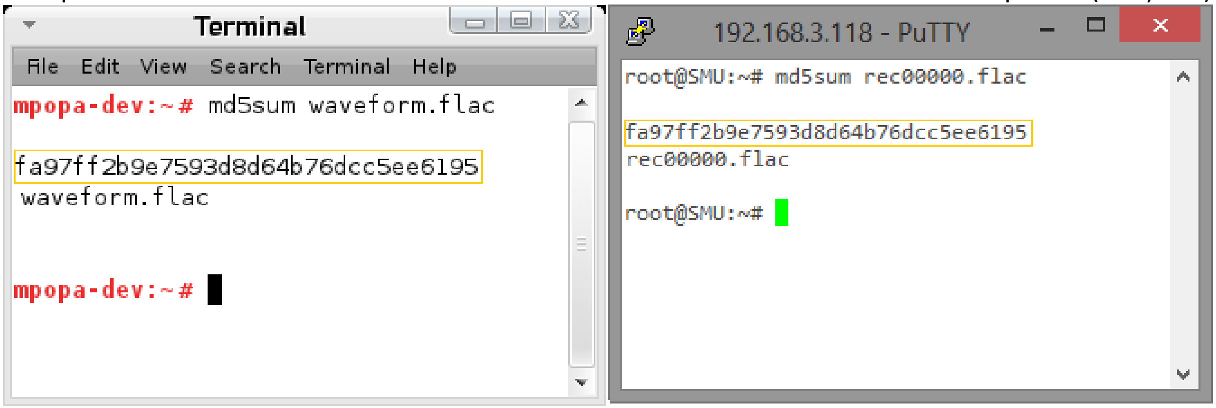

Figura 34. Validation of the fact that the transmitted waveforms by the SMU (right) and those received by the virtual machine (left) are identical (source: the author)

Results show as well that the data transmitted by a classic PMU towards the PDC are not encrypted and can be easily intercepted and altered (figura 36), in contrast to the data transmitted by the SMU prototype towards the receiving virtual machine, which are encrspted and thus secured against cyber-attacks (figura 36)

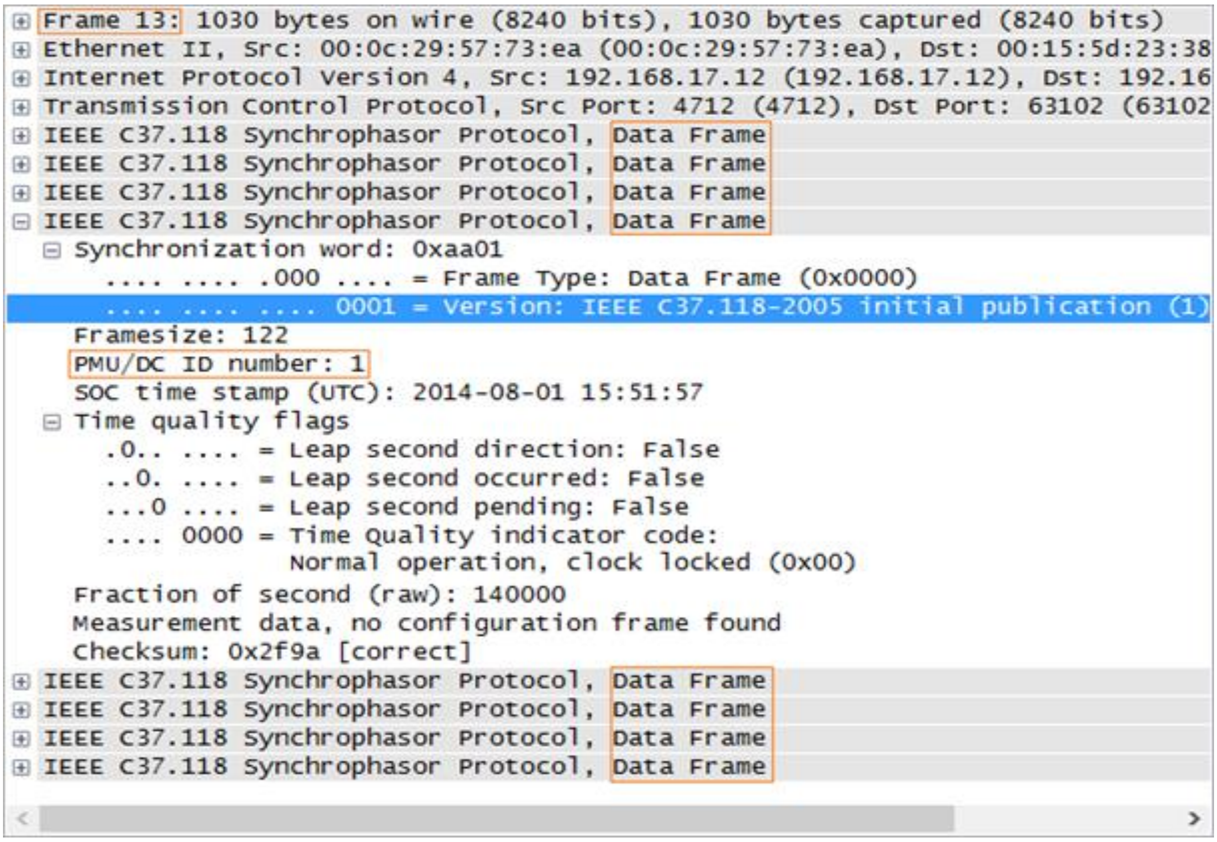

Figura 35. Validation of the fact that the PMU equipment within the WAMPAC network transmit the data unencrypted (source: the author, using the analysis tool Wireshark)

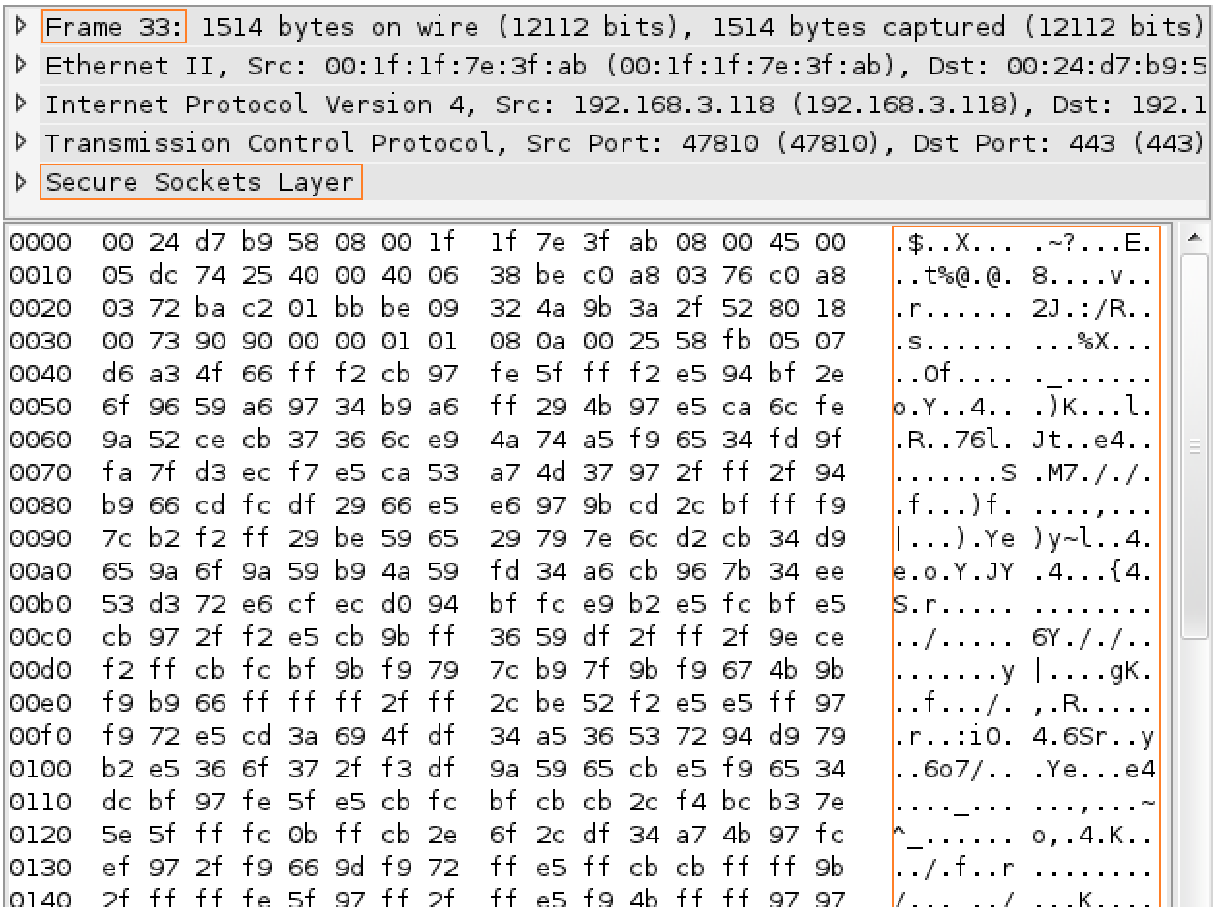

Figura 36. The detailed content of an encrypted packet sent by the SMU towards the stream receiver (source: the author, using the analysis tool Wireshark)

As result of the economic evaluation performed by the author in section 3.3 of the thesis, the amortisment time span of the initial investment in developing the SMU prototype is n=2 years, which represents a grat advantage from economic point of view, because having lower costs the SMU devices can be deployed at a much higher scale, thus improving the observability of the monitored power grid.

On the other hand the internal profitability rate indicates the upper limit of the interest rate to with it can grow for the SMU development project is still profitable. The result after the author's calculations is of 18% per year (figura 37).

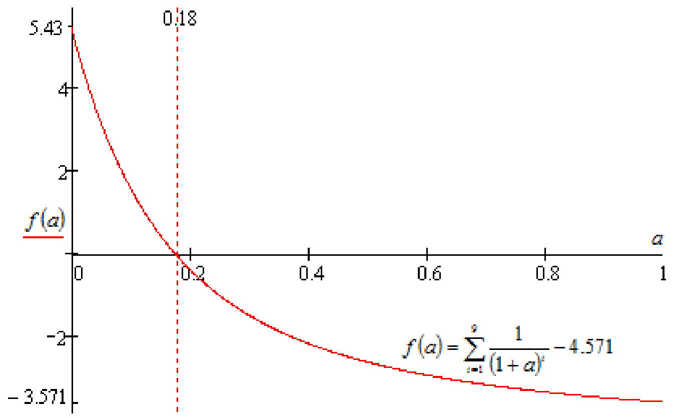

Figura 37. Maximum interest rate for which the SMU prototype project is still profitable (source: the author)

Chapter 4

Conclusions, Personal Contributions, New Research Directions

The WAMPAC and WAMS networks have had recently an accelerated development thanks to the technical progress from all the other adjacent technologies (such as: computer science, storage technologies and capacities, telecommunications, cryptography and lower costs due to more efficient production lines), which lead to the possibility of installing a continuously large number of PMU's inkey nodes of the power grids, with the final purpose of inproving the observability of the macro-scale phenomena happening at a global scale within th power system.

One might keep asking asking whether the increasing the number of deployed PMU's is the only solution maximizing the observability, especially given the new requirements that the larger the number of communication nodes (PMU's and PDC's) is, the higher is the complexity of the data network and the higher is the risk of being exposed to cyber attacks against the stability of the WAMS and WAMPAC network (and thus power grid)

4.1. Final Conclusions

Within the current thesis, the author has studied and analysed the limitations of the actual model of data acquisition (of the electric parameters) from the nodes of the power grid (sections 2.2 and 2.3), of the current data storage model of the current devices responsible with data collection (section 2.5.1), as well as the current model of transmitting the data in real-time (section 2.6.2), by means of tests and detailed simulations on a micro-WAMS network (section 3.1.).

Upon the performed tests, the author has compiled a nucleus of critical system requirements which eavery PMU or SMU equipment should fulfil (section 2.6.1), starting from multidisciplinary concepts, proven over time by other technologies, experience gathered from other research areas already existing inovations from other adjacent fields, such as telecommunications, cryptography/security and computer science (section 1.2).

In this thesis I have designed and built a prototype of the SMU equipment, whose name (*Synchronized Measurement Unit*) aims to reveal a slight similarity with the name of PMU (*Phasor Measurement Unit*) currently used widley in WAMS and WAMPAC networks and at the same time highlights the fact that the scope of such a SMU device is to record, store and transmit in real-time in a synchronized manner the high resolution power grid signal waveforms, in contrast to the actual PMU devices which only perform a local computation of the instantaneous frequency and phase angle, build the synchrophasors and delivers them to the PDC's. The design and the development of this prototype have imposed the resolution of the some technological challenges, such as:

a. The necessity to integrate a compression/decompression algorithm, because the high data volume caused by the high resolution recording seemed to make the aim difficult to achieve without such algorithms
   and
b. The necessity to interconnect some additional hardware into the SMU hardware platform such as the micro-UPS in order to offer authonomy to the SMU prototype (similar to a laptop battery), with the purpose of ensuring the recording of the power signals event before, during and after a power disturbance of outage.
   The reason for this solution is the fact that such a device could not guarantee recording during a disturbance, if the device itself is affected or even shut down by the outage.

The SMU prototype developed by the author is different than a classic data-logger equipment because the waveforms recorded by the SMU prototype from all three phases are also time stampted with a precise time GPS reference like a PMU does.

The usage of such a SMU prototype consists in the fact that the WAMPAC system cat take better proactive or corrective measures for some disturbances, by knowing additional aspects of the waveform before during and after the disturbance occurred, not only the frequency, phase angle and synchrophasors, increasing thus significantly the observability of the WAMPAC network.

4.2. Personal Contributions

As a whole, the thesis consists in a tight integration of both study elements on last generation of solutions and implementations in PMU and WAMPAC areas, performed by the author through intense research and detailed tests, as well as design, prototyping, hardware and software development elements performed entirely by the author and described in detail within this thesis.

The first sections of the thesis synthetisize concepts and technologies relevant from other adjacent research areas, due to the necessities of integrating them in to the SMU prototype. Here the author carefully selected essential aspects for the proposed solution, especially from the power engineering area, telecommunicationsm, data security, and computer science. Given these aspects, the current thesis represents a multi-disciplinary paper in the power engineering area.

In the next section of the chapter 2, I have presented the current status of the modern solutions, methods and models of implementing them in synchronized monitoring areas of smart grids, identifying their current limitations which represented the basis of the objective of this thesis.

I have created integrally a conceptual and technologic solution and I have developed a SMU prototype for implementing, testing and validating it. This prototype complements the set of measurements data offered by the PMU devices, by recording and transmitting in real-time a high definition waveform of the power grid-specific voltage and current signals.

Finally I have designed and built complex test for validation and monitoring the behaviour of the SMU prototype, within an original pilot, structured on two layers: a reference one (where the behaviour of PMU and PDC equipment behaviour has been tested within a test WMAS environment) and a second one, for the actual testing of the data acquisition, storage and transmission of the data in real-time from the SMU prototype towards an experimental data receiver achieved using a regular computer inside a virtual machine, on top of which I have developed a small testing software for comparing the received data against the original one.

The current status o fthe SMU prototype developed throughout my PhD researches, can be described by the following facts:

a. The temporary data is stored on the non-volatile memory support in one single file, whereas the logic model of the SMU prototype specifies that the temporary files are distributed in a sequence of files of equal size with the purpose of better manage their integrity in case of an outage of the memory drive;
b. The compression is done successfully by the FLAC component and the wavform does not suffer any data as result of a compression/decompression cycle; in addition the wavform data is streamed in realtime as result of its smaller size achieved through compression;
c. The waveforms transmitted by the SMU can be currently visualized only using a network inspection tool like the Wireshark, because the experimental waveform stream receiver running on top of the virtual machine does not have any graphical interface to show the received waveforms in realtime; the confirmation of the successfully received waveforms is only the comparison of the file.

d. The data security of the transmitted data is done, according to the specifications of the SMU's logical model, using asymmetric encryption keys, however their complexity is reduced at the moment;
e. Because of the current storage model of the PMU data inside the PDC's database, the transmitted waveform data sent by the SMU and received by the stream receiver are permanently stored in a completely separate database from the database of the PDC, because the C37.118 standard does not specify any database fields inside the scema to accommodate SMU's waveforms in binary form.
f. In the actual form, the SMU prototype acquired the waveforms fom its ADC analogue inputs offered by a signal generator which simulates the power grid-specific voltage waveform including any realistic distortions, harmonics, sag and swell; the SMU does not contain yet the hardware pieces needed to interface the analogue inputs of the ADC to the actual high voltage/currents of the grid and ensure the surge protection.

By designing, developing and prototypng the SMU concept during my research work, I have shown that a significantly large amount of power grid specific waveform data acquired in real-time can be transmitted as well in real-time over great distances in order for them to be correlated based on their GPS referenced timestamps, similarily to a dataset provided by the current classic PMU's.

This way I have proposed a new concept of synchronized measurement through which a centralization of the processing and decision unit is achieved, leaving only the acquisition elements (that is the SMU devices) in a distributed form on-field in various key measurement nodes of the power grid.

4.3. New Research Directions

The new research directions opened by this thesis have a common starting point – the current state if the SMU prototype implementation, presented in the previous section.

Therefore, for each of the previously enumerated points, the author proposes the following further research and development directions, not only for the SMU prototype itself, but also for the general concept of distributes synchronized measurement:

a. For the data storage in sequence of files with fixed size (with the purpose of better manage the integrity of the data in case of an memory drive outage), as described at point a de la pagina 44, I propose further researches in the new direction of developing a new software module of interfacing the SMU's software with the ext4 filesystem in Linux Embedded;

b. For the further research starting from point b at page 44, I propose new researches regarding the optimization of the FLAC compression/decompression algorithm by using parameters specific to the nature of the power grid signals;
c. For visualizing the received stream in real-time at the far end of the experimental receiver (according to point c from page 44), it is necessary to open a new research direction towards the developing of a graphical interface to be able to show in real-time the waveforms received from the SMU's; This new research direction is very complex one, because is involves not only developing graphical user interfaces, but also the whole mechanism of decompressing the received stream and time seeking the stored data based on the users search input;
d. Tightly related to the previous point, but also with point e from page 45, my fourth proposal regarding the new research directions consists in developing an algorithm of integrating the measurements provided by the classic PMU's and stored in the PDC's database, with the waveform sequences collected in real-time from the SMU. This Algorithm would require a little processing of the received stream just in order to identify the time staps needed to cross-reference with the PMU data and then perform the actual storage into a separate data bse, once the references are done.
e. For improving the encryption of the waveform send by the SMU (as described at point d from page 45), I propose a new research direction with the aim of keeping the SMU's encrypted communication up-to-date with the latest de-facto encryption algorithms, using key-pairs of 2048 or 4096 bits and less predictable random number generators, integrated with hardware based noise generators.
f. One of the most important proposals in terms of research directions is in a tight relation with point f from page 45. It refers to the fact that the SMU's ADC analogue inputs need to be directly connected to the power grid high voltage signals using some power electronic modules to prevent surge and other damage to the ADC inputs. For example one could use measurement transformers or amplifiers with separation in order to achieve the desired protection, however this elements might introduce delays or deformations of the waveform, which also need to be compensated via software based on their characteristics.

Chapter 5

Reference Summary

Author's publications

[1] Albu, M., Kyriakides, E., Chicco, G., Popa, M., Nechifor, A., "Online Monitoring of the Power Transfer in a DC Test Grid" in *Ieee Transactions on Instrumentation and Measurement* vol. 59, 2010, pp. 1104-1118, **Indexată IEEE Xplore și ISI Web of Knowledge**

[2] Albu, M., Diaz, J., Thong, V., Neurohr, R., Federenciuc, D., Popa, M., *et al.*, "Measurement and remote monitoring for virtual synchronous generator design" in *Applied Measurements For Power Systems (AMPS), 2010 IEEE International Workshop on*, 2010, pp. 7-11, **Indexată IEEE Xplore**

[3] Albu, M., Boicea, A. V., Calin, M., Popa, M., "Emerging smart grid topics in electrical engineering education" in *PowerTech, 2011 IEEE Trondheim*, 2011, pp. 1-6, **Indexată IEEE Xplore**

[4] Nechifor, A., Marin, M., Nicolaou, G., Popa, M., "Online measurements and control of a laboratory scale DC grid" in *Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE*, 2009, pp. 1701-1706, **Indexată IEEE Xplore și ISI Web of Knowledge**

[5] Popa, M., Albu, M., "Implementation overview of PMU functionalities on a regular computer" in *Smart Measurements for Future Grids (SMFG), 2011 IEEE International Conference on*, 2011, pp. 40-44, **Indexată IEEE Xplore**

[6] Popa, M., Calin, M., Ciornei, S., Dumitrescu, A. M., "Flexible and Mobile Deployment of Synchronized Measurement Systems for Active Distribution Grids" in *Applied Measurements for Power Systems, 2014 IEEE International Workshop on*, 2014, **Nepublicată, confirmată spre indexare IEEE Xplore și ISI Web of Knowledge**

Cited Sources

[7] Almas, M. S., Vanfretti, L., "Real-Time Hardware-in-the-Loop Validation for WAMPAC: Power System Protection and Communication" in *IEEE PES Innovative Smart Grid Technologies, Panel Session*, 2012.

[8] Bank, J., Kroposki, B., "Development of a real-time, high-speed distribution level data acquisition system" in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, 2012, pp. 1-6.

[9] Benjin, W., Lei, M., Huiying, D., Jiangtao, L., Zeguang, Z., Hongyi, L., "Design of dynamic synchronous multi-channel data acquisition system for lung sound based on FPGA" in *Information and Automation (ICIA), 2012 International Conference on*, 2012, pp. 768-771.

[10] Chakrabarti, S., Eliades, D., Kyriakides, E., Albu, M., "Measurement Uncertainty Considerations in Optimal Sensor Deployment for State Estimation" in *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, 2007, pp. 1-6.

[11] Chakrabarti, S., Kyriakides, E., Eliades, D. G., "Placement of Synchronized Measurements for Power System Observability" in *IEEE Transactions on Power Delivery* vol. 24, 2009, pp. 12-19.

[12] Daming, L., Yong, W., XiuXia, T., Jun, Y., "Research of High-speed Data Acquisition Card Driver Based on Embedded System for ICRH System on EAST" in *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on* vol. 3, 2010, pp. 409-413.

[13] Electric Power Research Institute (ESPRI), Kezunovic, M., Popovic, T., "Wide Area Monitoring, Protection, and Control Systems (WAMPAC) - Standards for Cyber Security Requirements", National Electric Sector Cybersecurity Organization Resource (NESCOR), Ed., 2012.

[14] Fan, Dawei, "Synchronized Measurements And Applications During Power System Dynamics - PhD Thesis", V. Centeno, Ed. Blacksburg: Virginia Polytechnic Institute - State University, 2008.

[15] IEEE, "Standard for Synchrophasor Data Transfer for Power Systems" in *Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, 2011, pp. 1-53.

[16] Kawano, F., Beaumont, P., Ishibashi, A., Hamamatsu, K., Tada, Y., Serizawa, Y., "Development of prototype wide-area monitoring, protection and control (WAMPAC) systems based upon international standards" in *PowerTech (POWERTECH), 2013 IEEE Grenoble*, 2013, pp. 1-6.

[17] Martinez, C., Eto, J., Dyer, J., "Real Time Performance Management Tools for Wide Area Operations in Competitive Markets" in *Control Center Issues session, IEEE General Meeting* vol. 2, 2003, pp. 1042-1044.

[18] Morris, T. H., Shengyi, Pan, Adhikari, U., "Cyber security recommendations for wide area monitoring, protection, and control systems" in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1-6.

[19] Novosel, D., Vu, K., Centeno, V., Skok, S., Begovic, M., "Benefits of Synchronized-Measurement Technology for Power-Grid Applications" in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 2007, pp. 118-118.

[20] Ogiboski, L., Bravo Pariente, C. A., Magrini, L. C., Jardini, J. A., "Real-Time System for Remote Data Acquisition in Power Station Equipments" in *Latin America Transactions, IEEE (Revista IEEE America Latina)* vol. 11, 2013, pp. 865-869.

[21] Thomas, J., Rajasekaran, C., "Real time data acquisition system with self adaptive sampling rate using GPU" in *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*, 2013, pp. 131-135.

[22] Vanfretti, L., Chenine, M., Almas, M. S., Leelaruji, R., Angquist, L., Nordstrom, L., "SmarTS Lab - A laboratory for developing applications for WAMPAC Systems" in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1-8.

[23] Wang, L., Chen, C., Ma, L., Shen, J., "Development of Multi-functional Virtual Oscilloscope Based on Soundcard" in *Measuring Technology and Mechatronics Automation (ICMTMA), 2011 Third International Conference on* vol. 1, 2011, pp. 1056-1059.

[24] Wen, L.-J., "The analysis of uncertainty of frequency measurement result of Quartz Crystal Oscillator" in *Gansu Science and Technology* vol. 22, 2006, pp. 1-2.

[25] Xu, X. S., Wang, G. Q., Wang, Y. l., Zhang, H. J., Li, X. H., "Study on precise frequency measurement based on sound card" in *Electronic Measurement & Instruments, 2009. ICEMI '09. 9th International Conference on*, 2009, pp. 2-455-2-458.

[26] Ye, Fan, "FPGA-based data acquisition system" in *Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on*, 2011, pp. 1-3.

[27] Ye, Z., Zhi-Zhong, X., "Error test and analysis of sound card sampling rate" in *Metallurgical Industry Automation Supplement*, 2007, pp. 1-3.

Web

[28] Boulton, R. J., Walthinsen, E., Baker, S., Johnson, L., Bultje, R. S., Kost, S., *et al.* (2013, mar.2014). *GStreamer Plugin Writer's Guide*. URL: http://gstreamer.freedesktop.org/data/doc/gstreamer/head/pwg/pwg.pdf

[29] Cai, J. Y., Zhenyu, Huang, Hauer, J., Martin, K. (2005). *Current Status and Experience of WAMS Implementation in North America*.

[30] Etek Navigation Inc. (2006, iun.2014). *MTK NMEA checksum calculator*. URL: http://www.hhhh.org/wiml/proj/nmeaxor.html

[31] Falk, H. (2012, ian.2014). *IEC 61850-90-5 - an Overview*. URL: http://www.pacw.org/issue/december_2012_issue/iec_61850905_an_overview/iec_61850905_an_overview/complete_article/1.html

[32] National Instruments. (2014, ian.2014). *CompactRIO Adviros - Build your own cRIO system*. URL: http://ohm.ni.com/advisors/crio/pages/common/intro.xhtml

[33] OpenWrt. (2014, mar.2014). *OpenWrt 12.09 Image Builder*. URL: http://downloads.openwrt.org/12.09/x86/generic/OpenWrt-ImageBuilder-x86_generic-for-linux-ix86.tar.bz2

[34] Parashar, M., Dyer, J., Bilke, T. (2005, dec.2013). *EIPP Real-Time Dynamics Monitoring System*. URL: http://www.phasor-rtdms.com/downloads/research/EIPPRTDMS_HICSS-Paper010506%20.pdf

[35] Quectel Wireless Solutions Co Ltd. (2012, feb.2014). *L10 GPS Module Specifications*. URL: http://www.quectel.com/UploadFile/Product/Quectel_L10_GPS_Specification_V1.0.pdf

[36] UfE - Umweltfreundliche Energieanlagen GmbH. (2009, dec.2013). *Grid recorder system description*. URL: http://www.ufegmbh.de/pool/download/datenblaetter/UFEGS/UfEGS_description_of_the_system.pdf