Degree Project
Master of Science in Energy Engineering

Design, Implementation and Validation of an IEC 61850-90-5 Gateway for IEEE C37.118.2 Synchrophasor Data Transfer

Author: Seyed Reza Firouzi

Director: Dr. Pedro Rodriguez Cortes
UPC, Universitat Politècnica de Catalunya
Barcelona, Spain

Supervisors Dr. Luigi Vanfretti
KTH Royal Institute of Technology
Stockholm, Sweden

Albert Ruiz-Alvarez
IREC, Institut de Recerca en Energia de Catalunya, Barcelona, Spain

Call: December 2015

ETSEIB

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

UPC

In the name of Allah

Dedicated to my parents
and
Dedicated to my wife Farzaneh and daughter Nadia

يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ

خداوند آنها را که ایمان آورده‌اند و کسانی را که از علم بهره دارند درجات عظیمی می‌بخشد.

﴿سوره: المجادلة(۵۸)، آیه ۱۱﴾

Allah will raise up in ranks,
those who believed among you and those who have been given knowledge

﴿Quran, Surah: Al-Mujadila(58), Verse:11﴾

Abstract

With the rapid growth of intermittent renewable energy sources of electricity throughout the whole power system including distribution grids and the fact that loads cannot be regarded as passive, there is an increasing need to develop new methods and technology to monitor and control the power grid in real-time. This is mainly because electric power networks are becoming more dynamic.

In this regard, there has been increasing interest in the development and implementation of Wide-Area Monitoring, Protection And Control (WAMPAC) Systems, and more importantly synchrophasor measurements that provide coherent real-time data can be used to enhance power system operation.

Synchrophasor measurement technology measures and transfers the Voltage and Current Phasors (amplitude and angle), Frequency and Rate-Of-Change-Of-Frequency (ROCOF) and other user-defined Analog and Digital state data. These measurements are synchronized to a common time reference

IEEE C37.118.2 standard which has been the dominant protocol for synchrophasor measurements systems, defines a method for real-time exchange of synchronized phasor measurement data between power system equipment.

IEC TR 61850-90-5 is a protocol for transfer of digital states and time synchronized phasor measurements over wide-area networks enabling implementation of WAMPAC systems in the context of the worldwide accepted IEC 61850 standard.

This work describes the design, implementation and validation of a library to receive and parse synchrophasor data from PMUs/PDCs based on IEEE C37.118 protocol, map it to the IEC 61850 data model and transmit the PMU data through either Routed-Sampled Value (R-SV) or Routed-GOOSE (R-GOOSE) services defined in the IEC 61850-90-5 protocol.

In addition to the gateway described above, another library developed to receive and parse R-SV or R-GOOSE streams. By parsing the received IEC 61850-90-5 messages, the raw synchrophasor data are extracted and fed into the applications using PMU data.

The functionality of the developed library is tested and validated in a Real-Time Hardware-in-the-Loop (HIL) simulation environment to assess its conformance to the functional requirements in the IEC 61850-90-5 standard.

Acknowledgments

First and foremost, I humbly thank Allah Almighty for the strength he has given me throughout my life.

I would like to thank people at KTH SmarTS Lab at which I succeeded to conduct this master thesis. My deepest gratitude goes to my supervisor, Professor Luigi Vanfretti, for providing me the valuable opportunity of doing this master project and for giving me clear perspective. I am particularly grateful to Farhan Mahmood and Dr. Hossein Hooshyar for their worthy contribution, supervision and support to my work.

I would also like to thank people at IREC research center in Barcelona. Many thanks go to Albert Ruiz-Alvarez for all his guidance and valuable suggestions. I thank Ignasi Cairo for his supports.

I would like to give my biggest thanks to my parents. Without their supports I would not have been able to reach to this stage of my life.

Finally and most importantly, I would like to express the most special gratitude to my wife Farzaneh and my daughter Nadia. Without your unconditional patience, support and love, this could have been impossible.

Contents

List of Figures

List of Tables

Nomenclature

ACSI	Abstract Communication Service Interface
APDU	Application Protocol Data Unit
API	Application Programming Interface
ASDU	Application Service Data Unit
ASN	Abstract Syntax Notation
BER	Basic Encoding Rule
CDC	Common Data Class
CFG	Configuration
GOOSE	Generic Object Oriented Substation Event
GPS	Global Positioning System
GSSE	Generic Substation State Event
HIL	Hardware-in-the-Loop
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISO	International Organization for Standardization
LAN	Local Area Network
LD	Logical Device
LN	Logical Node
LSB	Least Significant Bit
MMS	Manufacturing Message Specification
MSB	Most Significant Bit

OSI	Open Systems Interconnection
PI	Parameter Identifier
PDC	Phasor Data Concentrator
PDU	Protocol Data Unit
PMU	Phasor Measurement Unit
PV	Parameter Value
R-GOOSE	Routed-GOOSE
R-SV	Routed-Sampled Value
ROCOF	Rate Of Change Of Frequency
RMS	Root Mean Square
RT	Real-Time
SCADA	Supervisory Control And Data Acquisition
SCL	Substation Configuration Language
SCSM	Specific Communication Service Mapping
SI	Session Identifier
SPDU	Session Protocol Data Unit
SV	Sampled Value
TCP	Transport Control Protocol
T-L-V	Tag-Length-Value
TR	Technical Report
TSDU	Transport Service Data Unit
UDP	User Datagram Protocol
UTC	Coordinated Universal Time
WAN	Wide-Area Network
WAMPAC	Wide-Area Measurement, Protection And Control
WACS	Wide-Area Control System
WAMS	Wide-Area Measurement System
XML	Extensible Markup Language

1 Introduction

The concept of a phasor synchronized with the power system was introduced in the 1980s, [1] and standardized for the first time in 1995 with the standard IEEE 1344.

During the post mortem analysis of the Northeast blackout in 2003, it was understood that widespread synchrophasor measurements could help in preventing such great disturbances in the grid [2]. Then, in 2005 standard IEEE C37.118 with the title of "IEEE Standard for Synchrophasors for Power Systems" was approved and published.

In 2009, IEEE proposed dual logo standard and requested IEC to accept IEEE C37.118 as a part of IEC standard. However, IEC rejected the requests since the already available IEC 61850-9-2 standard can provide similar streaming functionalities. In consequence, a joint task force was formed between IEEE and IEC. The objective of this task force was to determine how to progress C37.118 in the context of IEC 61850.

In 2011, as the result of the joint work between IEEE and IEC, IEEE standard C37.117-2005 split into two parts. IEEE C37.118.1: that standardized how to measure synchrophasors and IEEE C37.118.2: that specified the data transfer requirements. The formation of this task force was the formal start of the development of IEC TR 61850-90-5 that was finally published in 2012.

IEC 61850-90-5, which is a technical report prepared by IEC technical committee 57-Power systems management and associated information exchange-, provides a way of exchanging synchrophasor data between PMUs, PDCs, WAMPAC (Wide Area Monitoring, Protection and Control), and control center applications [3] in the context of IEC 61850.

1.1 Motivation

Some major events such as blackout in the Northeastern part of the United States in 2003 and cyber security requirements of power system network (not addressed in IEEE C37.118), triggered the preparation of the IEC TR 61850-90-5. In addition, harmonization with the widely accepted concepts of IEC 61850 is the main achievement of 90-5 [4].

The advanced Internet Protocol (IP) multicast and IP subscription to provide a reliable communication infrastructure is a technology, specified in 90-5, which results in the minimization of the number of data concentrators required to distribute synchrophasor data over wide area networks [5].

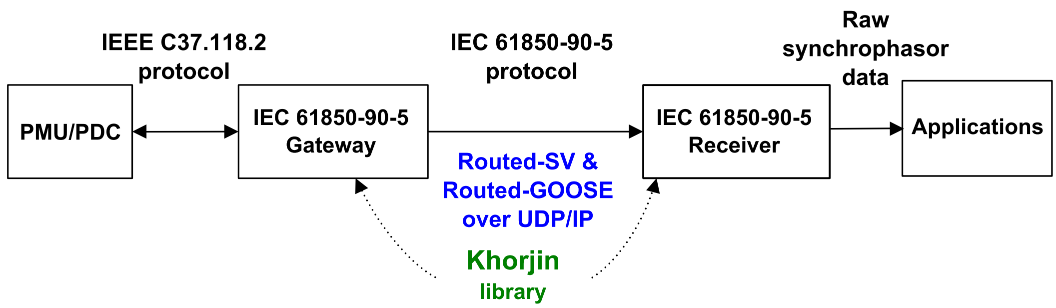


Figure 1.1: Khorjin Library functional diagram

IEC 61850-90-5 aims to create IP based routable profiles for GOOSE and Sampled Value services. The routable profile combined with multicast technologies allows a scalable use of GOOSE and Sampled Value over Wide Area Networks (WAN) and allows new and extended use cases of protection schemes in Distribution Automation (DA), and also for the integration of DER (Distributed Energy Resources) and DG (Distributed Generation) [6].

In addition to the general objectives of the introduction of part 90-5, the main goal of this work is to develop IEEE-IEC gateway capable of running in an embedded system with minimum hardware requirements (no operating system, minimum memory,...), enabling fast cyclic transfer of synchrophasor streams over wide-area networks, thereby minimizing latencies in real-time applications.

1.2 Scope of work

In this work, a library is designed and implemented to (1) communicate with PMU/PDC using the d C37.118 protocol, (2) map synchrophasor data to the IEC 61850 data model and (3) publish the IEC 61850-90-5 messages transmitting synchrophasor using either Routed-Sampled Value (R-SV) or Routed-GOOSE (R-GOOSE) services defined in [3].

In addition, another part in the library fulfills the requirements of receiving and parsing the R-SV and R-GOOSE frames and provide the raw synchrophasor data to subscribed applications.

For the sake of simplicity, the library is named "Khorjin"[1]. The functional diagram of the Khorjin library is shown in Figure 1.1.

1.3 Literature review

During recent years several documents has been published either before official release of part 90-5 [1] or after publication of part 90-5 [2], [4], that studied and reviewed the concept of synchrophasor data exchange in the context of IEC 61850.

[1] In the Persian language, "KHORJIN", is a special bag placed on the two sides of a horse, which was used for transferring of parcels.

Apart from standard review studies, some other works have been done in the field of utilization of "native" Sampled Value (SV) and GOOSE services, encapsulated directly into Ethernet frames. In [7], Lee *et al* have developed an IEC 61850-based PMU interface using IEC 61850-8-1 MMS and GOOSE services. In [8], Yong *et al* implemented synchrophasor data mapping to IEC 61850-9-2 Sampled Value service. However none of these works addressed the new mapping service of Routed-GOOSE and Routed-Sampled Value introduced in part 90-5.

While no academic literature has been found addressing the design and implementation of software to support part 90-5 of IEC 61850, Cisco, Inc and System Integration Specialists Company, Inc (SISCO) started a joint work in 2012 to provide an open source implementation framework for synchronized phasor measurement communications based on the IEC TR 61805-90-5 [5]. There has been no release from this project yet.

Another project has been announced at the Salzburg research center, named as OTITOS - Open Test Implementation for IEC 61850-90-5-based Transmission of Synchrophasor information. It is announced that the focus of the OTITOS project is to develop the test procedures for 90-5 [9].

1.4 Thesis Outline

In chapter 2 comprehensive background is provided to understand the context of the results in this report. At the beginning a short description describes the role of WAMPAC systems in Smart Grids, then the synchrophasr measurement system and relevant synchrophasor standards are introduced as they are the basis of WAMPAC systems.

In chapter 3, the IEEE C37.118.2-2011 standard is summarized and described. In chapter 4 the IEC 61850-90-5 standard and the Routed-GOOSE and Routed-SV frame structures are interpreted and explained in detail.

In chapter 5, after describing the design and implementation of the library, validation of the functionality of the library in a HIL real-time simulation environment is explained.

2 Background

2.1 Smart Grid and Synchrophasor Technology

2.1.1 Smart Grids

History

The commonly accepted concept of electric power systems developed until the past decade, divided the whole network into three main sectors of Generation, Transmission and Distribution. In this context, the bulk power generated in large scale power plants, of different types i.e. nuclear power, hydropower or fossil fueled, is fed into high voltage networks. This power is transmitted over long distances and then delivered to distribution networks, enabling the end-users of grid to be supplied with electrical energy required for their use.

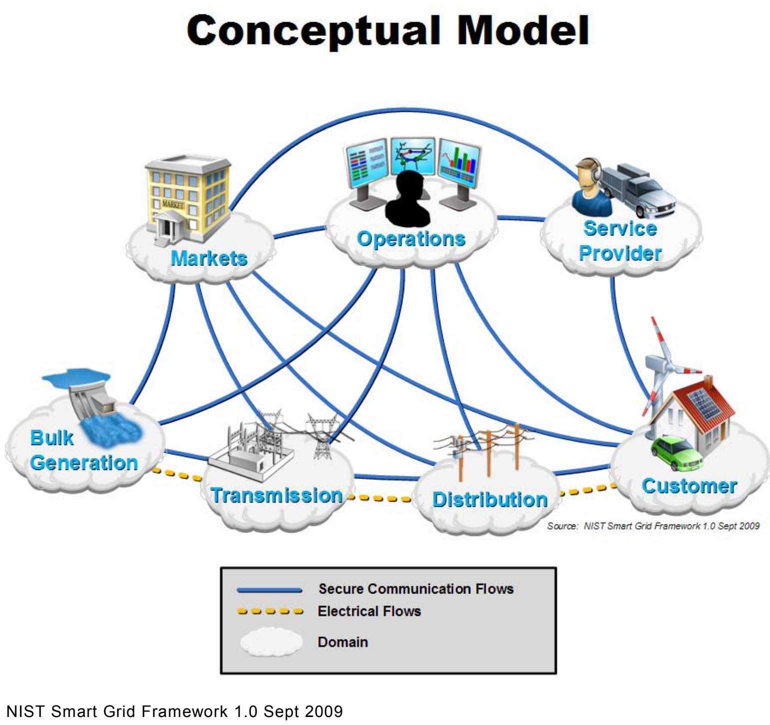
Nowadays, electricity production is not limited to the large scale power plants, as intermittent renewable energy sources of electricity (e.g., wind and solar power) are deployed throughout the whole network, including distribution grid. Furthermore, loads can no longer be regarded as the passive, as the development of Distributed Generation (DG) systems has allowed the loads to act as a generator, injecting power into the distribution network. In addition, new types of loads will be introduced to the grid (e.g., storage in the batteries of electric vehicles).

The aforementioned issues, alongside with market-driven behavior of the demands, are making electric power supply networks more and more dynamic.

The recent development of communication systems has resulted in new possibilities for monitoring and control of the power system, and in consequence, more effective, flexible and lower cost operation. The Smart Grid is an opportunity to use ICT (Information and Communication Technologies) to revolutionize the electrical power system [10].

What is the Smart Grid?

There is no single definition for the Smart Grid. And there is no single architecture for it either. For this reason, most of the literature use the conceptual model in [11] as a reference. According to this report, the Smart Grid conceptual model simply consists of **several interconnected domains** (see Figure 2.1). These domains are the following:



Source: Report to NIST on the Smart Grid Interoperability Standards Roadmap [11]

Figure 2.1: Smart Grid conceptual model

- Customers
- Markets
- Service Providers
- Operations
- Bulk Generation
- Transmission
- Distribution

Each domain contains its own actors. Actors may be devices, computer systems or software programs and/or the organizations that own them. Actors have the capability to make decisions and exchange information with other actors through interfaces.

In what ways are "Smart Grids" smarter?

Smart Grids are "smarter" than conventional power systems in two aspects. First, they provide a two-way flow of electricity and information to optimize supply and demand. While in traditional grids there is only one-directional flow of information from consumers to the grid (through meters) and one-directional flow of energy from the grid to the consumers, smart grids enable bi-directional flow of information (through a variety of interfaces) and energy (through distributed generation and storage).

Second, Smart Grids are "smarter" in the sense that they enable integration of a wide variety of energy sources and energy customer services in highly interconnected electricity systems [12].

Goal of the Smart Grid

In [13] the principle characteristics of a smart grid summarized by the US Department of Energy's Office of Electricity Delivery and Energy Reliability are described. Among them one key characteristic is mentioned: to *Anticipate and respond to system disturbances in a self-healing manner.* This characteristic provides the guiding principle for this work .

Based on this principle, it is expected that a Smart Grid would have the ability to foresee and avoid any potential disturbances. At the first stage, it would be ideal if disturbances could be avoided. If they occur, they should be contained quickly to limit their impact on the normal operation of the grid. Finally, when a disturbance occurs, it should be corrected as quickly as possible, so to bring the grid back to normal operation.

Key technologies of a Smart Grid

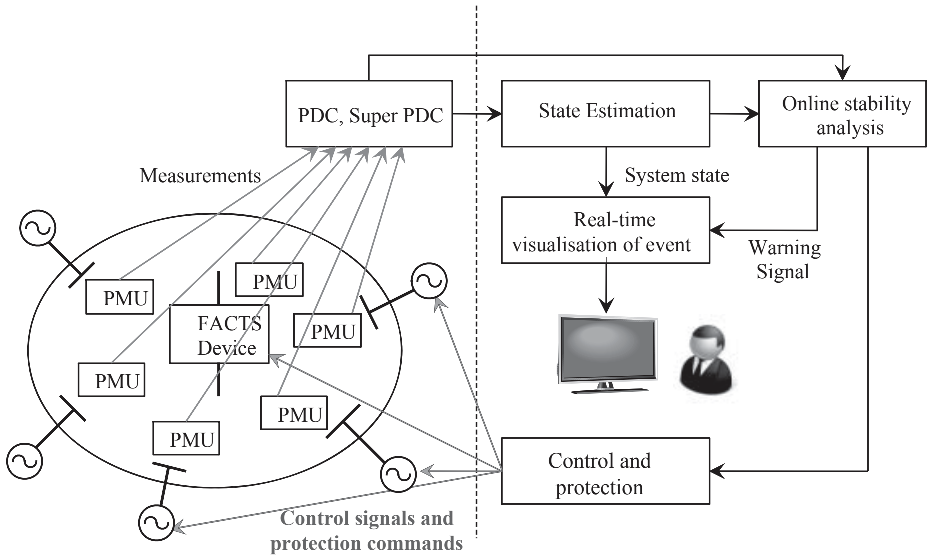
In order to fulfill the requirements of a Smart Grid, in [10] several technologies are introduced to be developed and implemented. Phasor Measurement Units (PMU) and Wide Area Monitoring, Protection and Control (WAMPAC) Systems are among the main components of "Sensing, Measurement, Control and Automation Technologies".

2.1.2 Wide Area Monitoring, Protection and Control (WAMPAC)

The management of the transmission system generally requires "Wide-Area" communications architectures and WANs. The definition of "local area", "metropolitan area", "wide area" or any of a number of other "area" communication network technologies is somehow loosely defined. However, from a power system point of view, the term "Wide-Area Monitoring (WAM)", "Wide-Area Monitoring System (WAMS)", or "Wide-Area Monitoring, Protection, and Control (WAMPAC)," or any number of other terms that begin with "wide-area" are all used to denote large-scale, global management of power systems, including the transmission system [13].

Wide-Area Measurement Systems (WAMS) are being installed on many transmission systems to supplement traditional SCADA. They provide the magnitudes and phase angle measurements of bus voltages and current flowing the transmission circuits. This information, measured over a wide area, is transmitted to the Control Center and is used for following applications [10]:

1. Power system state estimation



Source: Smart Grid Technology and Applications [10]

Figure 2.2: Simplified representation of WAMPAC

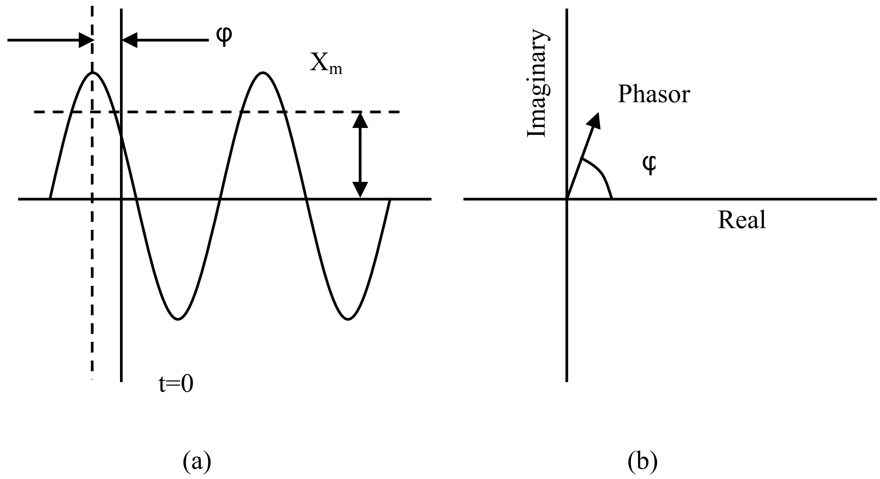
2. Power system monitoring and warning
3. Power system event analysis

"Wide-Area Control Systems(WACS)" are more complex systems which include the implementation of closed-loop wide-area controllers capable of providing reactive power compensation for mitigating power system transient stability, power system oscillations, and voltage collapse. "Wide-Area Protection Systems (WAPS)", on the other hand, involve intelligent tripping within Remedial Action Schemes (RAS) for different aims including optimized load shedding [14].

A typical configuration of a WAMPAC system is illustrated in Figure 2.2. In this architecture the synchrophasor measurements provided by PMUs throughout the network are used for online stability analysis. In case of occurrence of an event, its data including location, time, magnitude and type are first identified. Real-time monitoring of the event allows it to be replayed several seconds after it occurs. The future system condition is then analyzed using the information that has been collected. An online stability assessment algorithm continuously assesses whether the system is still stable and how fast the system would collapse if it became unstable. If instability is predicted, then the necessary corrective actions to mitigate problem or to avert system collapse are taken [10].

2.1.3 Synchronized phasor measurement system

Synchronized phasor measurements have become the main chosen measurement technique in electric power systems. Voltage and current measurements synchronized to a common time reference are provided by this technology. Availability of the Global Positioning System



Source: [15]

Figure 2.3: A sinusoid waveform (a) and its equivalent phasor representation (b)

(GPS) and development of the sampled data processing techniques, enabled this technology to emerge.

In addition to positive sequences of voltages and currents, frequency and rate of change of frequency are other available measurements furthermore, the system could be set to measure harmonics, negative and zero sequence quantities, as well as individual phase voltages and currents [15].

While positive sequence measurements enable utilities to know the state of the power system at any time, post-event analysis of recent blackouts in power systems has revealed the importance of synchrophasor measurements and PMUs. Synchrophasors are described as "the MRI of the bulk power system". [16]

Phasor Representation

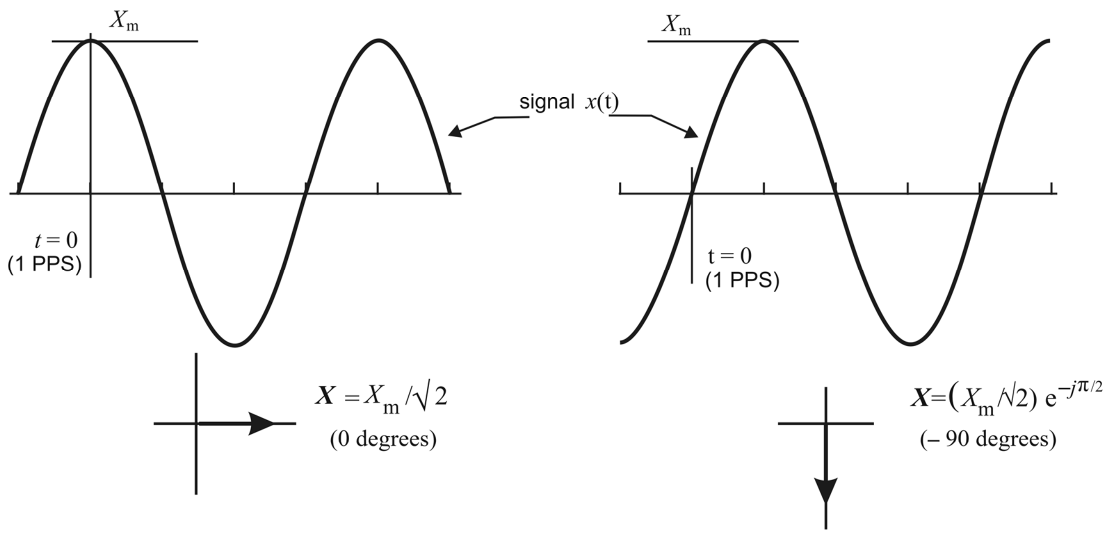
A pure sinusoidal waveform as shown in Equation 2.1 :

$$x(t) = X_m \cos(\omega t + \phi) \tag{2.1}$$

has an equivalent phasor representation as shown in Equation 2.2

$$X = (X_m/\sqrt{2})e^{j\phi} = (X_m/\sqrt{2})(cos(\phi) + jsin(\phi)) \tag{2.2}$$

Whereas ω is defined as the frequency of the signal in radians per second, and ϕ is the phase angle in radians. X_m is the peak amplitude of the signal and $(X_m/\sqrt{2})$ is the root mean square (RMS) value of the signal. A sinusoidial waveform and its equivalent phasor representation are depicted in Figure 2.3 [15].



Source: IEEE Std C37.118.1-2011 [17]

Figure 2.4: Convention for synchrophasor representation

Synchrophasor definition

Synchronized phasor or Synchrophasor is a phasor representation relative to a sinusoid at the nominal system frequency synchronized to UTC time. In section 4.2 of [17] following definition is presented:

"The synchrophasor representation of the signal $x(t)$ in Equation 2.1 is the value X in Equation 2.2 where ϕ is the instantaneous phase angle relative to a cosine function at the nominal system frequency synchronized to UTC."

Under this definition, ϕ is the offset from a cosine function at the nominal system frequency synchronized to UTC. A cosine has a maximum at $t = 0$, so the synchrophasor angle is 0 degrees when the maximum of $x(t)$ occurs at the UTC second rollover (1 PPS time signal), and –90 degrees when the positive zero crossing occurs at the UTC second rollover (sine waveform). Figure 2.4 illustrates the phase angle/UTC time relationship [17].

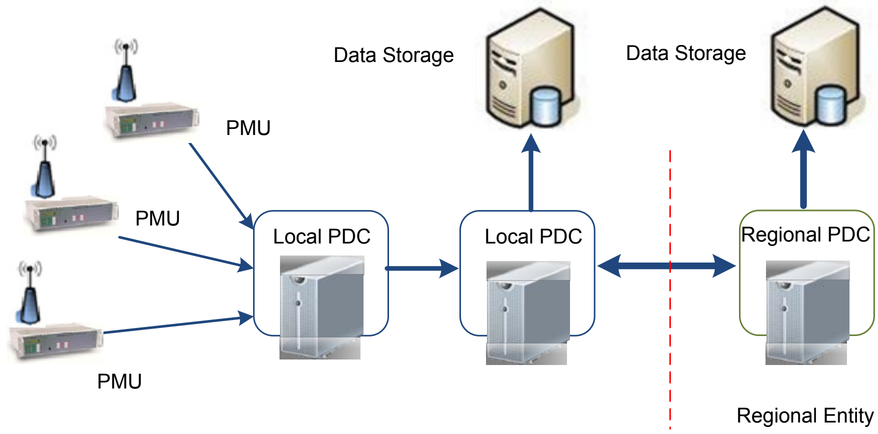
Frequency and rate of change of frequency

In addition to synchrophasor measurements, Frequency and Rate Of Change Of Frequency (ROCOF) shall be generated by a PMU. Consider a sinusoidal signal as shown in Equation 2.3:

$$x(t) = X_m \cos[\Psi(t)] \tag{2.3}$$

Frequency is defined by Equation 2.4:

$$f(t) = \frac{1}{2\pi} . \frac{d\Psi(t)}{dt} \tag{2.4}$$



Source: IEEE Std C37.118.2-2011 [18]

Figure 2.5: Synchrophasor data collection network

The ROCOF is defined as shown in Equation 2.5:

$$ROCOF(t) = \frac{df(t)}{dt} \tag{2.5}$$

Synchrophasor measurement system overview

A basic architecture of synchrophasor data transfer system consisting of PMU and Phasor Data Concentrator (PDC) is illustrated in Figure 2.5. When multiple *Intelligent Electronic Device (IED)* provide synchorphasor measurements inside the substation, a local PDC may be placed at the substation level. Typically several PMUs situated at key substations collect measurements and send them in real-time to a PDC at the utility level. In order to provide an interconnection-wide snapshot of the power grid measurements, many PDCs belonging to different utilities can feed data to a central PDC [18].

Phasor Measurement Unit

The PMU is a function within an IED or logical device that provides phasors, analogs and digital data and system frequency estimates, as well as other optional information such as calculated megawatts (MW) and megavars (MVAR), sampled measurements, and Boolean status words. The PMU may provide synchrophasor estimates from one or more voltage or current waveforms. The PMU can be realized as a stand-alone physical device or as a part of a multifunction device such as a protective relay. This information may be recorded locally or transmitted in real time to a central location as illustrated in Figure 2.5 [18].

Phasor Data Concentrator

A PDC acts as a node in a synchrophasor data transfer network where data from multiple PMUs or PDCs is correlated and fed out as a single stream to the higher level PDCs and/or applications [18]. PDCs enable to meet two objectives: the primary purpose is to provide more scalable distribution of synchrophasor measurements, since Phasor Measurement Units (PMUs) usually have limited communication capability and are not designed to distribute the measurements to a large number of clients. The other purpose is to provide a function to time-align measurements from different PMUs. [2].

2.1.4 Evolution of synchrophasor standards

The history of synchrophasor standards backs to 1995 by approval and publication of IEEE 1344 standard.This standard was in force until 2005, when the new standard of IEEE C37.118 with the title of "IEEE Standard for Synchrophasors for Power Systems" was approved and published.

One of the main stimulus for introduction of the new standard in 2005 was the Noreast blackout in 2003. During the post mortem analysis, it was understood that widespread synchrophasor measurements could enable to prevent such great disturbances in the grid [2].

In 2009, IEEE proposed dual logo standard and requests that IEC accept IEEE C37.118 as an IEC standard. However, IEC rejects the requests since the already available IEC 61850-9-2 can provide similar streaming functionality. In consequence, a joint task force was formed between IEEE and IEC. The objective of this task force was to determine how to progress C37.118 in the context of IEC 61850.

In 2011, as the result of the joint work between IEEE and IEC, IEEE standard C37.117-2005 split into two parts. IEEE C37.118.1: that standardized how to measure synchrophasors and IEEE C37.118.2: that specified the data transfer requirements. The formation of this task force was also the formal start of the development of IEC TR 61850-90-5 that was finally published in 2012.

This time-line is summarized below and also depicted in Figure2.6.

- 12-1995: IEEE publishes IEEE 1344 a synchrophasor standard.
- 08-2003: Northeast blackout.

  The importance of synchrophasor measurements became more apparent, resulting in an effort to define a new synchrophasor standard.
- 07-2005: IEEE C37.118 was published.

  It included measurement technique standardization as well as a packet format definition that could be used to convey the measured information.
- 03-2009: IEEE requests IEC for dual logo standard. IEC rejects the request, because IEC 61850-9-2 was already available with the required functionality.
- 09-2009: Formation of joint task force,between IEEE and IEC. With the purpose of determining how to progress C37.118 in the context of IEC 61850.
- 12-2011: Publication of IEEE C37.118.1 and IEEE C37.118.2
- 05-2012: Publication of IEC TR 61850-90-5

IEEE C37.118.1-2011 Standard

This standard presents synchronized phasor (synchrophasor) measurement system. The measurement terms: *Synchrophasors*, *Frequency*, and *Rate of change of frequency (ROCOF)* under all operating conditions are defined. In addition it presents the standard methods for evaluating these measurements and requirements for compliance with the standard under both steady-state and dynamic conditions. Time tag and synchronization requirements are included. It defines *Phasor Measurement Unit (PMU)* as a stand-alone physical unit or a functional unit within another physical unit. This standard does not specify hardware, software, or a method for computing phasors, frequency, or ROCOF [17].

IEEE C37.118.2-2011 Standard

This standard defines a method for real-time exchange of synchronized phasor measurement data between power system equipments. It specifies messaging protocol that can be used with any suitable communication protocol for real-time communication between phasor measurement units (PMU), phasor data concentrators (PDC), and other applications. As part of the standard in addition to different data types and formats, different types of messages, their contents and use are explained. Although the method introduced in the standard is independent of the underlying data transfer protocol, some major communication options and their requirements are described in annexes of the standard." [18].

IEC TR 61850-90-5

IEC TR 61850-90-5 provides a way of exchanging synchrophasor data between PMUs, PDCs, WAMPAC (Wide Area Monitoring, Protection, and Control), and between control center applications. The data, to the extent covered in IEEE C37.118-2005, are transported in a way that is compliant to the concepts of IEC 61850 [3].

Cyber security requirements was another issue that was not addressed in IEEE C37.118 standards and are considered in detail in IEC TR 61850-90-5. These requirements defined the need to make wide area power system communications more secure.

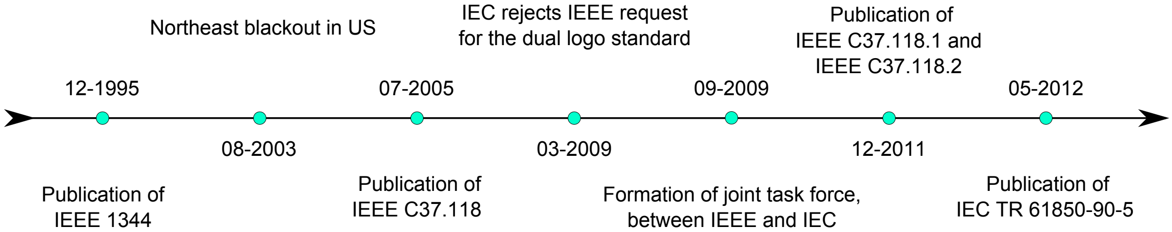


Figure 2.6: Evolution of synchrophasor standards

2.2 IEC 61850 Standard

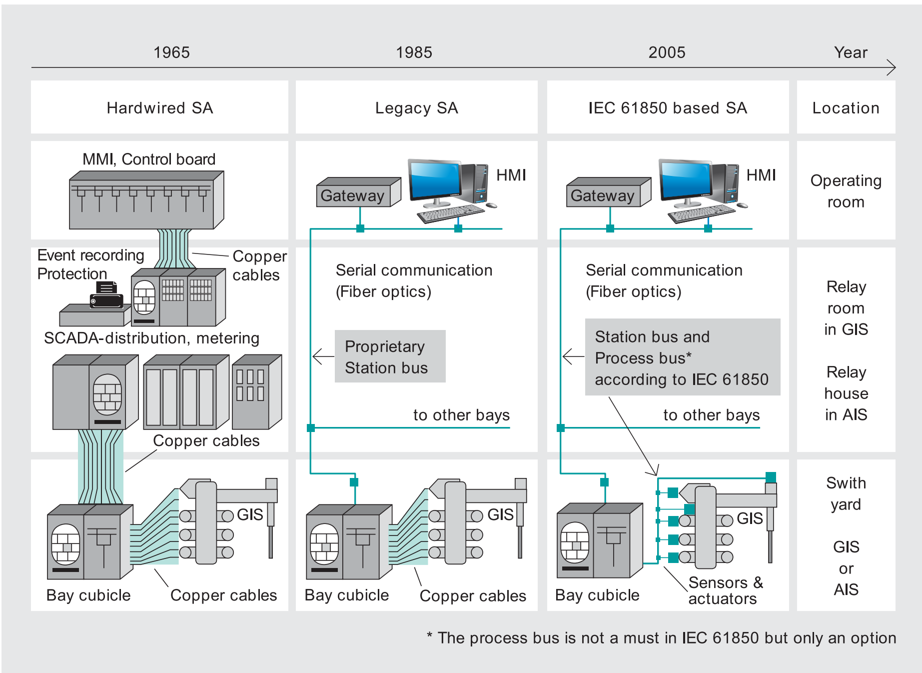
2.2.1 Introduction

IEC 61850 emerged from the wide introduction of Intelligent Electronic Devices (IED's) into power substations in the period of 1980-90s. During that period, new processor-based IEDs capable of performing multiple functions, replaced old single-function devices hardwired to other single function devices in substations. In spite of this flexibility, it quickly became impossible to keep track of what logical functions were doing what, where they were located, and what unwanted problems might result.

In the late 1990s, several operation and communication architectures which were already introduced in the 1980s -such as the Utility Communication Architecture (UCA)-were merged into IEC 61850 [1]. In Fig. 2.7 evolution of substation automation (SA) architecture from hardwires to proprietary protocols and recently to IEC 61850 is shown.

IEC 61850, as an important international standard for substation automation, is a part of the International Electrotechnical Commission's (IEC) Technical Committee 57 (TC57) architecture for electric power system [19].

As mentioned in section 4 of [20], the objective of this standard is to support interoperability of IEDs from different manufacturers. Interoperability in this case is the ability to operate on the same network or communication path sharing information and commands.



Source: [21]

Figure 2.7: Substation automation (SA) architecture from hardwires over proprietary protocols to IEC 61850

The initial scope of IEC 61850 standard was communications within the substation. However, after the introduction of IEC 61850, utility and vendor experts of non-substation related application domains began to realize both the benefits of a single international standard for the electrical energy supply system and the powerful approach and content of IEC 61850.

As the result, IEC 61850 has been extended to other domains, such as DR, Hydroelectric Power Plants and recently in part 90-5 to wide-area transmission of synchrophasor information according to IEEE C37.118. In fact, the title in the newest editions of IEC 61850 parts has been renamed "Communication networks and systems for power utility automation" (instead of "Communication networks and systems in substations" used in first editions).

2.2.2 Outline of IEC 61850

IEC 61850 defines the various aspects of the power utility communication networks and systems in various sections. The latest release versions of different parts are detailed in Table.2.1 [22].

Table 2.1: Structure of IEC 61850 standard

Reference	Title	Edition	Publication
IEC 61850-1	Communication networks and systems in substations - Part 1: Introduction and overview	1.0	2003-04-28
IEC 61850-2	Communication networks and systems in substations - Part 2: Glossary	1.0	2003-08-07
IEC 61850-3	Communication networks and systems for power utility automation - Part 3: General requirements	2.0	2013-12-12
IEC 61850-4	Communication networks and systems for power utility automation - Part 4: System and project management	2.0	2011-04-11
IEC 61850-5	Communication networks and systems for power utility automation - Part 5: Communication requirements for functions and device models	2.0	2013-01-30
IEC 61850-6	Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs	2.0	2009-12-17
IEC 61850-7	Communication networks and systems for power utility automation - Basic communication structure		
· IEC 61850-7-1	Principles and models	2.0	2011-07-15

Table 2.1: Structure of IEC 61850 standard

Reference	Title	Edition	Publication
· IEC 61850-7-2	Abstract Communication Service Interface (ACSI)	2.0	2010-08-24
· IEC 61850-7-3	Common Data Classes	2.0	2010-12-16
· IEC 61850-7-4	Compatible Logical Node classes and data classes	2.0	2010-03-31
· IEC 61850-7-410	Hydroelectric power plants - Communication for monitoring and control	2.0	2012-10-30
· IEC 61850-7-420	Distributed energy logical nodes	1.0	2009-03-10
IEC 61850-8	Communication networks and systems for power utility automation - Specific Communication Service Mapping (SCSM)		
· IEC 61850-8-1	Mapping to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3	2.0	2011-06-17
IEC 61850-9	Communication networks and systems for power utility automation - Specific Communication Service Mapping (SCSM)		
· IEC 61850-9-2[1]	Sampled Values over ISO/IEC 8802-3	2.0	2011-09-22
IEC 61850-10	Communication networks and systems for power utility automation - Part 10: Conformance testing	2.0	2012-12-14

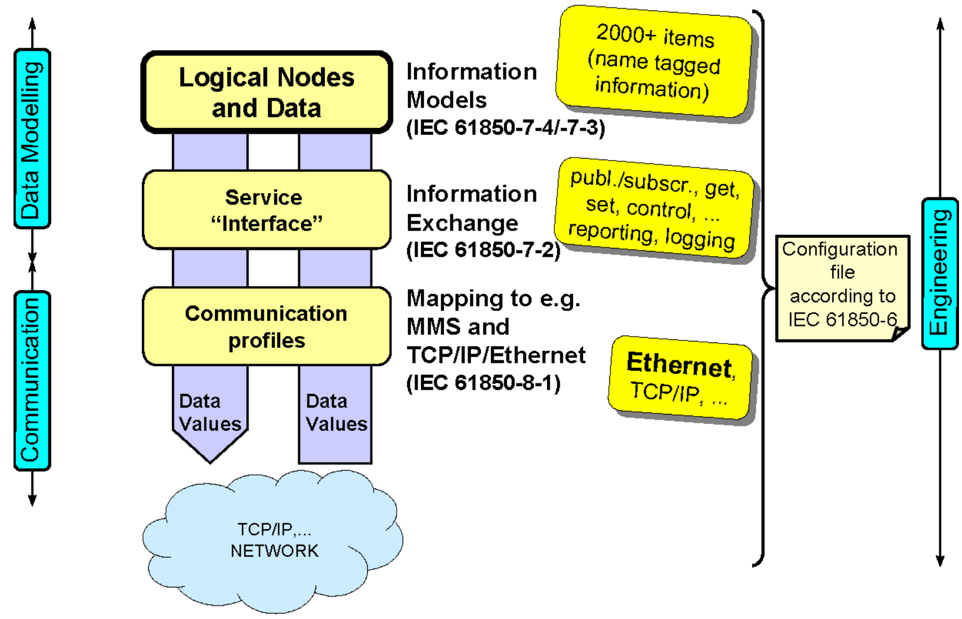
The architecture of IEC 61850 is based on "abstracting" the definition of the data items and the services. Abstract means the definition of the data and information to describe what the services provide [23]. IEC 61850 standard can be introduced from three aspects of: *Data Modeling*, *Communication* and *Engineering*.

Parts 3, 4, and 5 of the standard start by identifying the general and specific functional requirements. Part 7.2 of the standard includes the definition of the abstract of services and the abstraction of the data objects(referred to as Logical Nodes)is found in part 7.4. With the development of the implementation of the standard beyond the initial domain of substation to power utility, the complementary parts of the standard is published in parts 7-410, 7-420 and 7-510.

In this standard, any kind of data object is constructed by the defined common building blocks called "Common Data Classes" (CDC). All Common Data Classes are defined in part 7.3 of the standard. Hence, the *Data Modeling* aspect of IEC 61850 standard is defined in subparts of part 7 of standard.

The abstract definitions then allow "mapping" of the data objects and services to any other protocol that can meet the data and service requirements. Part 8.1, defines the mapping

[1]IEC 61850-9-1:2003 has been withdrawn and replaced by IEC 61850-9-2:2011 [22]



Source: IEC 61850 Part: 7-1 [24]

Figure 2.8: Overview of IEC 61850 series

of the abstract data object and services onto the Manufacturing Messaging Specification – MMS and section 9.2 defines the mapping of the Sample Measured Values onto an Ethernet data frame. The *Communication* aspect of IEC 61850 standard is expressed in these parts of the standard.

From an *Engineering* point of view, there is significant amount of configuration that is required in order to put all the pieces together and have them work. For this purpose, an XML based Substation Configuration Language (SCL) was defined in part 6 of standard [19].

The structure of all IEC 61850 components is illustrated in Figure 2.8.

Like any standard, IEC 61850 is in a constant updating process. For this purpose TC 57 uses the Technical Report (TR) procedure, through which new additional parts included in a TR goes through a review and approval process, and then becomes an official extension of the standard. Every few years, the standard itself is updated by editing the approved TRs into the body of the standard [1]. The detailed list of new parts of IEC 61850 standard which are published in form of Technical Reports (TR) is shown in detail in Table. 2.2.

Table 2.2: New additional parts of IEC 61850 standard

Reference	Title	Edition	Publication
IEC TR 61850-1	Communication networks and systems for power utility automation - Part 1: Introduction and overview	2.0	2013-03-14
IEC 61850-7	Communication networks and systems for power utility automation - Basic communication structure		
· IEC TR 61850-7-510	Hydroelectric power plants - Modeling concepts and guidelines	1.0	2012-03-22
IEC TR 61850-90-1	Communication networks and systems for power utility automation - Part 90-1: Use of IEC 61850 for the communication between substations	1.0	2013-03-16
IEC TR 61850-90-4	Communication networks and systems for power utility automation - Part 90-4: Network engineering guidelines	1.0	2013-08-06
IEC TR 61850-90-5	**Communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit synchrophasor information according IEEE C37.118**	**1.0**	**2012-05-09**
IEC TR 61850-90-7	Communication networks and systems for power utility automation - Part 90-7: Object models for power converters in distributed energy resources (DER)	1.0	2013-02-21

2.2.3 Data Modeling

IEC 61850 is not only a communication standard, but also another objective of this standard is to provide interoperability between IEDs from different vendors. These goals are achieved by the standard defining data model and data exchange based on these models [23]. The IEC 61850 data model is described below:

- *Physical Device*

  The IEC 61850 data model starts with a Physical Device, that is an Intelligent Electronic Device (IED) connected to the network and identified by its network address. Each Physical Device may contain one or more Logical Device.
  In Fig. 2.9, the circuit breaker and in Fig. 2.10, the protection relay are Physical Devices.

- *Logical Device*

  In chapter 3 of [25], Logical Device is defined as an "entity that represents a set of typical substation functions". Every Logical Device (LD) consists of a minimum of three Logical Nodes [23].

  Breaker Controller in Fig. 2.9 and PMU Function in Fig. 2.10 are examples of Logical Devices.

- *Logical Node*

  A Logical Node is a named grouping of data and associated services that is logically related to some power system function [19].In chapter 3 of [25], Logical Node is defined as an "entity that represents a typical substation function" [25].

  Circuit Breaker Logical Node of XCBR in Fig. 2.9 and Measurement Logical Node of MMXU in Fig. 2.10 are presented as examples of Logical Nodes in Logical Devices.

  In the first edition of IEC 61850 standard, 92 Logical Nodes were defined in 13 main groups. The LN name in each group is starting with a common letter as the indicator of the group the LN belongs to. For example, measuring Logical Node MMXU in the group of "Metering and Measurement" with "M" as the group indicator, or circuit breaker logical node-XCBR- in switchgear LN group with "X" as the indicator of the group.

- *Data Class*

  Each Logical Node includes one or more elements of Data. Each data element is of one of the Common Data Class (CDC) standardized in [26]. Each CDC has a standard name describing the type and structure of the data within Logical Node. Each CDC contains several individual attributes which are categorized based on their Functional Constraints (FC).

  As can be seen in Fig. 2.9 and can be found in section 5.12. of [27], the data object "Pos" (Switch Position) is of standardized simple Common Data Class type "DPC" (Controllable Double Point). Referring to section 7.5.3 of [26], it can be seen that among different data attributes defined for DPC Common Data Class, $stVal$, q and t are the mandatory data attributes with Functional Constraint type "ST" (Status).

  For another example, as it is shown in Fig. 2.10 and can be found in section 5.10.7 of [27], the MMXU Logical Node includes PhV (Phase Voltage) data attribute of standardized composite Common Data Class type "WYE" (Phase to ground related measured values of a three phase system). Referring to section 7.4.5 of [26], it can seen that "PhsA" (Phase A) with standardized simple Common Data Class type "CMV" (Complex Measured Value) is one of the data objects of WYE Common Data Class.

  In section 7.4.3 of [26], it is explained that "CMV" Common Data Class consists of $cVal$, q and t.

2.2.4 Mapping to communication protocols

The abstract data and object models of IEC 61850 enable the IEDs functioning in power systems to present data in a standardized structure. The Abstract Communication Service

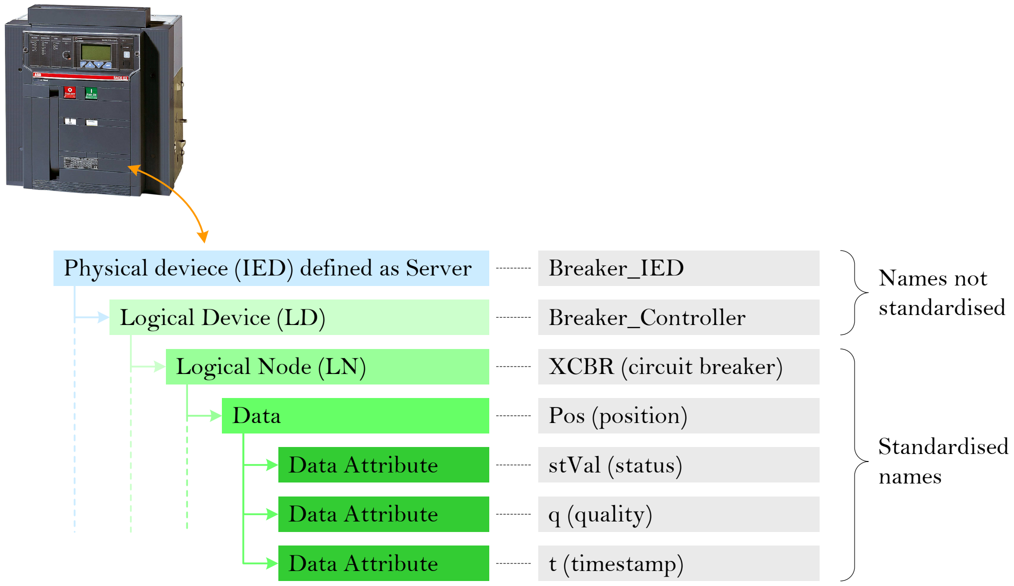


Figure 2.9: Circuit Breaker status modeled in terms of IEC 61850

.

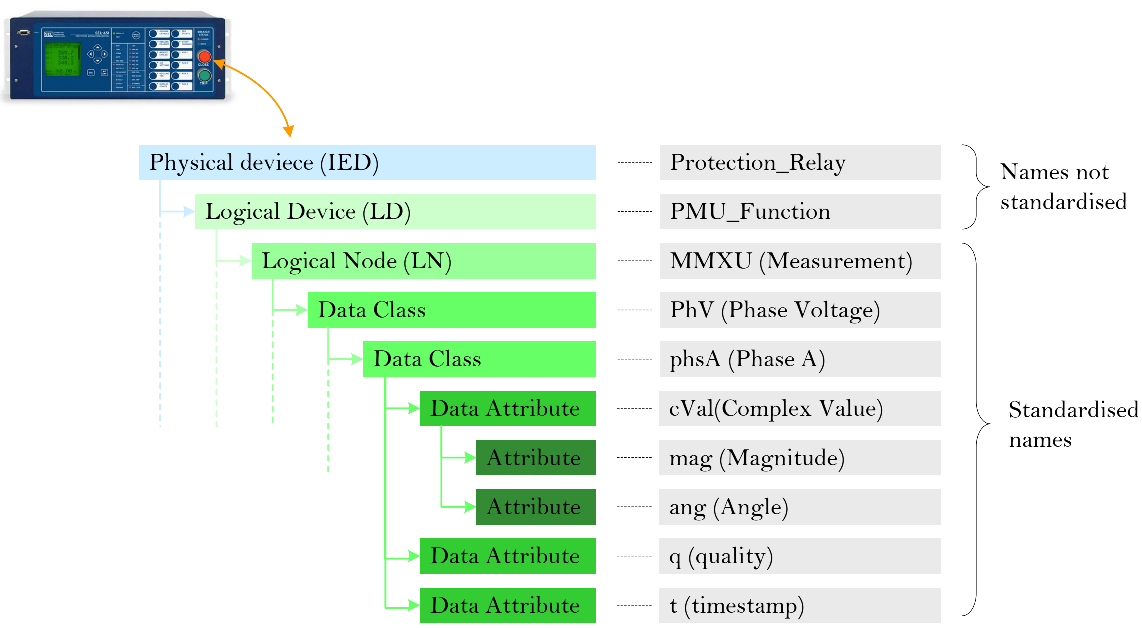


Figure 2.10: Protection Relay with PMU function modeled in terms of IEC 61850

.

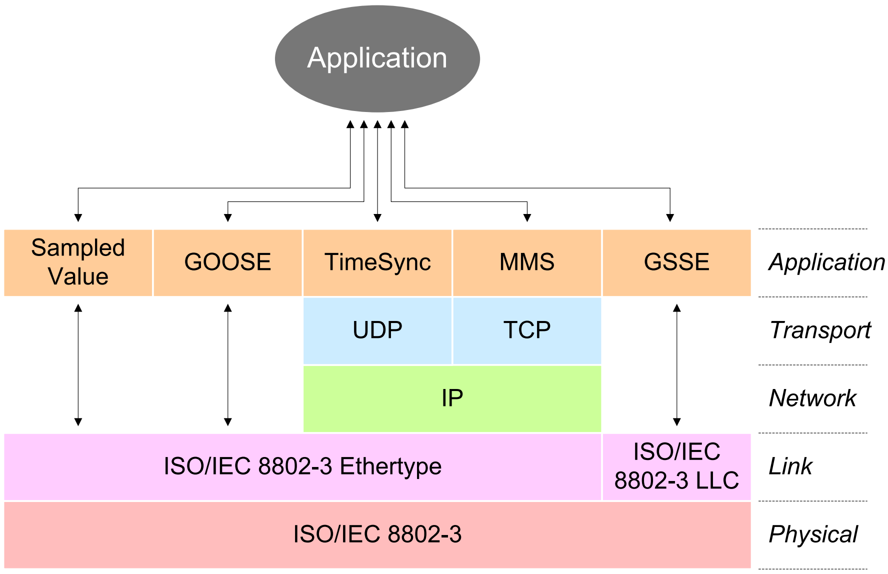


Figure 2.11: Mapping to real protocols in IEC 61850

Interface (ACSI) of IEC 61850 define a set of services and the responses to those services [19].

ACSI is a virtual interface to an IED providing abstract communication services (connection, variable access, unsolicited data transfer, device control, etc.) including the ones used for exchanging the information represented by Data and Data Attributes. The definition of ACSI is independent from any specific underlying protocol. Thus, the IEC 61850 series provide a Specific Communication Service Mapping (SCSM) to map ACSI models onto real protocols that are common in the power industry.

IEC61850	=	Abstract definition of communications	+	Mapping to real protocol
		↓		↓
		ACSI		SCSM

Parts 8-1 and 9-2 of IEC 61850 define SCSM. As shown in Figure 2.11 SCSM is a set of detailed instructions, mechanisms and rules to implement objects and services specified in ACSI through making use of real protocols.

2.2.5 Generic Object-Oriented Substation Event(GOOSE) Service

The Abstract Communications Service Interface (ACSI) that is detailed in [25], defines common utility services for substation devices and introduces the two groups of communication services:

- Client-Server model
- Peer-to-Peer model

The second model utilizes a peer-to-peer model for Generic Substation Event (GSE) services associated with time-critical activities such as fast and reliable communication between Intelligent Electronic Devices (IEDs) used for Protection purposes [28].

The GSE model is based on a publisher/subscriber mechanism and, supports the distribution of the same generic substation event information to more than one physical device through the use of multicast/broadcast services. GSE model includes two services:

- *Generic Object-Oriented Substation Event(GOOSE)* that supports the exchange of a wide range of possible common data.
- *Generic Substation State Event (GSSE)* which provides the capability to convey state change information.

GOOSE messages are exchanged between devices by means of the local Ethernet network, and enable the broadcast of multicast messages across the Local Area Network (LAN). GOOSE message is considered for time-critical events such as protection functions.

As can be seen in Figure.2.11, GOOSE message is associated with three layers of *Application layer*, *Data-link layer* and *Physical layer* defined in the Open Systems Interconnection (OSI) model.

2.2.6 Sampled Value Service

Among five communication services defined in IEC 61850 and illustrated in Figure 2.11, three fo them are time-critical and are used for protection and control applications inside substation. They include *Sampled Value*, *GOOSE* and *GSSE*.

Similar to GOOSE, the Sampled Value service is mapped directly to the Data-Link layer. Sampled Value service is defined for the fast and cyclic transmission of raw data generated by measurement equipment inside substation. For example, a Merging Unit (MU) which receives multiple analogue current and voltage samples coming from current and voltage transformers, put these measurements in context as Sampled Value(SV) packets and sends them over switched Ethernet network to Protection & Control functions [29].

3 IEEE C37.118.2-2011 standard

In this chapter IEEE C37.118.2 standard is summarized. In spite of being a summary, all the details are explained through figures and tables.

3.1 Synchrophasor measurement system

In section 2.1.3, the principle synchrophasor measurements are explained. In the following, some additional topics are introduced.

Synchrophasor measurement time tags

Each synchrophasor measurement is tagged with a UTC timestamp indicating the time of measurement. This timestamp consist of three components:

1. Second-Of-Century (SOC)
2. FRACtion-of-SECond (FRACSEC)
3. Message time quality flag

The SOC count is a four (4) byte binary count of seconds from UTC midnight (00:00:00) of January 1, 1970, to the current second. This count is represented as a 32-bit unsigned integer. In order to keep the time synchronized with UTC leap seconds are added or deducted from this count. After 136+ years of seconds since 1/1/1970, the SOC value will reset to 0 in year 2106.

Each second is divided into an integer number of subdivisions by the TIME_BASE parameter that is defined in configuration frame. The FRACSEC count is an integer representing the numerator of the FRACSEC with TIME_BASE as the denominator.

Time stamp = SOC + FRACSEC / TIME_BASE

Table 3.1: Required PMU reporting rates

System frequency	**50 Hz**			**60 Hz**					
Reporting Rates (F_s-frames per second)	10	25	50	10	12	15	20	30	60

Synchrophasor transmission rates

Synchrophasor measurements are transmitted at a rate F_s that is an integer number of times per second or integer number of seconds per frame as specified by the DATA_RATE variable in the configuration frame.

Based on IEEE standard, PMUs shall be capable of transmitting data at the rates of submultiples of the nominal power system frequency. standard rates for 50 Hz and 60 Hz systems are listed in Table 3.1. In addition to these ranges, higher rates of 100 frames/sec or 120 frames/sec and lower rates than 10 frames/sec such as 1 fram/sec are suggested to be considered for the capability of the PMUs.

3.2 Synchrophasor data exchnage mechanism

In the context of IEEE Std C37.118, synchrophasor data transmission is handled by exchange of four message types:

1. Data Message
2. Configuration Message
3. Header Message
4. Command Message

The first three message types are transmitted from the data source (PMU/PDC), and the last one (command) is received by the PMU/PDC.

The *Data message* contains the synchrophasor measurements made by the PMU. In normal operation, the PMU continuously streams the Data messages. In order to enable the receiver to interpret the Data messages and extract the real values of measurements, *Configuration message* is defined as a machine-readable frame describing the data types, calibration factors, and other meta-data.

The *Header message* is human readable, it provides user-defined descriptive information sent from the PMU/PDC. *Command messages* are machine-readable codes sent to the PMU/PDC for control or configuration.

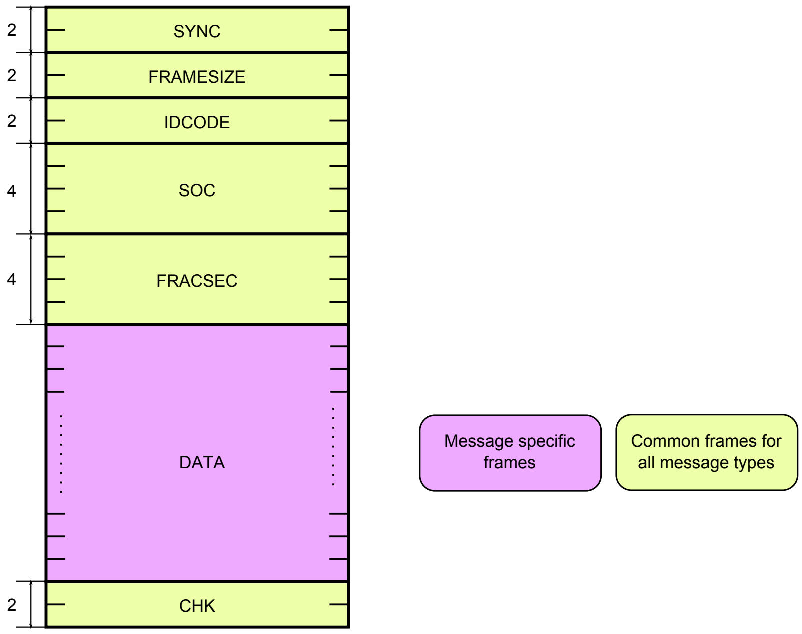


Figure 3.1: Common and Message-specific frame structure

3.2.1 General message structure

In spite of definition of four different types of messages, all messages have one standard structure. As shown in Fig. 3.1 a C37.118 message contains (1) Common and (2) Message-specific frames. The common part that is the same in all four types of messages contains:

- SYNC
  The 2-byte length word that provides synchronization and frame identification as detailed in Fig. 3.2. The first byte of the SYNC word always starts with `0xAA` value. Bits 0-3 are indicating the version of C37.118 standard used in the implementation. Bits 4-6 defines the type of the frame from any of the four defined ones.

- FRAMESIZE
  16-bit unsigned integer number, indicating the total number of bytes in the frame, including CHK.

- IDCODE
  This 2-byte length word identifies the source of a data, header, or configuration message, or the destination of a command message.

- SOC
  The 32-bit unsigned integer time stamp starting at midnight 01-Jan-1970.

- FRACSEC
  The 4-byte length word, containing Time Quality in bits 31-24 and 24-bit unsigned integer number of Fraction of Seconds in bits 23-00. The detailed definition of time stamp in IEEE C37.118 standard is shown in Fig. 3.3.

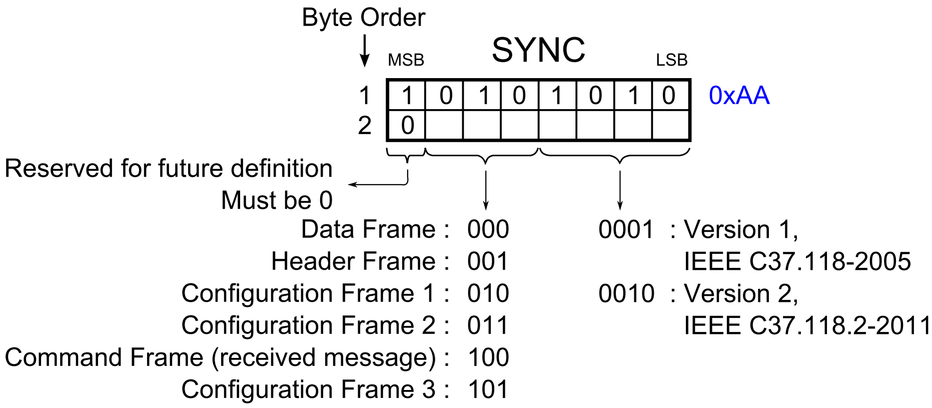


Figure 3.2: SYNC word definition

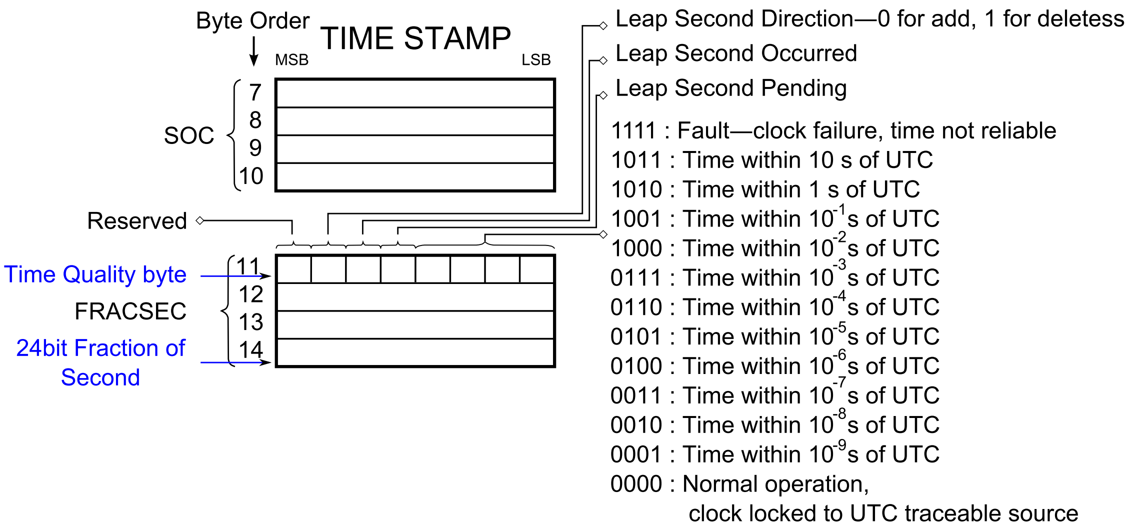


Figure 3.3: IEEE C37.118 time stamp word definition

No.	Field	Size (bytes)	Comment
1	SYNC	2	Sync byte followed by frame type and version number.
2	FRAMESIZE	2	Number of bytes in frame, defined in 6.2.
3	IDCODE	2	Stream source ID number, 16-bit integer, defined in 6.2.
4	SOC	4	SOC time stamp, defined in 6.2, for all measurements in frame.
5	FRACSEC	4	Fraction of Second and Time Quality, defined in 6.2, for all measurements in frame.
6	STAT	2	Bit-mapped flags.
7	PHASORS	4 × PHNMR or 8 × PHNMR	Phasor estimates. May be single phase or 3-phase positive, negative, or zero sequence. Four or 8 bytes each depending on the fixed 16-bit or floating-point format used, as indicated by the FORMAT field in the configuration frame. The number of values is determined by the PHNMR field in configuration 1, 2, and 3 frames.
8	FREQ	2 / 4	Frequency (fixed or floating point).
9	DFREQ	2 / 4	ROCOF (fixed or floating point).
10	ANALOG	2 × ANNMR or 4 × ANNMR	Analog data, 2 or 4 bytes per value depending on fixed or floating-point format used, as indicated by the FORMAT field in configuration 1, 2, and 3 frames. The number of values is determined by the ANNMR field in configuration 1, 2, and 3 frames.
11	DIGITAL	2 × DGNMR	Digital data, usually representing 16 digital status points (channels). The number of values is determined by the DGNMR field in configuration 1, 2, and 3 frames.
	Repeat 6–11		Fields 6–11 are repeated for as many PMUs as in NUM_PMU field in configuration frame.
12+	CHK	2	CRC-CCITT

Source: Section 6.3 of [18]

Table 3.2: Frame specification of Data message

- CHK
  All frames terminate in 2-byte length check word (CHK). This 16 bit word is used to verify (or at least indicate) that a set of data has not been corrupted.

  The algorithm of calculating CHK is explained in Annex A.

As already shown in Fig. 3.1, the SYNC word is transmitted first and CHK word last. Integer and floating-point numbers are transmitted most significant byte first (network or "big endian" order) and all frame types use this same order and format.

3.2.2 Data Message

The Data message contains the measured data. After the common words of SYNC, FRAMESIZE, IDCODE, SOC and FRACSEC, the PMU data is contained in *STAT*, *PHASORS*, *FREQ*, *DFREQ*, *ANALOG*, *DIGITAL* words. The byte order definition of Data message is presented in detail in Table 3.2. Following the transmission order shown in Fig. 3.4, the Data-frame specific part of a C37.118 message contains:

- STAT flag
  As can be seen in Fig. 3.4, each data message contains one block of data from a single

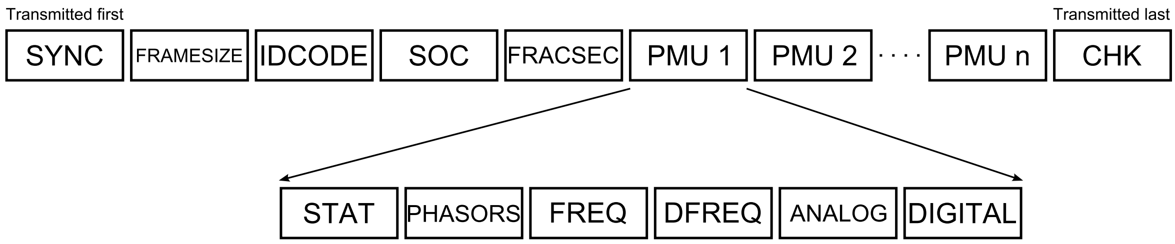


Figure 3.4: Data frame transmission order

PMU or multiple blocks from multiple PMUs. Each PMU data block is started by a 2-byte STAT word. The bits inside the STAT flag gives complete status of the data contained in that block. Bits are initially set by the PMU that generates the data, and then can be altered by next processors in the data chain, such as PDCs. The detail definition of STAT flag is presented in Fig. 3.5.

- PHASORS
  Based on the definition of FORMAT word in the configuration frames, PHASORS holds the phasor data in either *16-bit Integer* or *32-bit IEEE floating-point* format, and in each case, the complex value of phasors represented in *Rectangular* or *Polar* format.

  In polar format the first sent magnitude value is followed by angle value. In rectangular format, the real value is send first and then imaginary value.

  - In IEEE floating-point format, in either cases of polar and rectangular format, without any conversion, the exact measured values of phasors are transmitted. In polar format, the angle is in radians, ranging $-\pi$ to $+\pi$.

  In 16-bit integer format, each PHASORS word is limited to the range of -32767 to +32767, therefore the converted value of measured phasors are transmitted. In polar format, the angle is in radians$\times 10^4$, ranging -31416 to +31416.

  - In 16-bit integer format, the relation between the magnitude of measured phasor values and values contained in PHASORS word is described as below. The PHUNIT is a 4-byte word, defined in configuration frames as the conversion factor for phasor channels.

$$\text{Phasor value} \quad = \quad \text{PHASORS} \quad \times \quad \text{PHUNIT} \times 10^{-5}$$

- FREQ
  Based on the data type defined in the FORMAT word of configuration frame, the value of system frequency is transmitted either in *16-bit Integer* or *32-bit IEEE floating-point* format.

  - In floating-point format, actual value of frequency is transmitted.

  - In 16-bit integer format, instead of actual frequency, the value of "Frequency deviation from nominal system frequency" (50Hz or 60Hz) is transmitted. The unit of

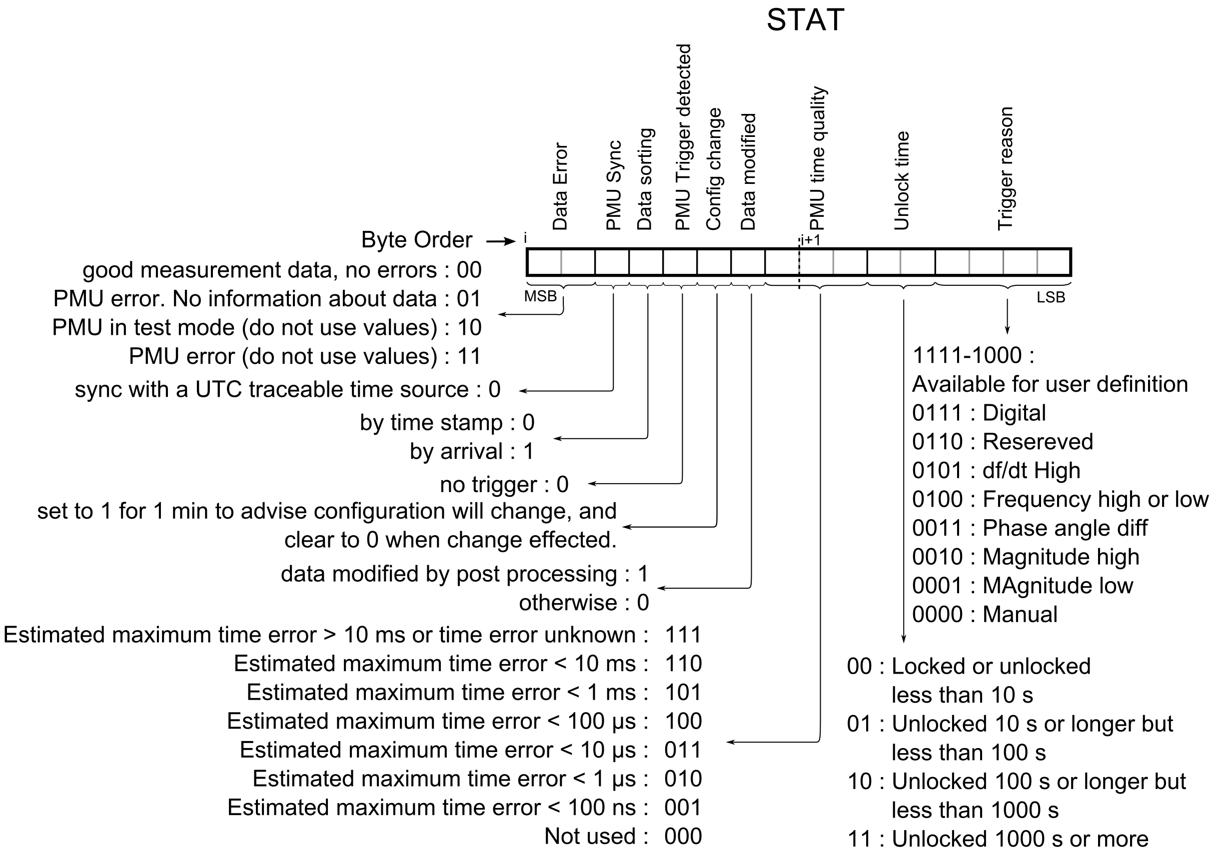


Figure 3.5: STAT word definition

this value is mHz and ranges -32767 to +32767 mHz. In this case, the relation between the actual system frequency value and value of FREQ word is described as below.

$$\text{Frequency value} \quad = \quad \text{Nominal system frequency} \quad + \quad \text{FREQ} \times 10^{-3}$$

- DFREQ
  Based on the definition of FORMAT word in configuration frames, the measured value of the "Rate of Change of Frequency (ROCOF)" is transmitted either in *16-bit Integer* or *32-bit IEEE floating-point* format.

  - In floating-point format, actual value of ROCOF is transmitted.

  - In 16-bit integer format, the value of ROCOF×100 (hertz per second) is transmitted with the range between -327.67 to +327.67 Hz/sec. In this format, the relation between the actual measured value of ROCOF and value of DFREQ word is described as below.

$$\text{ROCOF value} = \text{DFREQ} \times 0.01$$

- ANALOG
  The ANALOG word could be used to transmit the user-defined sampled data such as control signal or transducer value.

  Based on the FORMAT word definition in configuration frames, it could be in either *16-bit Integer* or *32-bit IEEE floating-point* format. The values and ranges are user defined.

- DIGITAL
  Digital status word is defined to transmit user-defined binary data such as bit mapped status or flag. Values and ranges are user defined.

3.2.3 Configuration Message

In order to enable the receiver to interpret the Data message streams, configuration Message containing information and processing parameters is transmitted by PMU/PDC.

Three types of configuration message are defined in the C37.118.2-2011 standard.

- Configuration Frame 1 (CFG-1)
- Configuration Frame 2 (CFG-2)
- Configuration Frame 3 (CFG-3)

CFG-1 and CFG-2 are already defined in the version 1 (IEEE Std C37.118-2005), and CFG-3 is a new type of message introduced with version 2 (IEEE Std C37.118.2-2011). In the current version of IEEE C37.118.2-2011, CFG-3 is optional.

CFG-1 provide information about the PMU/PDC capability. it defines the set of data that the PMU/PDC is capable of reporting [30]. CFG-2 gives information about the synchrophasor data being transmitted in data frame. As it is described in Table 3.3, CFG-1 and CFG-2 messages have both similar structure. They contains 19 fields, and based on the number of PMUs whose data are contained in the message, fields 8–19 are repeated. These fields are described as below:

- TIME_BASE

  The 32-bit unsigned integer word defines the resolution of the fractional second time stamp (FRACSEC) in all frames.

- NUM_PMU

  The 16-bit unsigned integer gives the number of PMU included in the data frame.

No	Field	Size (bytes)	Short description
1	SYNC	2	Sync byte followed by frame type and version number.
2	FRAMESIZE	2	Number of bytes in frame, defined in 6.2.
3	IDCODE	2	Stream source ID number, 16-bit integer, defined in 6.2.
4	SOC	4	SOC time stamp, defined in 6.2.
5	FRACSEC	4	Fraction of Second and Message Time Quality, defined in 6.2.
6	TIME_BASE	4	Resolution of FRACSEC time stamp.
7	NUM_PMU	2	The number of PMUs included in the data frame.
8	STN	16	Station Name—16 bytes in ASCII format.
9	IDCODE	2	Data source ID number identifies source of each data block.
10	FORMAT	2	Data format within the data frame.
11	PHNMR	2	Number of phasors—2-byte integer (0 to 32 767).
12	ANNMR	2	Number of analog values—2-byte integer.
13	DGNMR	2	Number of digital status words—2-byte integer.
14	CHNAM	16 × (PHNMR + ANNMR + 16 × DGNMR)	Phasor and channel names—16 bytes for each phasor, analog, and each digital channel (16 channels in each digital word) in ASCII format in the same order as they are transmitted. For digital channels, the channel name order will be from the least significant to the most significant. (The first name is for bit 0 of the first 16-bit status word, the second is for bit 1, etc., up to bit 15. If there is more than 1 digital status, the next name will apply to bit 0 of the second word and so on.)
15	PHUNIT	4 × PHNMR	Conversion factor for phasor channels.
16	ANUNIT	4 × ANNMR	Conversion factor for analog channels.
17	DIGUNIT	4 × DGNMR	Mask words for digital status words.
18	FNOM	2	Nominal line frequency code and flags.
19	*CFGCNT*	2	Configuration change count.
	Repeat 8–19		Fields 8—19, repeated for as many PMUs as in field 7 (NUM_PMU).
20+	DATA_RATE	2	Rate of data transmissions.
21+	CHK	2	CRC-CCITT.

Source: Section 6.4 of [18]

Table 3.3: Configuration frame 1 and 2 organization

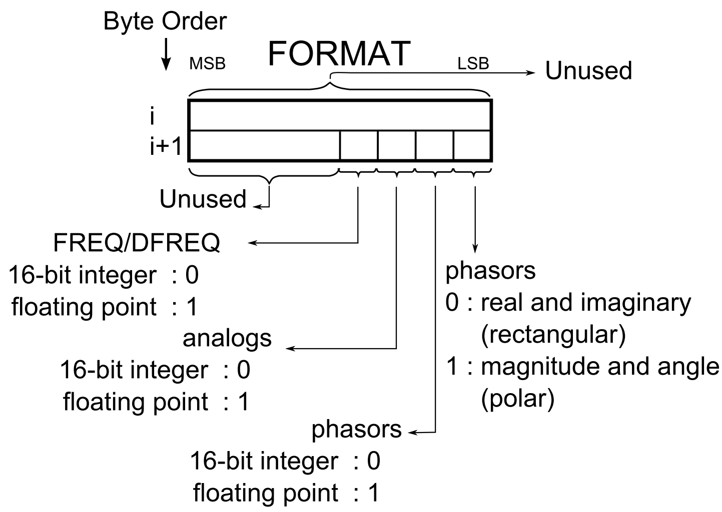


Figure 3.6: FORMAT word definition

- STN

  The 16-byte in ASCII format gives the "Station Name".

- IDCODE

  The 16-bit integer gives the Data stream ID number.

  The IDCODE in field 3 of common frames identifies the data stream being transmitted. The IDCODE in field 9 (and higher if more than one PMU data is present) of configuration-message specific frames identifies the original source of the data i.e. a particular PMU.

- FORMAT

  The 16-bit flag which defines the format of data in Data message. The specification of bits are described in Fig. 3.6. Bit 0 defines if the phasors are transmitted in *Polar* or *Rectangular* format. Bits 1, 2 and 3 consecutively define if the Phasors, Analogs and FREQ/DFREQ are sent in *16-bit Integer* or *32-bit floating-point* format.

- PHNMR

  The 16-bit integer word defines the number of phasors contained in the data frame.

- ANNMR

  The 16-bit integer word defines the number of analog values contained in the data frame.

- DGNMR

  The 16-bit integer word defines the number of digital status words contained in the data frame.

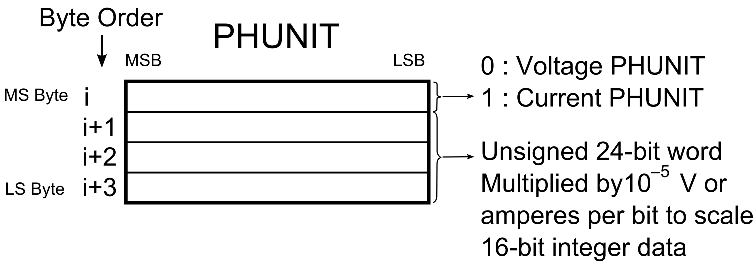


Figure 3.7: PHUNIT word definition

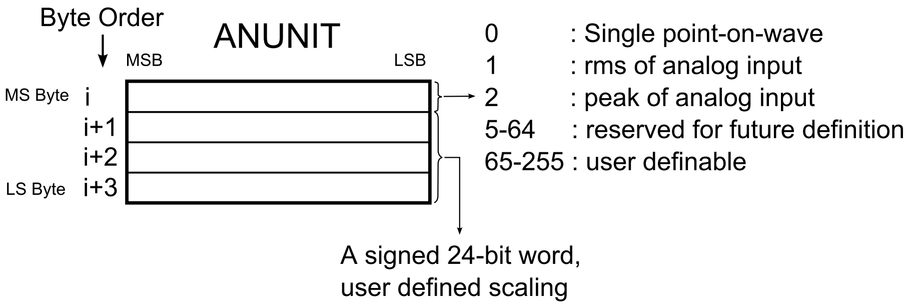


Figure 3.8: ANUNIT word definition

- CHNAM

  This frame consist of 16-byte channel names for each Phasor, Analog value and each status bit in Digital word.

- PHUNIT

  For each phasor the 4-byte Conversion factor is defined. This word is defined in Fig. 3.7. In case of transmission of phasors in floating-point format, this value is ignored. The first byte of this word indicates if the phasor is current of voltage.

- ANUNIT

  For each analog value the 4-byte Conversion factor is defined. This word is defined in Fig. 3.8. The first byte of this word defines the type of value either *Single point-on-wave*, *RMS* or *Peak*.

- DIGUNIT

  Mask words for digital status words. Two 16-bit words are provided for each digital word. The first byte is used to indicate the normal status of the digital inputs by returning a 0 when exclusive ORed (XOR) with the status word. The second byte indicates the current valid inputs to the PMU by having a bit set in the binary position corresponding to the digital input and all other bits set to 0.

- FNOM

  16-bit unsigned integer indicating the nominal line frequency as shown in Fig. 3.9. while the 15 most significant bits of the word is reserved, the least significant bit, bit 0, indicates if the nominal frequency either 50 Hz or 60 Hz.

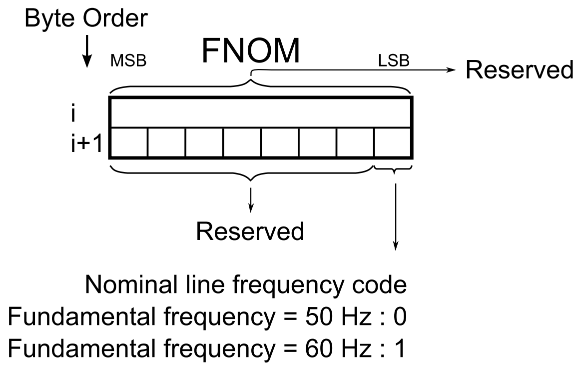


Figure 3.9: FNOM word definition

- DATA_RATE

  The 2-byte integer word in the range of –32767 to +32767, indicates the Rate of data transmissions.
  If $DATA\_RATE > 0$, rate is number of frames per second.
  If $DATA\_RATE < 0$, rate is negative of seconds per frame.

- CFGCNT
  Configuration change count is incremented each time a change is made in the PMU configuration.

CFG-3 frame is similar to other configuration frames, but with some additional and modified data words in compare with CFG-1 and CFG-2 frames. Some information from CFG-1 and CFG-2 are considered without change in CFG-3 (for instance: FORMAT, PHNMR,...), however some others are changed and presented in different format (for example: PHSCALE is replaced by PHUNIT) and finally some new words are added (for example: G_PMU_ID,...).

CFG-3, similar to CFG-2, provides configuration data required for interpreting the measurements currently being streamed in the data frame. CFG-3 is optional and has variable length fields, added PMU and signal information, and extendable frame.

3.2.4 Header Message

This message type transmits the human readable description information about the PMU, the data sources, scaling, algorithms, filtering, or other related information. The Header message structure is presented in Table 3.4.

3.2.5 Command frame

PMU or PDC that are generating synchrophasor data shall be able to receive commands to perform specific functions. For this purpose, Command frames are defined. The command message specification is shown in Table 3.5.

Following functions are defined to be sent as commands to PMU/PDC.

No	Field	Size (bytes)	Comment
1	SYNC	2	Sync byte followed by frame type and version number (AA11 hex).
2	FRAMESIZE	2	Number of bytes in frame, defined in 6.2.
3	IDCODE	2	PMU/PDC data stream ID number, 16-bit integer, defined in 6.2.
4	SOC	4	SOC time stamp, defined in 6.2.
5	FRACSEC	4	Fraction of Second and Time Quality, defined in 6.2.
6	DATA 1	1	ASCII character, 1st byte.
K+6	DATA k	1	ASCII character, Kth byte, K>0 is an integer.
K+7	CHK	2	CRC-CCITT.

Source: Section 6.5 of [18]

Table 3.4: Header frame organization

No	Field	Size (bytes)	Comment
1	SYNC	2	Sync byte followed by frame type and version number (AA41 hex).
2	FRAMESIZE	2	Number of bytes in frame, defined in 6.2.
3	IDCODE	2	PMU/PDC ID data stream number, 16-bit integer, defined in 6.2.
4	SOC	4	SOC time stamp, defined in 6.2.
5	FRACSEC	4	Fraction of Second and Time Quality, defined in 6.2.
6	CMD	2	Command being sent to the PMU/PDC (0).
7	EXTFRAME	0–65518	Extended frame data, 16-bit words, 0 to 65518 bytes as indicated by frame size, data user defined.
8	CHK	2	CRC-CCITT.

Source: Section 6.6 of [18]

Table 3.5: Command frame organization

- Turn off transmission of data frames
- Turn on transmission of data frames
- Send Header frame
- Send CFG-1 frame
- Send CFG-2 frame
- Send CFG-3 frame

3.3 Communication mechanisms

In IEEE Std C37.118.2-2011, no specific underlying communication protocol is introduced and just define four frame types and their specification are defined. However in Annex E and F of the standard two protocols (*Serial* and *Internet protocol*) are suggested. The Serial communication protocol is out-dated, and in practice TCP/IP or UDP/IP protocols are used for synchrophasor data transmission.

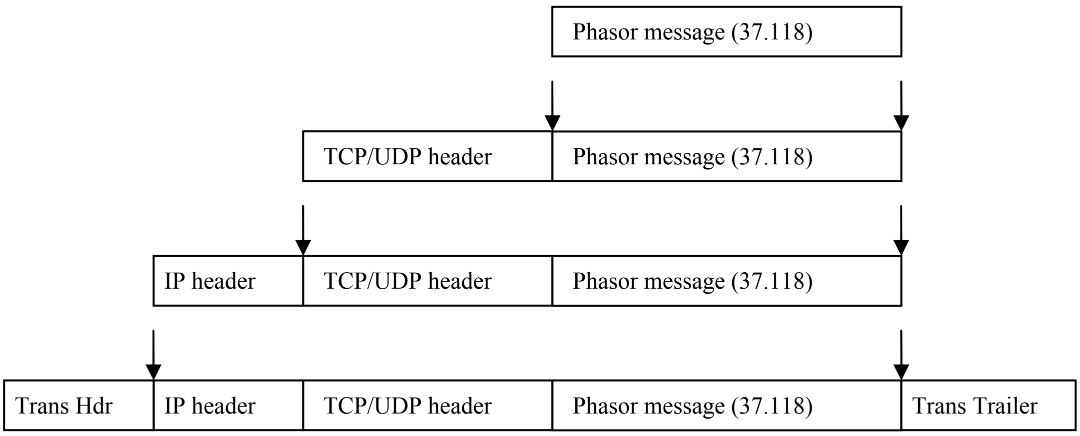


Figure 3.10: Mapping of IEEE C37-118 data into a TCP or UDP packet

3.3.1 Internet protocol (IP)

According to IEEE Std C37.118-2011, the synchrophasor messages shall be mapped in their entirety into TCP or UDP segments. Then they will be written to and read from using IP datagram.

$$\begin{aligned}\text{Default Port Number (TCP)} &= 4712\\ \text{Default Port Number (UDP)} &= 4713\end{aligned}$$

Standard introduced the default port numbers for both TCP and UDP, however, it is mentioned that the user shall be able to set the desired port numbers. The IP may be carried over Ethernet or another transport means. The encapsulation process of the sunchrophasor messages over TCP(UDP)/IP is illustrated in Fig. 3.10.

In Annex F of the standard, both TCP and UDP methods are reviewed and their advantages and disadvantages are explained. TCP is a two-way connection based protocol that is performed based on a 1-to-1 connection, ensures transmission of all packets through a Client-Server connection. UDP is connection-less protocol that transmits the frames based on a publisher-subscriber connection without ensuring if all packets are transmitted correctly.

Then the TCP/UDP method is suggested as the most complete of the phasor communication methods. In this method, TCP is used to transmit the commands, header and configuration messages which contains the critical data and UDP is considered for streaming the data messages.

In UDP method, the data could be transmitted to a unicast IP address or to a multicast IP address, enabling numerous clients to receive data and in consequence reduce the traffic.

4 IEC TR 61850-90-5

4.1 Introduction

IEC TR 61850-90-5 is a protocol for transmitting digital state and time synchronized power measurement over wide area networks enabling implementation of wide area situational awareness (WAMS) as well as wide area measurement and protection and control (WAMPAC) systems based on the IEC 61850 protocols commonly used in substation automation [2].

4.2 Data Modeling

To describe a system in IEC 61850, each client and server shall be modeled as a logical node on some IEDs. Protection, monitoring and control systems that use synchrophasors are complex hierarchical systems with PMUs at the bottom of the hierarchy and PDCs at the different levels of the hierarchy.

In this regard, PMU is modeled as a function within an IED, calculating and publishing the synchrophasor measurements as defined in IEEE C37.118.1.

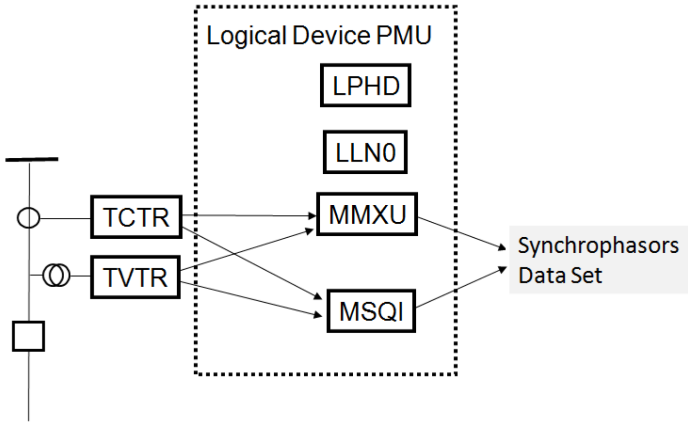
As it is shown in Fig. 4.1 if PMU is publishing phase current and voltages, one or more instances of MMXU logical node will be used and if the PMU is publishing sequence currents and voltages, one or more instances of MSQI logical node will be considered in the model. [3]

Measurement Logical Node named as MMXU is defined in part 7-4 of IEC 61850 to be used for calculation of currents, voltages, powers and impedances in a three-phase system [27]. In section 13.3 of [3] it is mentioned that in case of mapping IEEE C37-118 to IEC, the phasor data and the frequency contained in C37-118 telegram map directly to data objects of MMXU within IEC 61850.

The unspecified analog and digital data in a C37-118-2 telegram should be mapped to any IEC 61850 data object fitting to the appropriate data type and carrying the needed semantic

Data object for Rate Of Change Of Frequency(ROCOF) is newly defined in the MMXU. By introduction of IEC 61850-90-5, the new data object of HzRte is added to MMXU logical node to accommodate the ROCOF data. Then the Phasors, Frequency and ROCOF data in the context of IEEE C37-118 are modeled as the data objects of logical node MMXU as depicted in Fig. 4.2.

In addition to phasor data, the information about the status of the PMU is transmitted using the common data class named ”PhyHealth” in an instance of LPHD Logical Node.



Source: IEC TR 61850-90-5 [3]

Figure 4.1: PMU Model

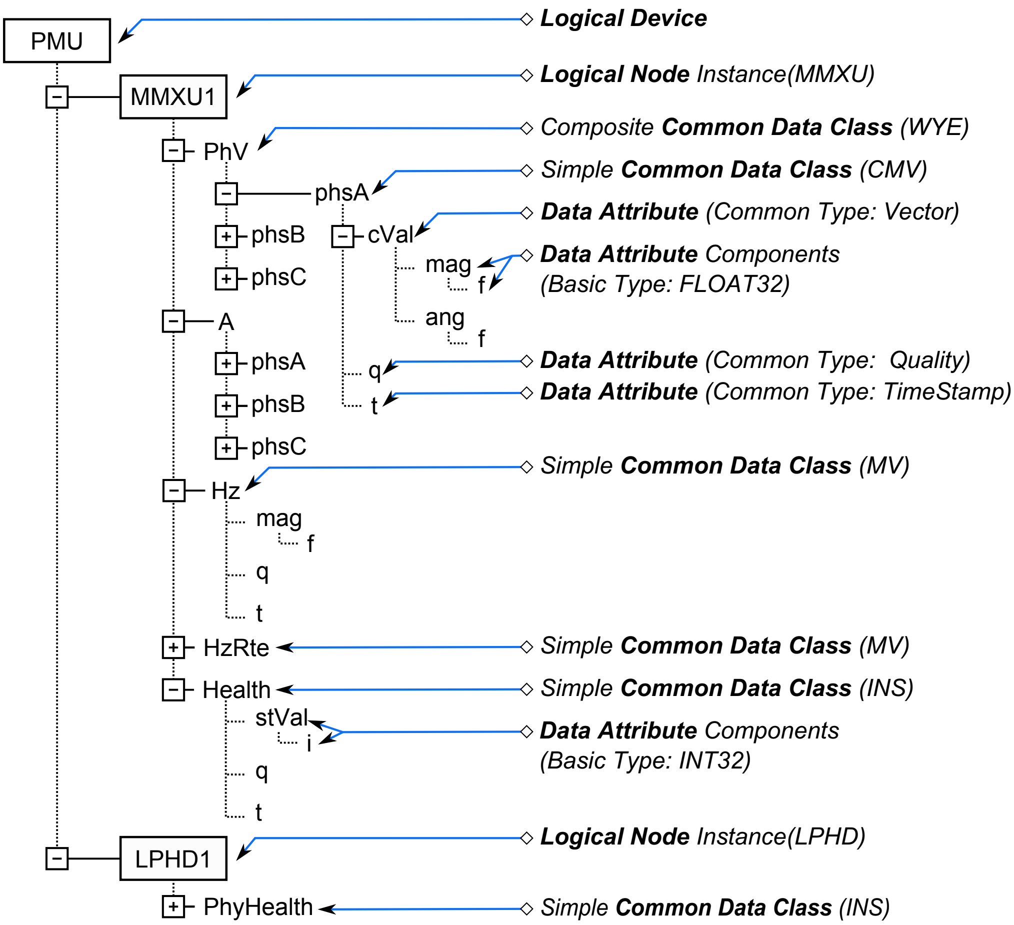


Figure 4.2: IEC 61850 PMU Data Model

Based on the data model presented in Figure. 4.2, the dataset of the PMU/PDC synchrophasor data to be transferred using either Routed-Sampled Value or Routed-GOOSE services shall be defined. In our work, the dataset is defined to include the whole PMU data. For this purpose, based on the example SCL source codes presented in Annex A and B of [3], disregarding the Analog and Digital data of the IEEE C37.118.2 data streams, the following XML lines define the most comprehensive dataset covering the three phases of Voltage, Current, Frequency, ROCOF and PMU health status data.

```
<DataSet name = "PMUdata">
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="PhV.phsA" fc="MX"/
    >
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="PhV.phsB" fc="MX"/
    >
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="PhV.phsC" fc="MX"/
    >
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="A.phsA" fc="MX"/>
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="A.phsB" fc="MX"/>
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="A.phsC" fc="MX"/>
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="Hz" fc="MX"/>
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="HzRte" fc="MX"/>
<FCDA ldInst="PMU" prefix="" lnClass="MMXU" lnInst="1" doName="Health" fc="ST"/>
<FCDA ldInst="PMU" prefix="" lnClass="LPHD" lnInst="1" doName="PhyHealth" fc="ST"
    />
</DataSet>
```

4.3 Communication mechanism

In IEC 61850, the SV service is introduced for cyclic transfer of data inside substation and the GOOSE service is defined for event-based transfer of data. However, these communication protocols are over Ethernet inside a substation.

For the purpose of data transfer outside of the substation, there are two options introduced in [3]:

- *Tunneling*
  In this mechanism, SV and GOOSE services are tunneled across some high speed communication networks like SDH or SONET.

- Internet Protocols (IP) networks
  In this mechanism, SV and GOOSE services are communicated via IP networks.

In the second option, the IEC 61850 has been enhanced by mapping of SV and GOOSE messages onto an IP based protocol. Based on the cyclic nature of these services, UDP with multicast addressing is the transport protocol is chosen in the standard.

In this regard, the new mapping of the SV and GOOSE services uses routable UDP, and are called Routed-Sample Value and Routed-GOOSE or in brief, R-SV or R-GOOSE.

While both Tunneling and IP Protocol mechanisms are addressed in [3], the scope of this work is limited to implementation of R-SV and R-GOOSE services.

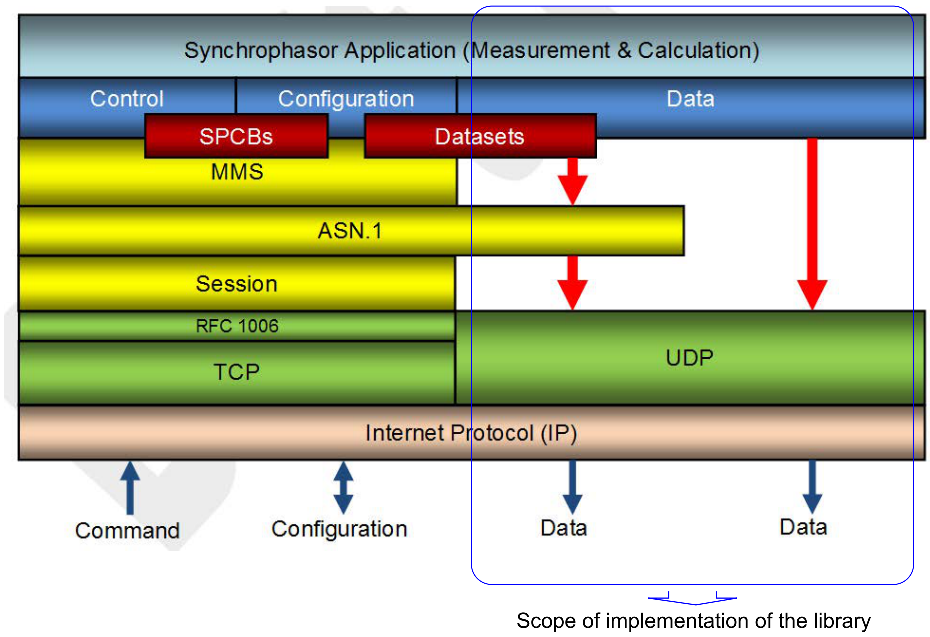


Figure 4.3: Mapping services in the scope of IEC TR 61850-90-5

4.4 Synchrophasor data transfer services

As it is illustrated in Fig. 4.3, the functions performed by transfer of four different *Data*, *Configuration*, *Command* and *Header* message types defined in IEEE C37.118.2 and described in Chapter. 3 are mapped to IEC 61850 services.

Command service

The command service is defined as the equivalent to the function executed by the command message. Table 4.1 describes the equivalent IEC 61850 services performing different equivalent functions defined by IEEE C37.118.2 command message.

Configuration request service

As could be seen in Table 4.1, getting the information of the capabilities of server using the CFG-1 in IEEE standard is equivalent to obtaining data model structure of the server.

In the domain of IEC 61850, the Substation Configuration Language (SCL) is used to define the overall information contents of a unit, that is equivalent to exchange of the configuration messages defined in IEEE C37.118 standard. Therefore in the IEC 61850 world, SCL is used to communicate the configuration of a C37.118 device [3]. In addition, the transferred information in Header frames are already provided in SCL.

The equivalent of the CFG-2, is to read actual measurements from data model. It should be noted that in IEC 61850 data model, the actual value of synchrophasors are transmitted.

Command word	Definition (IEEE C37.118.2)	Equivalent IEC 61850 action /service
1	Turn off transmission of frames	Disable SVCB (set SvEna in SVCB to False)
2	Turn on transmission of frames	Enable SVCB (set SvEna in SVCB to True)
3	Send header	Read information for FC "DC" and read SvCB
4	Send CFG-1 information	Obtain data model of PMU related functions (e.g. logical device PMU or LNs representing PMU functions) as well as all other LNs in the device.
5	Send CFG-2 information	Read actual measurements from data model (MMXU, etc.)
8	Extended frame	Out-of-scope

Source: Section 9.2.1 of [3]

Table 4.1: Equivalent commands

Header information service

The Header message in IEEE C37.118 is defined to provide human readable information. However the high degree of readability provided by data modeling in IEC 61850, resulted in no need for defining special methods to resemble the functions of the Header messages.

Data transmission service

While in IEEE standard the synchrophasor data are suggested to be transmitted through either RS-232 serial or TCP(UCP)/IP protocol, in the context of IEC 61850-90-5, for the purpose of synchrophasor data transfer in wide-area applications the following services are introduced:

1. Routed Sampled Value service
2. Routed GOOSE service
3. Tunneled GOOSE or Sampled Value service
4. Management service

Among the four defined services for Data exchange, the first two services are implemented in this thesis project. Implementation of Command and Configuration exchange requires implementation of Manufacturing Message Specification (MMS) protocol, that is introduced as the future works of this project.

4.5 R-Sampled Value and R-GOOSE Message specifications

As mentioned in previous section, two services of Routed-SV and Routed-GOOSE are proposed for data transfer services. In this case, while the application layer specifications of

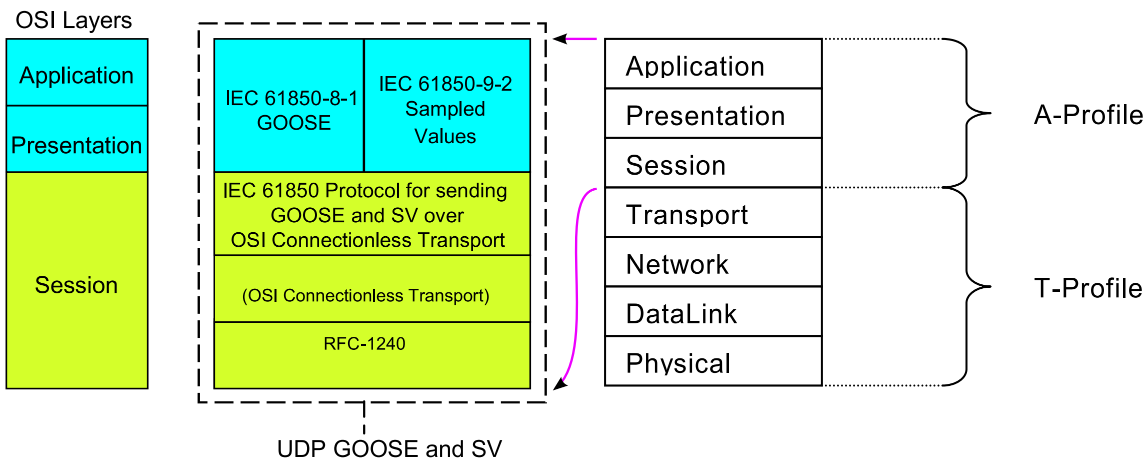


Figure 4.4: OSI Reference model and R-SV/R-GOOSE Application profiles

GOOSE service based on IEC 61850-8-1 and SV service based on IEC 61850-9-2 standard remained unchanged, IEC 61850-90-5 introduced a new protocol in session layer for sending the GOOSE and SV over OSI connectionless transport.

As illustrated in Fig. 4.4, the seven layers of Open System Interconnect (OSI) model are divided into two profiles of *Application* (A-profile) and *Transport* (T-Profile). The R-GOOSE and R-SV services are in fact implementation of the Application-profile based on the requirements of IEC 61850-90-5.

4.5.1 Application profile specification

In order to explain the A-profile defined in [3], the protocol introduced at the session layer shall be explained. Therefore, in the following the IEC 61850-90-5 *Session Protocol* is described:

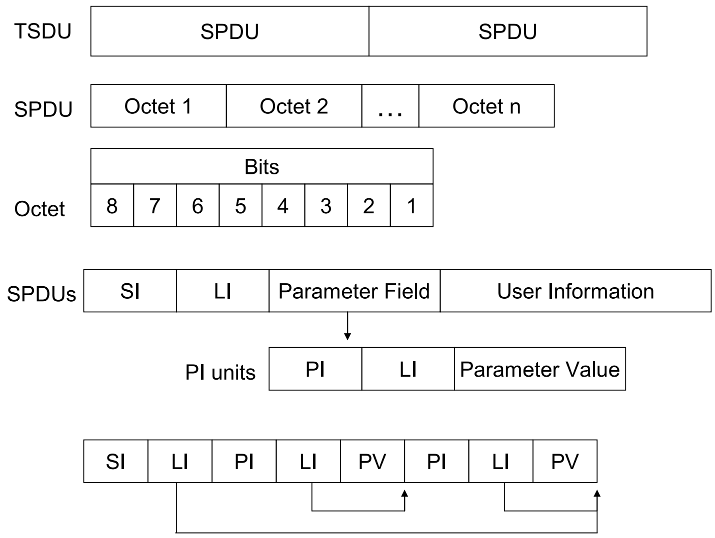
Session Header

In Fig. 4.5 the general construction of the session protocol and the bit and byte ordering of the transmission is presented. The data packets which are generated at session layer called Session Protocol Data Unit (SPDUs) are fed into the transport layer as Transport Service Data Unit (TSDU).

Each SPDU starts with a single-byte Session Identifier (SI), followed by the single-byte length. This length covers the length of all of the parameter fields for the session header, but not the user data of the session protocol.

There are four Session Identifier defined in the standard that are detailed in Table 4.2. The session protocol structure presented in IEC TR 61860-90-5 is shown in Fig. 4.6.

The associated Length field of the Session Identifier is the length of entire session header. The Session Identifier and associated length fields are followed by Common Session header as the Parameter Identifier (PI) with the hexadecimal value of 0x80. The Parameter Value (PV) of the session header shall contain a sequence of the following values:



Source: Section 11.3.2 of [3]

Figure 4.5: General byte ordering of session protocol

- SPDU Length
  The fixed size 4-byte word with maximum value of 65,517 octets. The length that is indicated by SPDU Length is shown in detail in Fig. 4.7.
- SPDU Number
  The fixed size 4-byte unsigned integer word in the range of 0 to 4,294,967,295 that can be used by the subscriber to detect duplicate or out-of-order packet delivery.
- Version
  This fixed size 2-byte unsigned integer attribute contains the session protocol version number. The value assigned in this standard is 1.
- TimeofCurrentKey
  The fixed size 4-byte unsigned integer word indicates the SecondsSinceEpoch.
- TimetoNextKey
  This fixed size 2-byte signed integer value represents the number of minutes prior to a new key being used.

Item	**SI Description**	**SI Hexadecimal value**
1	Tunneled GOOSE and Sampled Value Packets	0xA0
2	Non-tunneled GOOSE APDUs:	0xA1
3	Non-tunneled SV APDUs:	0xA2
4	Non-tunneled management APDUs	0xA3

Table 4.2: SI values and descriptions

Source: Section 11.3.2 of [3]

Figure 4.6: IEC TR 61850-90-5 session protocol

- SecurityAlgorithms
  This fixed size 2-byte attribute, indicates the type of encryption and HMAC algorithm information regarding the signature generation.

- Key ID
  This fixed size 4-byte attribute is assigned by the Key Distribution Center (KDC).

The session header is then followed by Session User Information.

User data

The session user data, consists of two fields: *Length* and *Payload*, then ended by *Signature*.

- Length
  The fixed size 4-byte unsigned integer attribute with maximum value of 65,399 octets.

- Payload
  In R-SV and R-GOOSE messages, the payload section is started with following words as the attributes of the payload.

  - *Payload type*
    This tag specifies the type of the payload. The hexadecimal value of this tag is presented in Table 4.3.
  - *Simulation*
    The one byte boolean word that indicates if the payload is sent for test or not.
  - *APPID*
    The 2-byte word as defined in IEC 61850-8-1
  - *APDU Length*
    The fixed size 2-byte unsigned integer value that contains the value of the number of octets of the SV or GOOSE APDU.

  Then the payload continued with GOOSE or SV APDUs. The GOOSE APDU is defined as the goosepdu introduced in IEC 61850-8-1 and explained in detail in section 4.5.2. The SV APDU is defined as the savPdu introduced in IEC 61850-9-2 and described in detail in section 4.5.3.

Item	Payload Type	Tag Hexadecimal value
1	Non-tunneled GOOSE APDUs:	0x81
2	Non-tunneled SV APDUs:	0x82
3	Tunneled GOOSE and Sampled Value Packets	0x83
4	Non-tunneled management APDUs	0x84

Table 4.3: Payload type tag definition

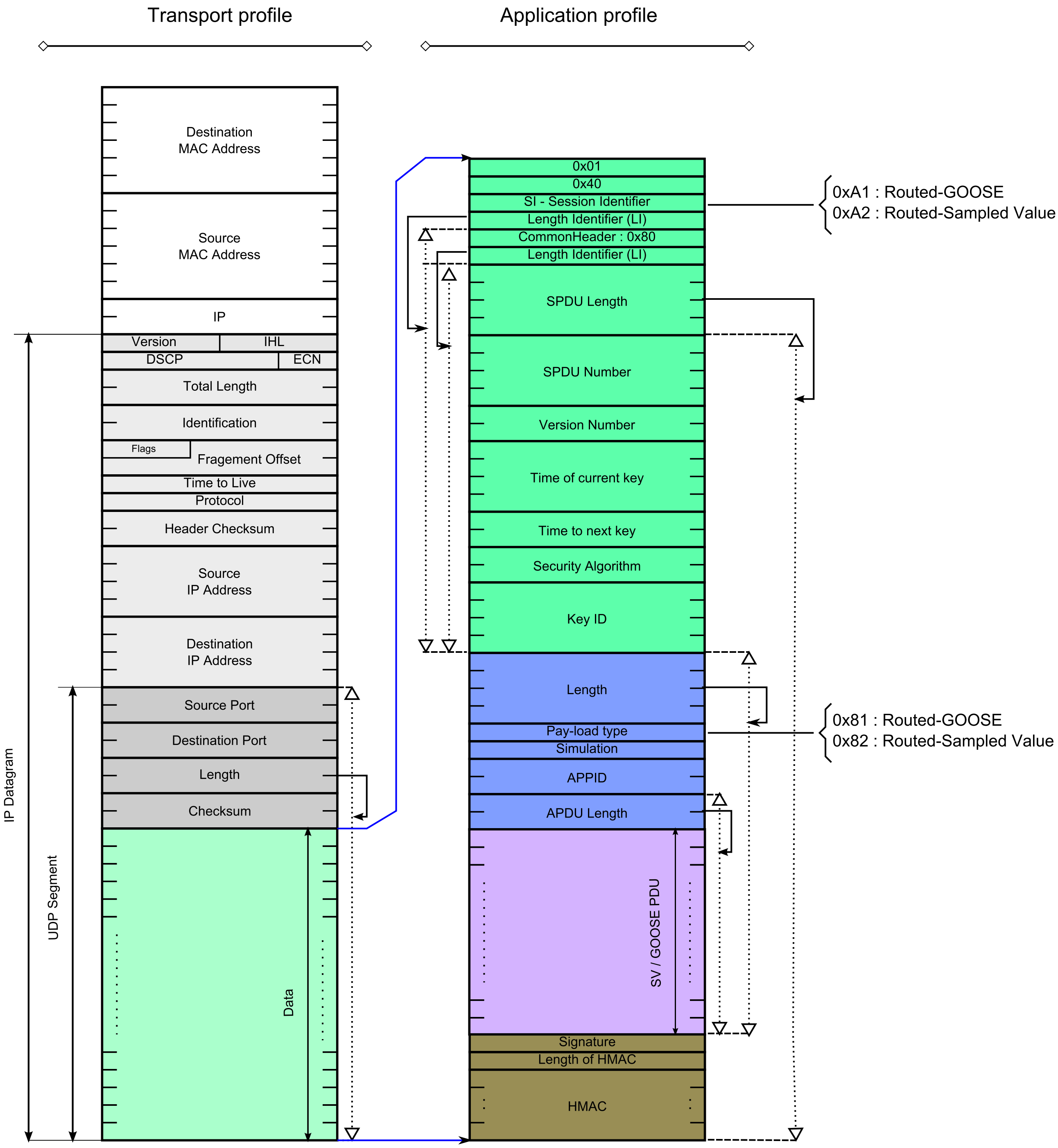


Figure 4.7: Complete encoding structure of Routed-GOOSE and Routed-SV message

- Signature
  The signature production starts with a one-octet tag with the hexadecimal value equal to 0x85. This tag is followed by a one-byte Length, indicating the length of the calculated signature. The third octet is the most significant byte of the calculated signature value.

  In the standard, for the testing purpose it is allowed to use MAC-None option. In this work, signature production is not considered, then no signature is generated and the value of the length is considered as zero.

The complete encoding structure of Routed-GOOSE and Routed-Sampled Value message including both A-profile and T-profile are illustrated in Fig 4.7. In the left hand side of the figure the Transport profile (T-Profile) and in the right hand side the Application Profile (A-Profile) is shown in detail.

In the following, the GOOSE Protocol Data Unit (goosePdu) and the Sampled Value Protocol Data Unit (svPdu) are described in detail.

4.5.2 GOOSE Protocol Data Unit (goosePdu) specification

Data contained in GOOSE PDU is encoded using OSI's Abstract Syntax Notation One (ASN.1), Basic Encoding Rules (BER).

ASN.1 Basic Encoding Rule

ASN.1 is a standard for data networks and open system communications. ASN.1 is an international standard used to define protocols of communication by means of encoding rules. Based on this standard, as shown in Fig. 4.8, every component of data is presented in form of *Tag*, *Length* and *Value* structure [28].

- **Tag**
  The one-byte Tag word indicates the type of information represented by the frame.
- **Length**
  The one-byte Length word gives the length of the data in form of the number of bytes of *Data*.

  If the length is less than 128 bytes, a single octet is used with its MSB set to 0. If the length is greater than 128, the MSB is set to 1, and the remaining 7 bits expressing the number of bytes that will contain the following parameter length.
- **Value**
  The value contains the actual data to be specified. The data inside the Value is consistent with the type specified by Tag word. In addition, each *Value* word may contains several other data coded in form of other TLVs.

The GOOSE message is not encoded exactly based on the original ASN.1/BER, but using a version of ASN.1/BER that is adapted to Manufacturing Message Specification (MMS) protocol.

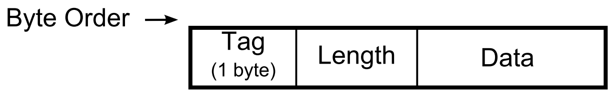


Figure 4.8: ASN. 1 encoding format

GOOSE Protocol Data Unit

In the following, the contents of the GOOSE pdu is described, that is formed based on the ASN.1 Basic Encoding Rule. The complete detailed structure of the GOOSE pdu is illustrated in Fig. 4.9.

The whole GOOSE pdu is coded, with *Tag* value equal to 0x61. The Tag is followed by the *Length* indicating the whole length of the GOOSE pdu.

The *Data* part of the GOOSE pdu contains a sequence of data as listed below:

- **GoCBRef**
  This Visible-String value with maximum size of 65 octets is the reference to the GOOSE control block that is controlling the GOOSE message.

  The Tag associated with GoCBRef is set to 0x80.

- **timeAllowedtoLive**
  This Integer value has a range of 1 to 4 294 967 295. The units of the value is milli seconds. Since GOOSE messages have a re-transmission process, each message carries a timeAllowedToLive parameter that informs the receiver of the maximum time to wait for the next re-transmission. If a new message is not received within that time interval, the receiver assumes that the association is lost.

  The Tag associated with timeAllowedtoLive is set to 0x81.

- **datSet**
  This Visible-String value with maximum size of 65 octets, contains the value of the DataSetReference, as found in the GOOSE control block specified by GoCRef.

  The Tag associated with datSet is set to 0x82.

- **goID**
  This optional Visible-String valuea with maximum size of 65 octets, contains the value of the DataSetReference, as found in the GOOSE control block specified by GoCRef.

  The Tag associated with goID is set to 0x83.

- **t**
  This data is the 8-byte UtcTime TIMESTAMP of the GOOSE message.

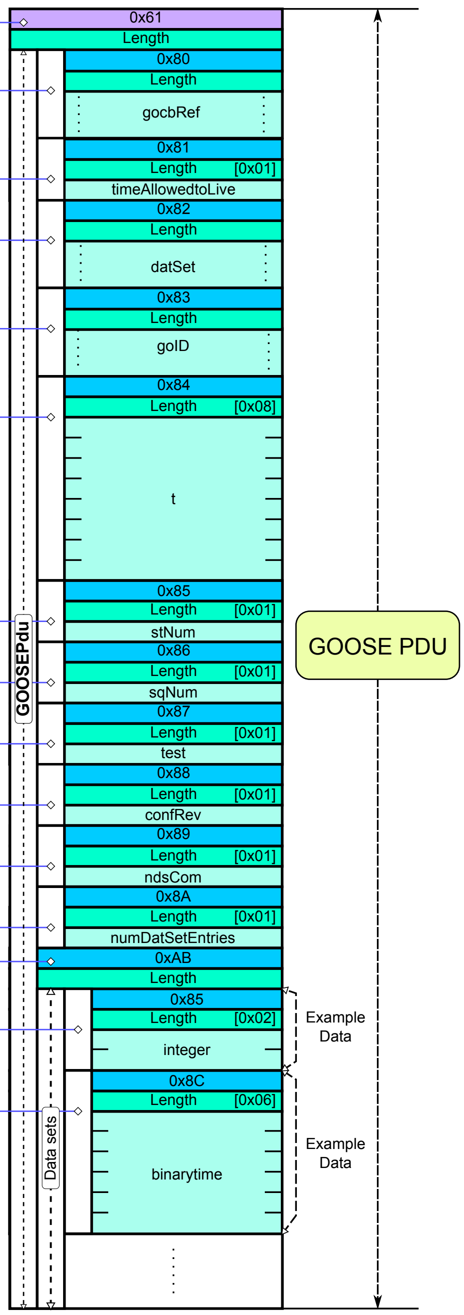
  The Tag associated with TIMESTAMP is set to 0x84.

- **stNum**
  This Integer value has ranges of 1 to 4,294,967,295.

  The Tag associated with stNum is set to 0x85.

```
IEC61850 DEFINITIONS ::= BEGIN

IEC 61850-8-1 Specific Protocol ::= CHOICE {
        gseMngtPdu          [APPLICATION 0] IMPLICIT GSEMngtPdu,
        goosePdu            [APPLICATION 1] IMPLICIT IECGoosePdu,
... }

IECGoosePdu ::= SEQUENCE {
        gocbRef             [0]  IMPLICIT VISIBLE-STRING,
        timeAllowedtoLive   [1]  IMPLICIT INTEGER,
        datSet              [2]  IMPLICIT VISIBLE-STRING,
        goID                [3]  IMPLICIT VISIBLE-STRING OPTIONAL,
        t                   [4]  IMPLICIT UtcTime,
        stNum               [5]  IMPLICIT INTEGER,
        sqNum               [6]  IMPLICIT INTEGER,
        test                [7]  IMPLICIT BOOLEAN DEFAULT FALSE,
        confRev             [8]  IMPLICIT INTEGER,
        ndsCom              [9]  IMPLICIT BOOLEAN DEFAULT FALSE,
        numDatSetEntries    [10] IMPLICIT INTEGER,
        allData             [11] IMPLICIT SEQUENCE OF Data,
        security            [12] ANY OPTIONAL,
                            -- reserved for digital signature
        }
Data ::= CHOICE{
        -- context tag 0 is reserved for AccessResult
        array               [1]  IMPLICIT DataSequence,
        structure           [2]  IMPLICIT DataSequence,
        boolean             [3]  IMPLICIT BOOLEAN,
        bitstring           [4]  IMPLICIT BIT STRING,
        integer             [5]  IMPLICIT INTEGER,
        unsigned            [6]  IMPLICIT INTEGER, -- shall not be negative
        floatingpoint       [7]  IMPLICIT FloatingPoint,
        --                  [8]  is reserved
        octetstring         [9]  IMPLICIT OCTET STRING,
        visiblestring       [10] IMPLICIT VisibleString,
        generalizedtime     [11] IMPLICIT GeneralizedTime,
        binarytime          [12] IMPLICIT TimeOfDay,
        bcd                 [13] IMPLICIT INTEGER,
        booleanArray        [14] IMPLICIT BIT STRING,
        --objId             [15] IMPLICIT OBJECT IDENTIFIER,
        mMSString           [16] IMPLICIT MMSString,  -- unicode string
        utctime             [17] IMPLICIT UtcTime    -- UTC Time
        }

DataSequence ::= SEQUENCE OF Data

FloatingPoint ::= OCTET STRING

UtcTime ::= OCTET STRING

MMSString ::= UTF8String

TimeOfDay ::= OCTET STRING  -- (SIZE (4 | 6))

END
```

(A) ASN.1 Encoding



(B) Frames structure

Figure 4.9: goosePdu frame structure

- **sqNum**
  This Integer value has ranges of 0 to 4,294,967,295. The value of 0 is reserved for the first transmission of a StNum change. SqNum will increment for each transmission, but will rollover to a value of 1.

  The Tag associated with sqNum is set to 0x86.

- **test**
  This Boolean value has ranges of TRUE or FALSE.

  The Tag associated with test is set to 0x87.

- **ConfRev**
  This Integer value has ranges of 0 to 4,294,967,295.

  The Tag associated with ConfRev is set to 0x88.

- **ndsCom**
  This Boolean value has ranges of TRUE or FALSE.

  The Tag associated with ndsCom is set to 0x89.

- **numDatSetEntries**
  This Integer parameter specifies the number of members Data-Set in the GOOSE message.

  The Tag associated with numDatSetEntries is set to 0x8A.

- **allData**
  The last data in the sequence, is the dataSet that is going to be transmitted by means of GOOSE pdu.

  The Tag associated with allData is set to 0xAB.

  Any data to be sent in the dataSet of the goosePdu is encoded based on the MMS adapted ASN.1/BER encoding rule as listed in Table 4.4 [31].

goosePdu DataSet specification

Using the Routed-GOOSE service, the PMU/PDC data are transfered as the dataset of the goosePdu. The frame strucuture of the dataset defined in Section. 4.2, is illustrated in Figure. 4.10.

As it can be seen in the Figure. 4.10, the goosePdu dataset contains the ASN.1/BER encoded elements of the data (Tag-Length-Value). Based on the type, each component of data will be tagged according to Table. 4.4. For instance, the Floating-point type magnitude component of the complex value of Phase A Voltage in MMXU Loginal Node represented as `MMXU.PhV.PhsA.cVal.mag.f` is tagged with hexadecimal value of 0x87.

4.5.3 Sampled Value Protocol Data Unit (svPdu) specification

The svPdu is formed based on the ASN.1 Basic Encoding Rule. The complete detailed structure of the svPdu is illustrated in Fig. 4.11.

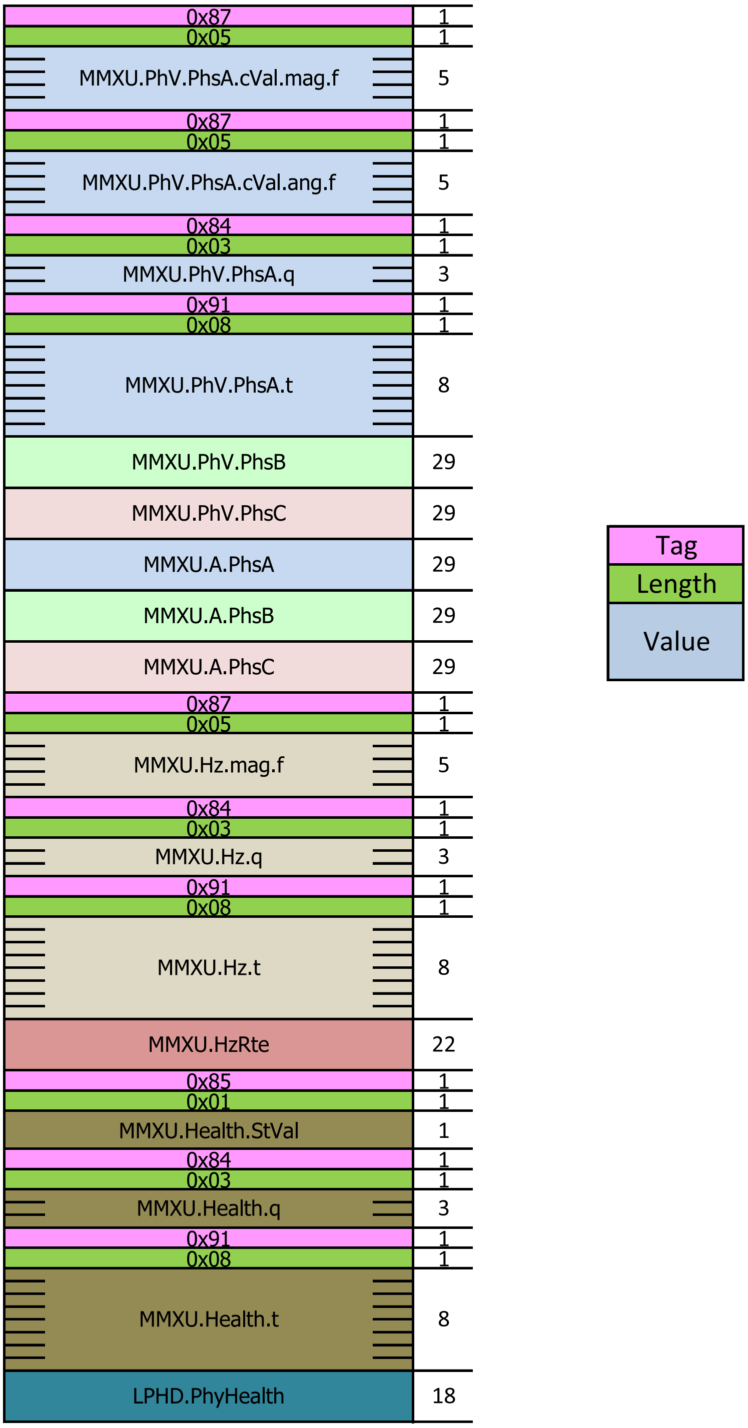
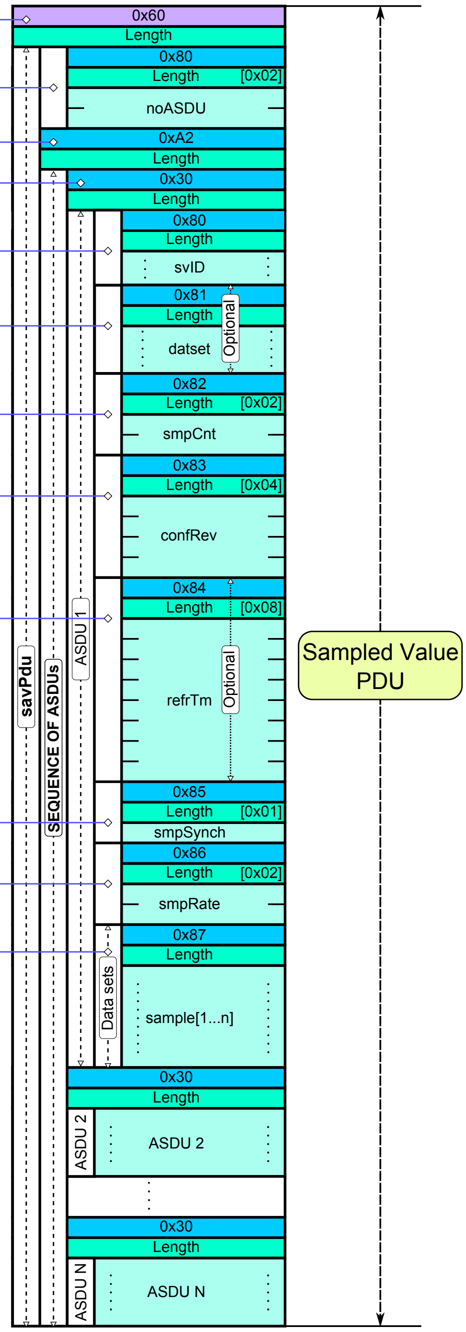


Figure 4.10: DataSet structure in goosePdu

```
IEC61850 DEFINITIONS ::= BEGIN

IEC 61850-9-2 Specific Protocol ::= CHOICE {
        savPdu              [APPLICATION 0] IMPLICIT SavPdu,
        }

SavPdu ::= SEQUENCE {
        noASDU              [0]  IMPLICIT INTEGER (1..65535),
        security            [1]  ANY OPTIONAL,
        asdu                [2]  IMPLICIT SEQUENCE OF ASDU,
        }
ASDU ::= SEQUENCE{
        svID                [0]  IMPLICIT VisibleString,
        datset              [1]  IMPLICIT VisibleString OPTIONAL,
        smpCnt              [2]  IMPLICIT OCTET STRING (SIZE(2)),
        confRev             [3]  IMPLICIT OCTET STRING (SIZE(4)),
        refrTm              [4]  IMPLICIT UtcTime OPTIONAL,
        smpSynch            [5]  IMPLICIT OCTET STRING (SIZE(1)),
        smpRate             [6]  IMPLICIT OCTET STRING (SIZE(2)),
        sample              [7]  IMPLICIT OCTET STRING (SIZE(n)),
        }
.
.
.
.

END
```



(A) ASN.1 Encoding (B) Frames structure

Figure 4.11: Sampled Value structure

Item	Data Type	MMS Tag (hex)
1	Array	0x81
2	Structure	0x82
3	Boolean	0x83
4	Bit-String	0x84
5	Integer	0x85
6	Unsigned	0x86
7	Floating-Point	0x87
8	Octet-String	0x89
9	Visible-String	0x8A
10	Timeofday	0x8C
11	BCD	0x8D
12	BooleanArray	0x8E

Table 4.4: Common MMS Data Value Encoding

The svPdu is encoded, with *Tag* value equal to 0x60. The Tag is followed by the *Length* indicating the whole length of the svPdu.

The *Data* part of the svPdu contains a sequence of data as listed below:

- **noASDU**
  Each Sampled Value (SV) packet contains one or multiple SV Application Service Data Unit (ASDU) encapsulated inside the SV APDU.

  This integer value gives the number of ASDUs which will be concatenated into one APDU.

  The Tag associated with noASDU is set to 0x80.

- **Sequence of ASDUs**
  All the ASDUs associated to the svPdu are put in Value part of a T-L-V encoding whose Tag value is set to the hexadecimal value of 0xA2.

  Then each ASDU has as series of data which is described below:

  - **svID**
    This Visible-String value, contains the system-wide unique identification of the ASDU.

    The Tag associated with datSet is set to 0x80.
  - **dataset**
    This optional Visible-String value, contains the value of the DataSetReference, as found in the Sampled Value control block.

The Tag associated with datSet is set to 0x81.

- **smpCnt**

  This fixed 2-byte size Unsigned Integer value, contains the value of Sample count which will be incremented each time a new sampling value is taken.

  The Tag associated with datSet is set to 0x82.

- **ConfRev**

  This 32-bit Unsigned Integer value, indicates the count of configuration changes regard to Sampled Value Control Block.

  The Tag associated with ConfRev is set to 0x83.

- **refrTm**

  This *optional* data is the 8-byte UTC Time TIMESTAMP of the refresh time of the SV Buffer.

  The Tag associated with TIMESTAMP is set to 0x84.

- **smpSynch**

  This Boolean value has ranges of TRUE or FALSE. If TRUE, indicates that SV are synchronized by a clock signal.

  The Tag associated with test is set to 0x85.

- **smpRate**

  This fixed 2-byte size Unsigned Integer value, contains the value of Sample rate.

  The Tag associated with datSet is set to 0x86.

- **Sample**

  The last data in the sequence, is the Data-Set that is going to be transmitted by means of svPdu.

  The Tag associated with Sample is set to 0x87.

svPdu DataSet specification

Unlike the goosePdu, the dataset of the svPdu is not encoded. In this case, the dataset components are transfered consecutively in their basic forms. The svPdu datset specification for transfer of a PMU data defined in Section. 4.2, based on the guidelines presented in [32], is illustrated in Figure. 4.12.

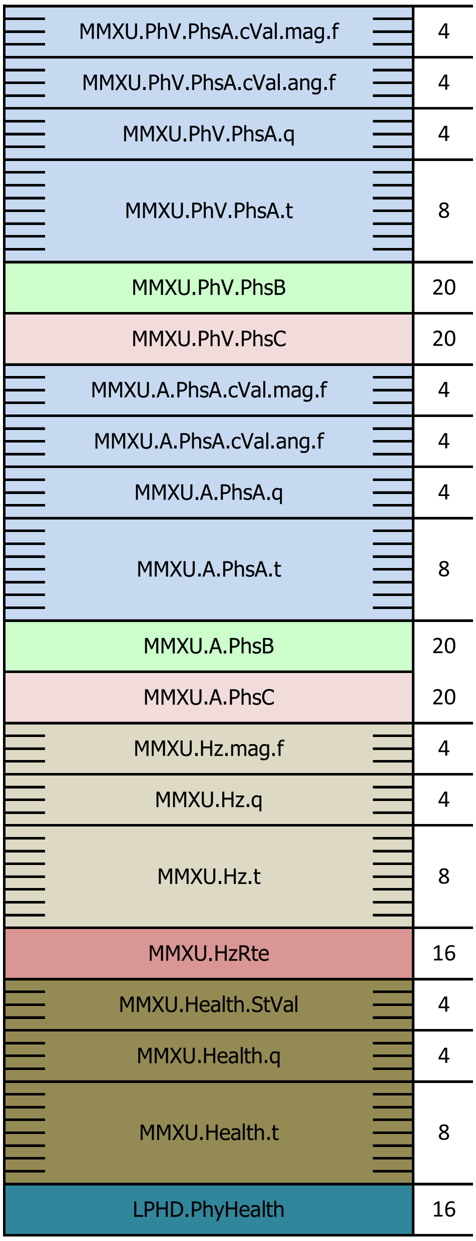


Figure 4.12: DataSet structure in svPdu

5 Implementation and Results

As mentioned in chapter 1, in this project "Khorjin" library is developed to act as an IEEE-IEC Gateway. Khorjin library is designed in such a way to receive synchrophasor data from PMU/PDC based on IEEE C37.118, map synchrophasor data to IEC 61850 data model, and publish IEC 61850-90-5 messages in either Routed-SV or Routed-GOOSE format.

Another part in the Khorjin library is developed to receive and parse the Routed-Sampled Value and Routed-GOOSE messages and provide the extracted raw synchrophasor data to the subscribed applications.

Hence, the Khorjin library is designed and implemented in two parts to be utilized on (1) Gateway and (2) Receiver platforms. The general design of the Khorjin library is depicted in Figure 5.1.

In the process of the design and development of this library, the open source "libiec61850" library has been analyzed and interpreted, which has provided a great tool for understanding and learning the structure of the IEC 61850 standard. The "libiec61850", developed by Michael Zillgith provides a server and client library for IEC 61850/MMS and IEC 61850/GOOSE communication protocols [33].

In order not to re-invent the wheel, some source codes developed in [33], and some other source codes developed by other people and used similarly in [33] are used in the implementation of the Khorjin library. For instance, the *mms_value.c*, *ber_encoder.c* and *ber_decoder.c* source codes are used in ASN.1 BER encoding and decoding of goosePdu datasets.

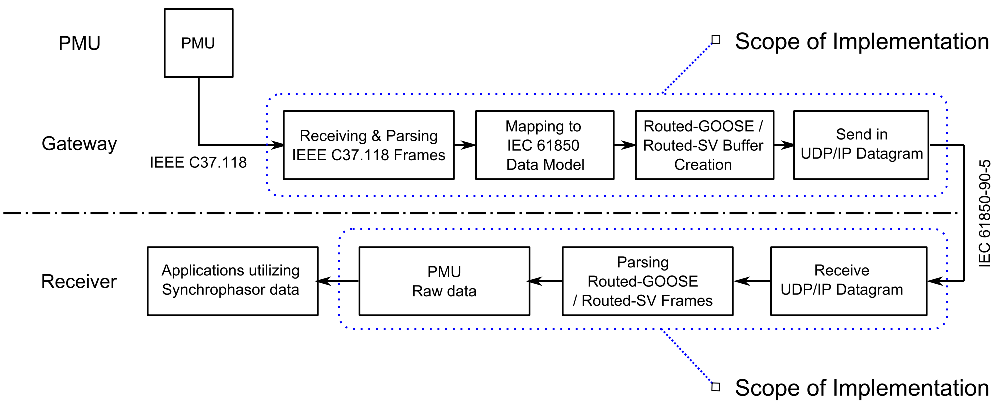


Figure 5.1: General structure of library utilization process

5.1 Design and implementation of Khorjin library

In this section, the Khorjin library design for both Gateway and Receiver parts is introduced.

5.1.1 IEEE-IEC Gateway

In the Gateway part of the Khorjin library, two sets of connection data are required as input:

1. PMU/PDC-Gateway Connection

   In order to establish a TCP/IP connection with a PMU/PDC, (1)*IP address*, (2) *Port number* and (3) *ID Code* information of the PMU/PDC server is the entry data to the library.

2. Gateway-Receiver Connection

   The UDP/IP connection for publishing the IEC 61850-90-5 Routed-SV or Routed-GOOSE messages, is established with input data of (1) *Receiver IP address* (In case of Unicast UDP/IP), (2) *Port number*(In the [3], it is set to 102) and (3) *APPID*.

The Gateway part of the Khorjin library can be divided into three components of:

- Generation and parsing of IEEE C37.118.2 messages
- Mapping to IEC 61850 data model
- IEC 61850-90-5 R-SV / R-GOOSE traffic generation

In the following, these components are explained in more detail.

IEEE C37.118.2 message generation and parsing

In this part of Gateway, having the connection information of the PMU/PDC, first, an object of `C37_118client` (a pointer to the `sC37_118client` structure defined below) is created and second, a TCP/IP connection is established between PMU/PDC and the Gateway.

```
struct sC37_118client{
        uint8_t* bufferSnd;
        uint8_t* bufferRcv;
        Socket socket;
        uint16_t idcode;
        uint16_t numPmu;
        uint32_t timeBase;
        uint16_t dataRate;
        Time configTimeStamp;
        Time headerTimeStamp;
        Time dataTimeStamp;
        Configuration configuration;
        bool running;
};

typedef struct sC37_118client* C37_118client;
```

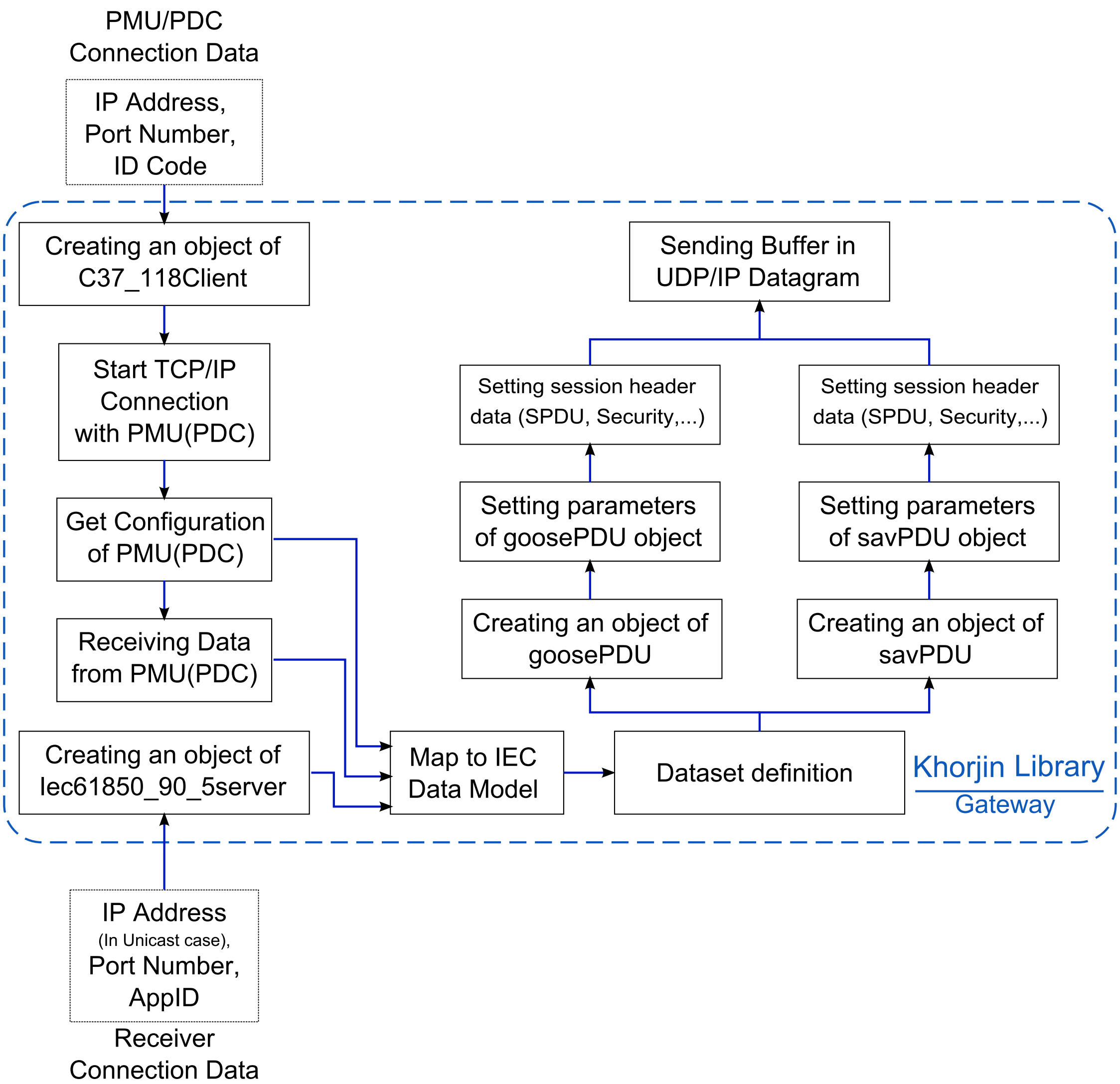


Figure 5.2: Khorjin library design - Gateway

As illustrated in Fig. 5.2, after successful connection with PMU/PDC, a "Stop Data Transmission" command message is generated and sent to the PMU/PDC to ensure that no Data message is streamed from PMU/PDC.

Then a "Send Configuration Type 2" command message is sent to the PMU/PDC. In response, by parsing the received CFG-2 message, the relevant configuration variables of the `C37_118client` object are defined.

Having the configuration data of the PMU/PDC, by sending a "Start Data Transmission" command message from Gateway, the PMU/PDC data message streaming is started.

Upon receiving each Data message, it will be parsed and mapped to IEC 61850 data model as already shown in Fig. 4.2

The API functions of Khorjin library for the IEEE C37.118.2 data exchange is presented in Annex B.1.

Mapping to IEC 61850 data model

To describe more in detail the process introduced above, it should be mentioned that upon receiving and parsing the PMU configuration message type 2, an object of `Iec61850_90_5Message` (a pointer to the below structure) is created. Then the configuration data of PMU enable the source code, to define the IEC 61850 data model of the PMU.

```
struct sIec61850_90_5Message{
        uint8_t* buffer;
        UdpSocket socket;
        uint8_t sessionIdentifier;
        uint32_t spduLength;
        uint32_t spduNumber;
        uint16_t versionNumber;
        SecurityInformation securityInformation;
        uint32_t userInformationLength;
        int spduLengthBufferPosition;
        uint8_t payLoadType;
        bool simulation;
        int simulationBufferPosition;
        uint16_t appID;
        uint16_t apduLength;
        int apduLengthBufferPosition;
        LinkedList dataset;
        SampledValuePDU sampledValuePDU;
        GoosePDU goosePDU;
        MmsValue* dataSetTimeStamp;
        MmsValue* dataSetQuality;
        MmsValue* mmxuHealth;
        MmsValue* mmxuMode;
        MmsValue* lphdPhyHealth;
        MmsValue* stat;
};

typedef struct sIec61850_90_5Message* Iec61850_90_5Message;
```

R-SV / R-GOOSE traffic generation

Running `c37_118client_getDataStream` function in a `while` loop, data streams are received and parsed. For the defined datset (containing the synchrophasor data) in the `Iec61850_90_5Message` object, in either cases of using Sampled Value or GOOSE services, an object of svPdu or goosePdu is created.

Using the API functions presented in Annex B.2 and B.3, the Sampled Value PDU / GOOSE PDU parameters are set.

In the next step, the session header information are written to the buffer. The following functions are defined to set the session header variables.

```
void
iec61850_90_5Message_setSPDUNumber(Iec61850_90_5Message self,
uint32_t spduNumber);

void
iec61850_90_5Message_setSimulation(Iec61850_90_5Message self,
bool status);

void
iec61850_90_5Message_setSecurityInformation(Iec61850_90_5Message self,
uint16_t timeToNextKey, uint16_t securityAlgorithm, uint32_t keyId);
```

Having the buffer prepared, based on the input connection data for IEC 61850-90-5 traffic generation part, using the `c37_118client_startTransmission` function, the R-SV / R-GOOSE message is sent through UDP/IP datagram.

```
int
iec61850_90_5Message_Send(C37_118client c37_118Object,
Iec61850_90_5Message iec61850_90_5Object);
```

The implementation of the Security part of the session layer is out of scope of this thesis and could be developed in the next versions of the library.

5.1.2 IEC 61850-90-5 R-GOOSE traffic Receiver

In the Routed-GOOSE traffic parsing part of Khorjin, at first, an object of `RoutedGOOSEReceiver` is created with input parameters of (1) PortNumber and (2) APPID.

In addition, the dataset of goosePdu is defined in an object of `RoutedGOOSESubscriber`. This subscriber object is then linked to the `RoutedGOOSEReceiver` object.

After this initialization, a UDP socket is created and a receiver loop continuously listen to receive UDP messages.

Upon receiving a UDP datagram, the word `Payload Type` of the session header will be checked to verify if the received frame is a R-GOOSE APDU. In case of `Payload Type` being equal to hexadecimal value of 0xA1, the R-GOOSE message is detected and the received buffer is parsed as a R-GOOSE message. Then, `APPID` of the received R-GOOSE message will be checked to be equal to the input APPID in the `RoutedGOOSEReceiver` object, otherwise the received frame will be discarded.

API functions are defined to process the security information words in the session header, however, the implementation of the security algorithm in the session layer is not in the scope of the project and further functions could be developed in future to add the security functionality.

Other API functions are defined to get common goosePdu parameters of `goCbRef`, `timeAllowedToLive`, `datSetRef`, `goID`, `stNum`, `sqNum`, `test`, `confRev` and `ndsCom`. Dataset of goosePdu is then decoded and raw PMU data is provided to subscribed applications.

The general design of the R-GOOSE traffic parsing of Khorjin is presented in Figure. 5.3.

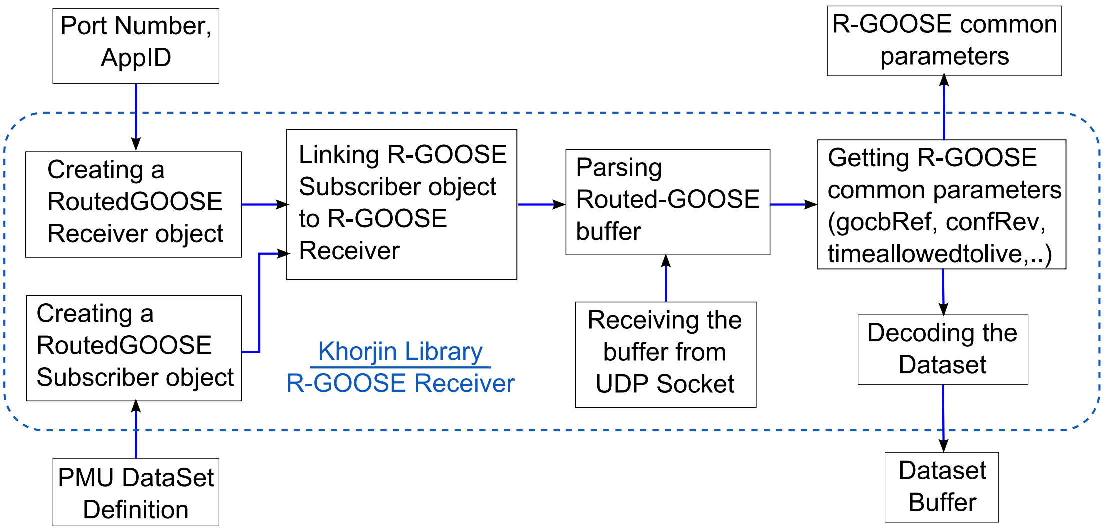


Figure 5.3: Routed-GOOSE traffic parsing algorithm

5.1.3 IEC 61850-90-5 R-SV traffic Receiver

In the Routed-Sampled Value receiver part of Khorjin, similar to R-GOOSE receiver, at first an object of `RoutedSVReceiver` is created and input parameters of (1) PortNumber and (2) APPID are defined in this object.

In addition, the dataset of svPdu is defined by creation of an `RoutedSVSubscriber` object. This subscriber object is then linked to the `RoutedSVReceiver` object.

After this initialization,a UDP socket is created and a receiver loop will be in continuous listening mode to receive the UDP messages.

Upon receiving a UDP datagram, the word `Payload Type` of the session header will be checked to verify if it is a R-SV APDU. In case of `Payload Type` being equal to hexadecimal value of 0xA2, the R-SV message is detected and the received buffer is parsed as a R-SV message. Then, `APPID` of the received R-SV message will be checked to be equal to the input APPID in the `RoutedSVReceiver` object, otherwise the received message will be discarded.

API functions are defined to get the security related words of the session header, however, since the implementation of the security control of the frames is not in the scope of the project, further functions could be developed in future to add the security check functionality.

Other API functions are defined to get common svPdu parameters of `svID`, `datSetRef`, `smpCnt`, `confRev`, `smpSynch`, `smpRate`. At the end, the raw PMU data is provided to subscribed applications by means of parsing the dataset of svPdu.

The general design of the R-GOOSE traffic parsing of Khorjin is presented in Figure. 5.4.

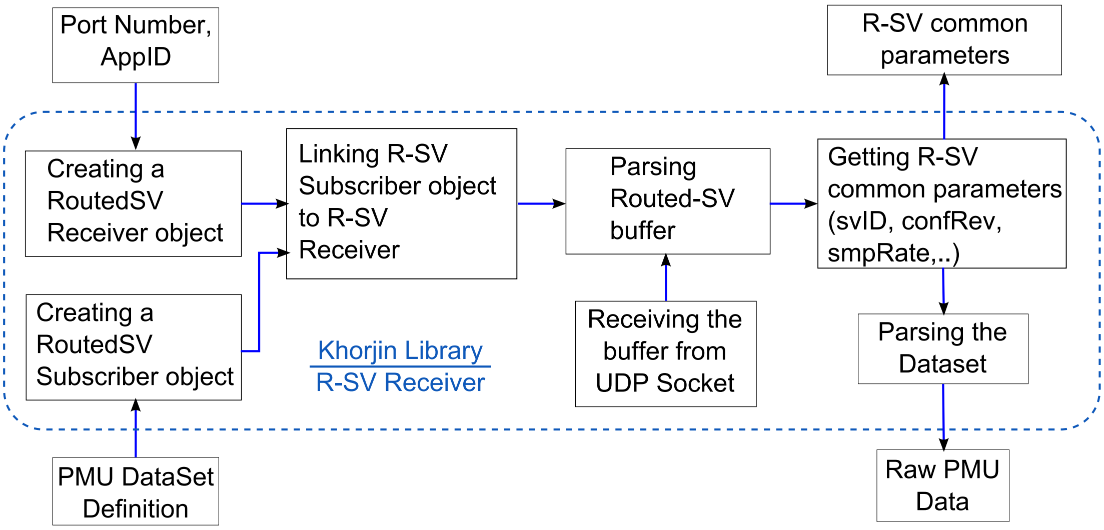


Figure 5.4: Routed-SV traffic parsing algorithm

5.2 RT-HIL validation of the functionality of Khorjin

In order to validate the functionality of Khorjin, testing was performed using the HIL setup in Figure. 5.5, and two separate computers where Khorjin is used either as Gateway and Receiver.

5.2.1 HIL Real-Time Simulation Setup

This section describes the real-time Hardware-in-the-Loop simulation setup in which the Khorjin library is validated. As shown in Figure. 5.5, a measurement location has been specified on a grid model that is simulated by the OPAL-RT real-time simulator. The measured voltages and currents are fed to PMU through the analogue output ports of the OPAL-RT simulator.

As indicated in the Figure. 5.5, the PMU used in this setup is Compact Reconfigurable IO systems (CRIO) from National Instruments Corporation [34]. As Figure. 5.5 shows, the signals from RT simulator are passed through the current amplifiers before being fed to the PMU. Syncrophasors are then sent to a PDC which streams the data over TCP/IP to the workstation computer holding IEEE-IEC Gateway. On another workstation the Receiver part of Khorjin receives the real-time streams of data in IEC 61850-90-5 format and parses the R-GOOSE or R-SV messages.

In order to verify the functionality of Khorjin, following results are captured and evaluated from the simulation setup.

- Screenshots of (1) CRIO PMU web interface, (2) IEEE-IEC Gateway and (3) Receiver
- PDC archive of PMU data streams

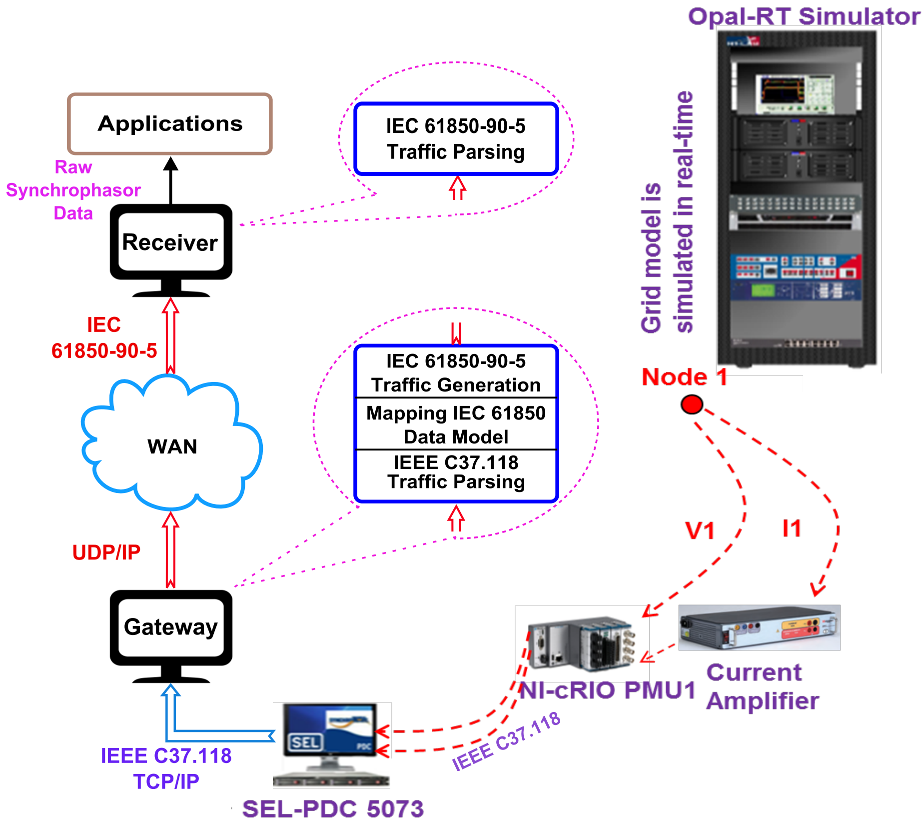


Figure 5.5: Hardware In the Loop setup

- Wireshark captures of (1) IEEE C37.118.2 data exchanged between PDC and Gateway and (2) IEC 61850-90-5 data transferred between Gateway and Receiver.

These results are captured in both R-GOOSE and R-SV traffic generation validation tests.

5.2.2 R-GOOSE traffic generation and parsing validation test

In this test, the IEEE-IEC Gateway part of Khorjin is set to transfer the received PMU data using the Routed-GOOSE protocol.

PDC archive of PMU data streams

Upon running the Gateway, as it is described in previous sections, a TCP/IP connection is established between Gateway and PDC. Then, after receiving and parsing the configuration data message, the Gateway starts to receive the continuous streams of the data messages. PDC is capable of recording the streamed PMU data. For the purpose of capturing the results, while the PMU data transfer was in continuous real-time streaming status, the IEEE-IEC Gateway was paused and simultaneously the screenshots of the three computers running (1) CRIO PMU web interface, (2) Gateway and (3) Receiver were captured. The PMU

Timestamp	GIZMO:Va: Magnitude	GIZMO:Va: Angle	GIZMO:Vb: Magnitude	GIZMO:Vb: Angle	GIZMO:Vc: Magnitude	GIZMO:Vc: Angle	GIZMO:Ia: Magnitude	GIZMO:Ia: Angle	GIZMO:Ib: Magnitude	GIZMO:Ib: Angle	GIZMO:Ic: Magnitude	GIZMO:Ic: Angle	
11:28.5	0.9998	86.4399	0.9995	-33.5626	0.9996	-153.5716	1.7509	116.1072	1.2867	-134.9468	1.2893	-15.0187	
11:28.5	0.9998	86.4437	0.9995	-33.5624	0.9996	-153.5682	1.7512	116.0958	1.287	-134.9585	1.2896	-15.0452	PMU Screenshot
11:28.5	0.9998	86.4423	0.9995	-33.5664	0.9995	-153.5624	1.7514	116.0981	1.2872	-134.9743	1.2904	-15.0603	
11:28.5	0.9998	86.4397	0.9995	-33.5622	0.9996	-153.5686	1.7515	116.0889	1.2876	-134.9881	1.2908	-15.0648	
11:28.5	0.9998	86.4345	0.9995	-33.5647	0.9996	-153.5735	1.7513	116.076	1.2878	-134.998	1.2907	-15.0649	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
11:29.0	0.9997	86.4048	0.9996	-33.5994	0.9995	-153.6025	1.7456	116.3153	1.2781	-134.646	1.2814	-14.7192	
11:29.1	0.9996	86.406	0.9995	-33.6014	0.9995	-153.6038	1.7453	116.3138	1.2782	-134.6411	1.2811	-14.7131	
11:29.1	0.9997	86.4035	0.9997	-33.6023	0.9997	-153.6012	1.7451	116.3151	1.2783	-134.6366	1.281	-14.7041	
11:29.1	0.9997	86.4011	0.9997	-33.602	0.9997	-153.6015	1.7453	116.315	1.2784	-134.6444	1.2813	-14.7118	Gateway & Receiver Screenshot
11:29.1	0.9997	86.4005	0.9995	-33.6057	0.9996	-153.606	1.7456	116.3115	1.2784	-134.6533	1.2816	-14.7273	

Figure 5.6: PDC archive of PMU data streams in R-GOOSE traffic generation test

data streams recorded by PDC archive, (at the time of pausing Gateway and capturing the screenshots) in R-GOOSE traffic generation test is shown in Figure. 5.6.

The stream highlighted in green color is the stream of PMU data captured in the PMU web interface screenshot shown in Figure. 5.7.

The stream highlighted in blue color is the stream of PMU data sent by PDC at the time of pausing the Gateway. This data that is analyzed in Figures. 5.8 and 5.9 is parsed and re-transmitted by the Gateway and similarly received and parsed at the Receiver.

The conformance of performance of Khorjin with IEEE C37.118.2 and IEC 61850-90-5 protocols is verified by means of analyzing the screenshots shown in Figure. 5.8 and Wireshark captures illustrated in Figure. 5.9.

IEEE-IEC Gateway and Receiver screenshots

The SEL-PDC 5073 Phasor Data Concentrator (PDC), utilized in this test, is streaming PMU data based on IEEE C37.118 protocol. In this regard, (1) Successful connection of the Gateway to this PDC, (2) Sending command messages to the PDC and receiving correct response from PDC and (3) Receiving and parsing of configuration and data messages and printing the correct PMU data are indication of conformance of the functionality of Khorjin with IEEE C37.118.2 protocol.

In Figure. 5.8.A, it is shown that the PMU data highlighted in blue color in Figure. 5.6, is correctly parsed on the computer running the IEEE-IEC Gateway part of Khorjin.

Figure. 5.8.B shows the parsed IEC 61850-90-5 R-GOOSE message at the Receiver side. It can be seen that in addition to some of general goosepdu information i.e. stNum and sqNum, the PMU data depicted in Figure. 5.8.A is correctly received at the Receiver and printed as the sequence of data in goosePdu dataset described in Section. 4.2

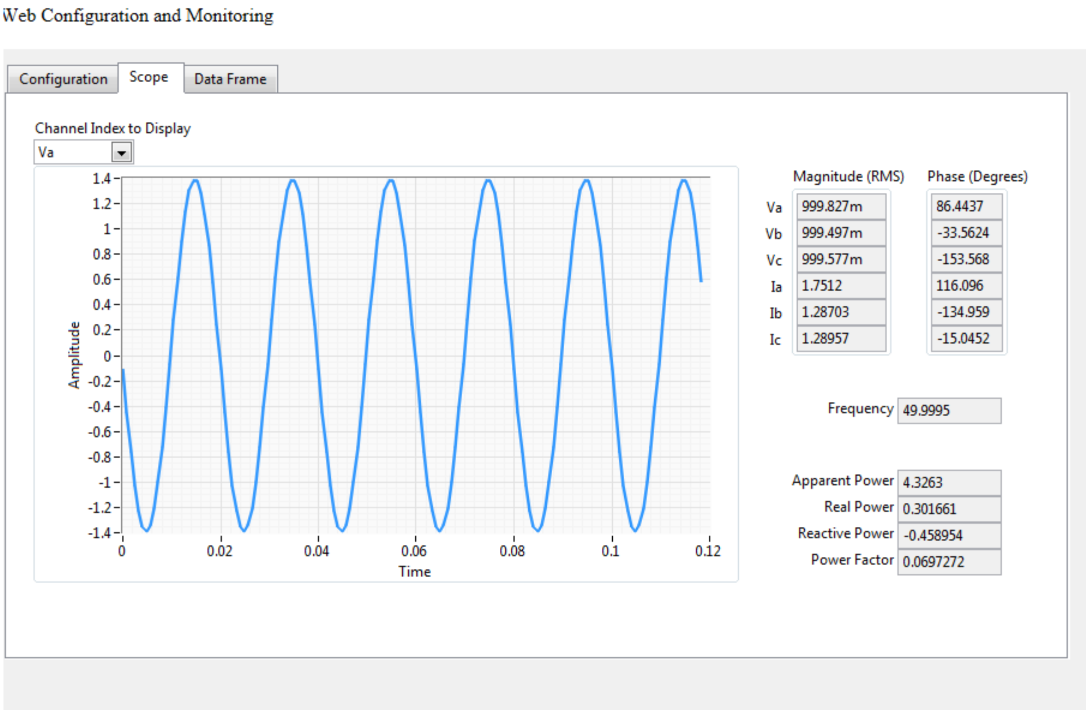


Figure 5.7: Screenshot of compactRIO PMU web interface in R-GOOSE traffic generation test

Wireshark capture analysis of exchanged traffic

As described in previous section, the conformance of the Gateway part of the Khorjin with IEEE C37.118.2 is verified by successful connection and communication with the PDC. However, verification of conformance with requirements of the IEC 61850-90-5 is done by analyzing the frames captured by Wireshark network protocol analyzer software [35].

Figure. 5.9.A shows the captured TCP/IP frame transmitting the IEEE C37.118.2 data message containing the PMU stream shown in Figure. 5.8.A. In this frame, the Phasor data is transmitted in floating-point and rectangular format as described in Section. 3.2.2.

It can be seen that the data message captured in Figure. 5.9.A is consistent with the IEEE C37.118.2 data message introduced in Section 3.2.2 and its frame specification depicted in Figures. 3.1 and 3.4. The data message started by common words of: (1) SYNC, (2) FRAMESIZE, (3) IDCODE, (4) SOC, (5) FRACSEC and ended by (21) CHK frame. The frames specific to data messages are : (6) STAT, (7) PHASOR 1 (Real), (8) PHASOR 1 (Imag), ... , (19) FREQ and (20) DFREQ.

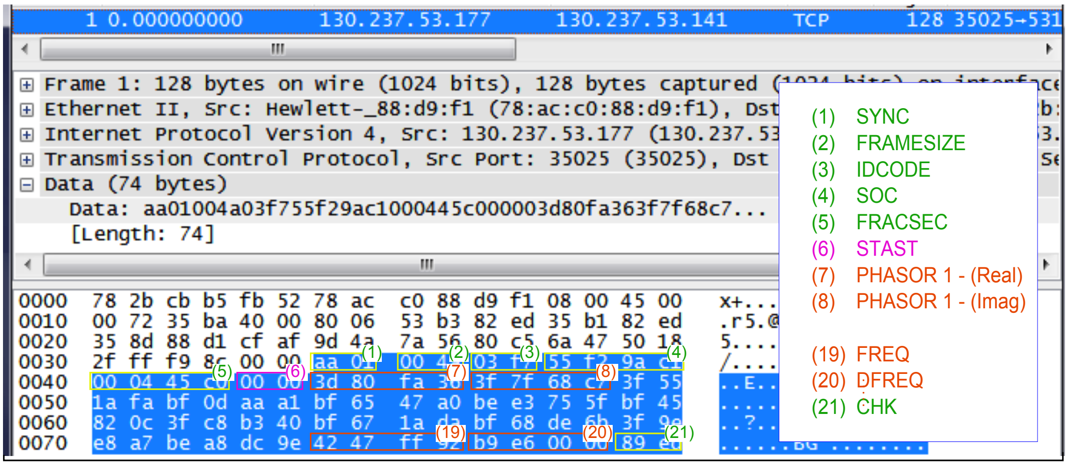
```
Data Frame Time Stamp: quality ->Leap Second Occured = FALSE
Data Frame Time Stamp: quality ->Leap Second Pending = FALSE
Data Frame Time Stamp: quality ->Time is Locked =      TRUE
Data Frame Time Stamp: quality ->Time error =          0.000000

Data Frame Time Stamp: timeValueInMs =                 1441962689100      (5)
Data Time Stamp: 2015-9-11 , 11:11:29.100000

Data of PMU No.1 :
=====================

Status of Data Frame:
---------------------

1. Data Error:                     GOOD MEASUREMENT DATA, NO ERRORS
2. Is PMU syncronised with a UTC:  TRUE
3. PMU data sorting metrice:       TIME STAMP
4. Is PMU triggered:               FALSE                                 (1)
5. Is PMU configuration to change: FALSE
6. Is PMU data modified:           FALSE
7. PMU Time Quality:               NOT USED
8. PMU Unlocked time:              SYNC LOCKED OR UNLOCKED <10s <BEST QUALITY>

Values of Data Frame:
---------------------

Frequency of PMU 1 = 49.999992
ROCOF      of PMU 1 = 0.000000

Phasor 1 of PMU 1 -> Channel Name = Va
Phasor 1 of PMU 1 -> Magnitude    = 0.999710    (2)
Phasor 1 of PMU 1 -> Angle        = 86.401123   (3)

Phasor 2 of PMU 1 -> Channel Name = Vb
Phasor 2 of PMU 1 -> Magnitude    = 0.999679
Phasor 2 of PMU 1 -> Angle        = -33.601955

Phasor 3 of PMU 1 -> Channel Name = Vc
Phasor 3 of PMU 1 -> Magnitude    = 0.999703
Phasor 3 of PMU 1 -> Angle        = -153.601456

Phasor 4 of PMU 1 -> Channel Name = Ia
Phasor 4 of PMU 1 -> Magnitude    = 1.745339
Phasor 4 of PMU 1 -> Angle        = 116.315025

Phasor 5 of PMU 1 -> Channel Name = Ib
Phasor 5 of PMU 1 -> Magnitude    = 1.278358
Phasor 5 of PMU 1 -> Angle        = -134.644440

Phasor 6 of PMU 1 -> Channel Name = Ic
Phasor 6 of PMU 1 -> Magnitude    = 1.281345
Phasor 6 of PMU 1 -> Angle        = -14.711807
```

(A) IEEE-IEC Gateway

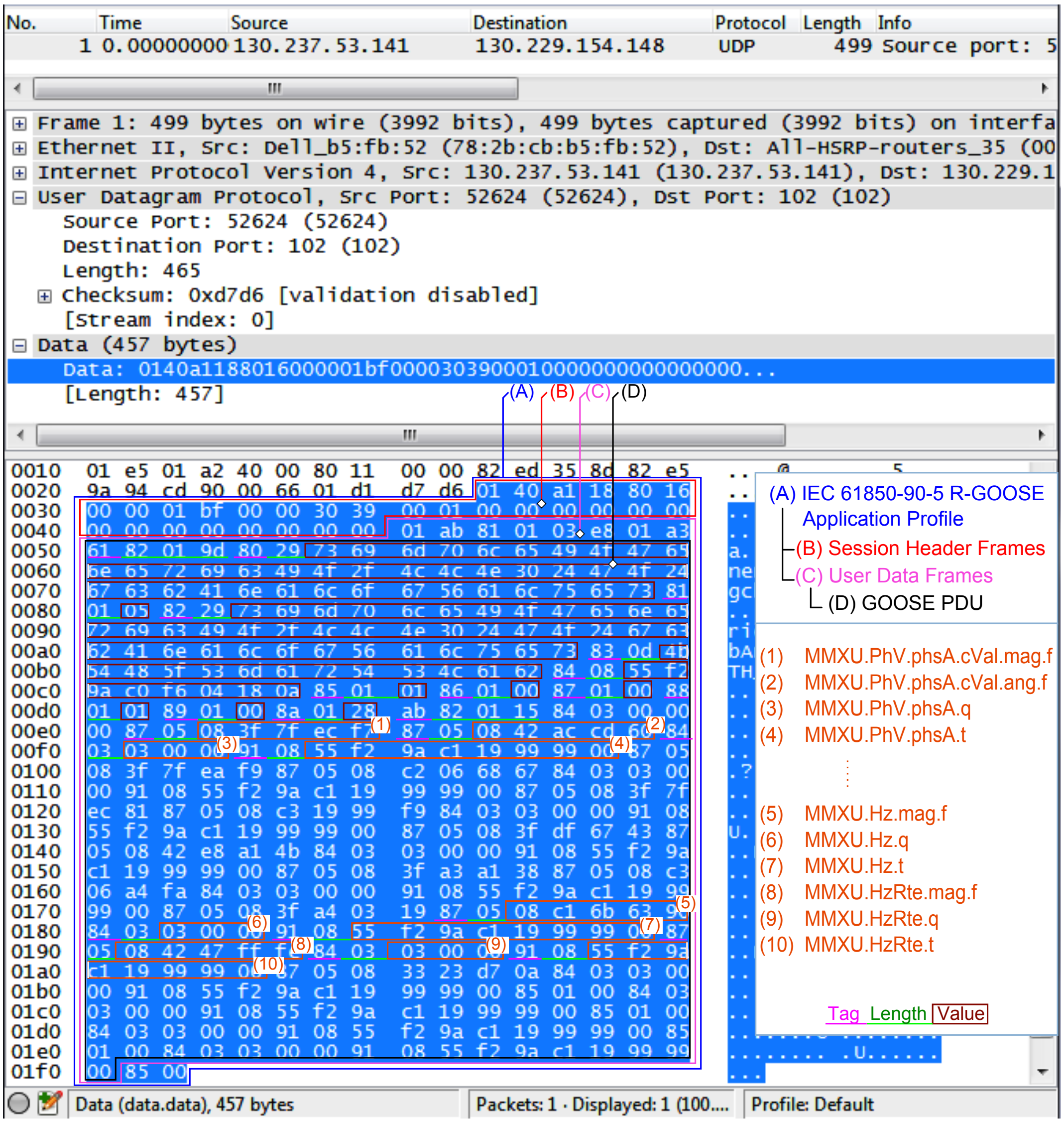
```
Iec61850_90_5 Routed GOOSE Message:
  stNum: 1 sqNum: 0
  timeToLive: 5 (1)        (2)        (3)           (4)                   (5)
  timestamp: 1441962688961
{0000000000000000,0.999710,86.401123,0000000000000,20150911091129.100Z,0.999679,
-33.601955,0000000000000,20150911091129.100Z,0.999703,-153.601456,0000000000000,
20150911091129.100Z,1.745339,116.315025,0000000000000,20150911091129.100Z,1.2783
58,-134.644440,0000000000000,20150911091129.100Z,1.281345,-14.711807,00000000000
00,20150911091129.100Z,49.999992,0000000000000,20150911091129.100Z,0.000000,0000
000000000,20150911091129.100Z,0,0000000000000,20150911091129.100Z,0,000000000000
0,20150911091129.100Z,0,0000000000000,20150911091129.100Z}
```

(B) IEC 61850-90-5 R-GOOSE Receiver

Figure 5.8: Screenshots of (A) IEEE-IEC Gateway and (B) Receiver in Routed-GOOSE traffic generation test



(A) IEEE C37.118 Data Message



(B) IEC 61850-90-5 R-GOOSE Message

Figure 5.9: Wireshark capture analysis of (A) IEEE C37.118.2 Data message and (B) IEC 61850-90-5 R-GOOSE message transferring the same PMU data

For instance, using the IEEE floating-point converter available in [36], the real and imaginary components of PHASOR 1 value in Figure. 5.9.A are calculated and compared with the value printed in Figure. 5.8.A:

$$Phasor1(Real) = 0 \text{ x } \{\text{3D 80 FA 36}\} = 0.0629772$$
$$Phasor1(Imag) = 0 \text{ x } \{\text{3F 7F 68 C7}\} = 0.9976925$$

$$Phasor1 = 0.0629772 + j0.9976925$$

is equivalent to:

$$MMXU.Phv.phsA.cVal.mag.f = 0 \text{ x } \{\text{3F 7F EC F7}\} = 0.99970955$$
$$MMXU.Phv.phsA.cVal.ang.f = 0 \text{ x } \{\text{42 AC CD 60}\} = 86.40112$$

$$MMXU.Phv.phsA.cVal = 0.99970955\angle 86.40112$$

The IEC 61850-90-5 Routed-GOOSE message specification is described in Section. 4.5. The session layer structure is illustrated in Figures .4.7, and the specification of goosePdu encapsulated in session layer is shown in Figure. 4.9. The goosePdu dataset holding the synchrophasor data has the structure presented in Figure. 4.10.

The detailed analysis of R-GOOSE message shown in Figure. 5.9.B, confirms its conformity with the specification defined in IEC 61850-90-5 standard. In Figure. 5.9.B, the bytes in blue background color constitute the complete R-GOOSE session layer. The range of bytes assigned as (B) show the session header frames and the bytes assigned as (C) are the user data of session layer. The frames marked as group (D) are the goosePdu bytes.

The group of bytes marked as (B) in Figure. 5.9.B contains the session header frames of : (1) *LI* (0x01), (2) *TI* (0x40), (3) *Session Identifier* (0xA1), (4) *Length Identifier* (0x18), (5) *Common Header* (0x80), (6) *Length Identifier* (0x16), (7) *SPDU Length* (0x{0x00 00 01 BF}), (8) *SPDU Number* (0x{00 00 30 39}), (9) *Version Number* (0x{0x00 01}), (10) *TimeOfCurrentKey* (0x{00 00 00 00}), (11) *TimeToNextKey* (0x{00 00}), (12) *Security Algorithms* (0x{00 00}), (13) *KeyID* (0x{00 00 00 00}).

The frames marked as group (C) in Figure. 5.9.B, starts with: (1) *User Data Length* (0x{00 00 01 AB}), (2) *Payload Type* (0x{81}), (3) *Simulation* (0x{01}), (4) *APPID* (0x{03 E8}), (5) *APDU Length* (0x{01 A3}) and continued with the goosePdu bytes marked as group (D).

The goosePdu in Figure. 5.9.B, starts with goosePdu Tag (0x61) and its length field (0x{0x82 01 9D}), then followed by general goosePdu information in Tag-Length-Value format:

(1)	*GoCBRef*	(0x{80}-0x{29}-0x{73 ...}),
(2)	*timeAllowedtoLive*	(0x{81}-0x{01}-0x{05}),
(3)	*datSet*	(0x{82}-0x{29}-0x{73 ...}),
(4)	*goID*	(0x{83}-0x{0D}-0x{4B ...}),
(5)	*t*	(0x{84}-0x{08}-0x{55 ...}),
(6)	*stNum*	(0x{85}-0x{01}-0x{01}),
(7)	*sqNum*	(0x{86}-0x{01}-0x{00}),
(8)	*test*	(0x{87}-0x{01}-0x{00}),
(9)	*ConfRev*	(0x{88}-0x{01}-0x{01}),
(10)	*ndsCom*	(0x{89}-0x{01}-0x{00}),
(11)	*numDatSetEntries*	(0x{8A}-0x{01}-0x{28}),
(12)	*allData*	(0x{AB}-0x{82 01 15}-0x{84 ...}),

The *allData* field contains the goosePdu dataset in T-L-V encoded format. In this test, the 6 PHASORS (Va, Vb, Vc, Ia, Ib, Ic), FREQ and DFREQ are transferred within the dataset described in Figure. 4.10. For instance, the $MMXU.PhV.PhsA$ and $MMXU.Hz$ data objects in Figure. 5.9.B are described below:

(1)	*MMXU.PhV.PhsA.cVal.mag.f*	(0x{87}-0x{05}-0x{08 3F ...}),
(2)	*MMXU.PhV.PhsA.cVal.ang.f*	(0x{87}-0x{05}-0x{08 42 ...}),
(3)	*MMXU.PhV.PhsA.q*	(0x{84}-0x{03}-0x{03 00 00}),
(4)	*MMXU.PhV.PhsA.t*	(0x{91}-0x{08}-0x{55 F2 ...}),
(5)	*MMXU.Hz.mag.f*	(0x{87}-0x{05}-0x{08 C1 ...}),
(6)	*MMXU.Hz.q*	(0x{84}-0x{03}-0x{03 00 00}),
(7)	*MMXU.Hz.t*	(0x{91}-0x{08}-0x{55 F2 ...}),

Timestamp	GIZMO:Va: Magnitude	GIZMO:Va: Angle	GIZMO:Vb: Magnitude	GIZMO:Vb: Angle	GIZMO:Vc: Magnitude	GIZMO:Vc: Angle	GIZMO:Ia: Magnitude	GIZMO:Ia: Angle	GIZMO:Ib: Magnitude	GIZMO:Ib: Angle	GIZMO:Ic: Magnitude	GIZMO:Ic: Angle	
00:03.4	0.9996	130.9319	0.9996	10.9374	0.9995	-109.0734	1.7519	160.5655	1.2851	-90.4739	1.2901	29.5292	
00:03.5	0.9996	130.9352	0.9995	10.9234	0.9995	-109.0733	1.7521	160.5441	1.2857	-90.4946	1.2904	29.5111	
00:03.5	0.9997	130.9352	0.9995	10.9225	0.9995	-109.0733	1.7523	160.5305	1.2861	-90.5182	1.2907	29.4862	PMU Screenshot
00:03.5	0.9996	130.9316	0.9996	10.9313	0.9996	-109.0765	1.7528	160.5208	1.2863	-90.5269	1.291	29.4614	
00:03.5	0.9997	130.9253	0.9996	10.9283	0.9997	-109.0774	1.7532	160.5069	1.2867	-90.5353	1.2913	29.4478	
00:03.5	0.9998	130.9222	0.9995	10.9222	0.9996	-109.0756	1.7533	160.5019	1.287	-90.5437	1.2915	29.449	
00:03.6	0.9997	130.9258	0.9995	10.9247	0.9996	-109.0741	1.7533	160.5054	1.2869	-90.5408	1.2915	29.4562	
00:03.6	0.9997	130.9235	0.9995	10.9254	0.9995	-109.0759	1.7532	160.5014	1.287	-90.5428	1.2915	29.4511	
00:03.6	0.9998	130.9264	0.9995	10.9234	0.9995	-109.0766	1.753	160.4984	1.287	-90.5562	1.2917	29.4369	
00:03.6	0.9997	130.9272	0.9996	10.9254	0.9995	-109.0835	1.753	160.5017	1.2869	-90.5559	1.2917	29.4368	
00:03.6	0.9996	130.9227	0.9995	10.9225	0.9995	-109.0878	1.7529	160.5024	1.2869	-90.5463	1.2916	29.4549	
00:03.7	0.9997	130.9141	0.9995	10.9211	0.9996	-109.0841	1.7527	160.5048	1.2868	-90.5412	1.2915	29.4684	
00:03.7	0.9997	130.9197	0.9995	10.9197	0.9996	-109.0838	1.7526	160.5143	1.2862	-90.5203	1.2909	29.475	
00:03.7	0.9997	130.9258	0.9995	10.9177	0.9995	-109.0904	1.7524	160.5266	1.2856	-90.4951	1.2903	29.4945	
00:03.7	0.9997	130.9195	0.9995	10.9176	0.9995	-109.091	1.7519	160.5365	1.2854	-90.4912	1.29	29.5207	
00:03.7	0.9997	130.912	0.9995	10.9172	0.9995	-109.0887	1.7517	160.5445	1.2852	-90.4863	1.2896	29.5344	
00:03.8	0.9997	130.9122	0.9995	10.916	0.9996	-109.0957	1.7513	160.5648	1.2844	-90.4599	1.2889	29.5473	
00:03.8	0.9997	130.914	0.9995	10.916	0.9996	-109.0961	1.7508	160.5885	1.2834	-90.4286	1.2882	29.5664	Gateway & Receiver Screenshot
00:03.8	0.9997	130.9109	0.9995	10.9116	0.9995	-109.097	1.7505	160.6004	1.2829	-90.4012	1.2876	29.5878	
00:03.8	0.9997	130.9111	0.9995	10.9073	0.9995	-109.099	1.75	160.6141	1.2825	-90.3738	1.2869	29.6159	

Figure 5.10: PDC archive of PMU data streams in R-SV traffic generation test

5.2.3 R-SV traffic generation and parsing validation test

In this test, the IEEE-IEC Gateway part of Khorjin is set to transfer the received PMU data using the Routed-SV protocol.

PDC archive of PMU data streams

Similar to the R-GOOSE validation test, in R-SV test, the PMU data streams recorded by PDC archive, (at the time of pausing the Gateway and capturing the screenshots) is shown in Figure. 5.10.

The stream highlighted in green color is the stream of PMU data captured in the PMU web interface screenshot shown in Figure. 5.11.

The stream highlighted in blue color is the stream of PMU data sent by PDC at the time of pausing the Gateway. This data that is analyzed in Figures. 5.12 and 5.13 is parsed and re-transmitted by the Gateway and similarly received and parsed at the Receiver.

The conformance of performance of Khorjin with IEC 61850-90-5 R-SV protocol is verified by means of analyzing the screenshots shown in Figure. 5.12 and Wireshark captures illustrated in Figure. 5.13.

IEEE-IEC Gateway and Receiver screenshots

In Figure. 5.12.A, it is shown that the PMU data highlighted in blue color in Figure. 5.10, is correctly parsed on the computer running the IEEE-IEC Gateway part of Khorjin.

Figure. 5.12.B shows the parsed IEC 61850-90-5 R-SV message at the Receiver side. It can be seen that in addition to some of general svPdu information i.e. smpCnt and confRev, the

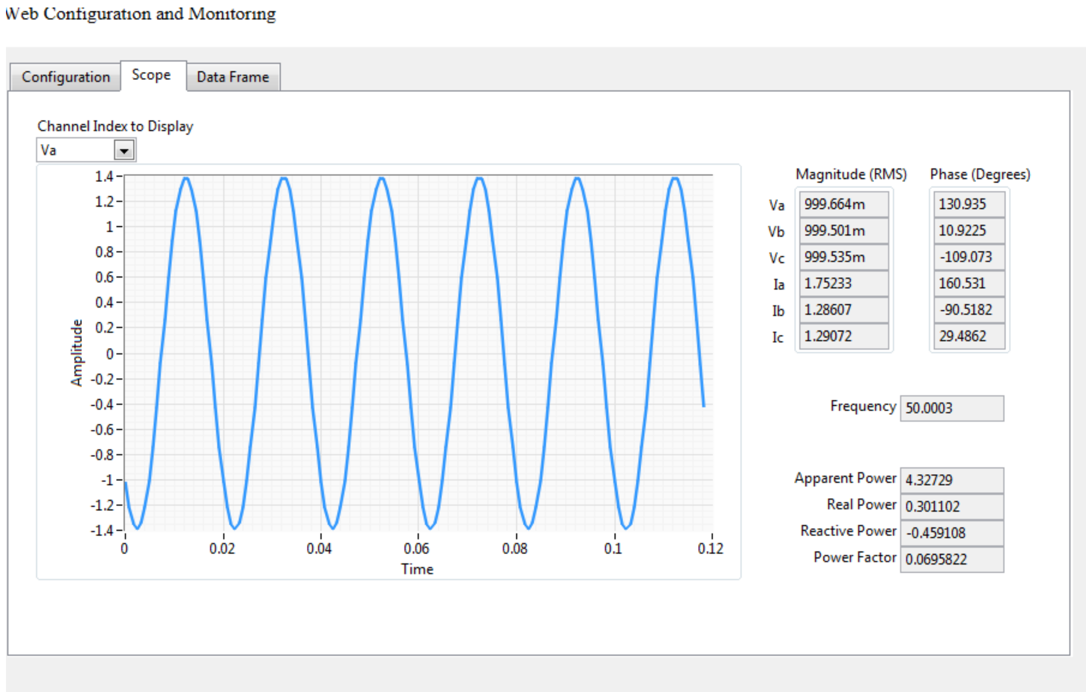


Figure 5.11: Screenshot of compactRIO PMU web interface in R-SV traffic generation test

PMU data depicted in Figure. 5.12.A is correctly received at the Receiver and printed as the sequence of data in svPdu dataset described in Section. 4.2.

Wireshark capture analysis of exchanged traffic

Figure. 5.13.A shows the captured TCP/IP frame transmitting the IEEE C37.118.2 data message containing the PMU stream shown in Figure. 5.12.A. In this frame, the Phasor data is transmitted in floating-point and rectangular format as described in Section. 3.2.2.

The data message captured in Figure. 5.13.A is consistent with the IEEE C37.118.2 data message introduced in Section 3.2.2 and its frame specification depicted in Figures. 3.1 and 3.4. The data message started by common words of: (1) SYNC, (2) FRAMESIZE, (3) IDCODE, (4) SOC, (5) FRACSEC and ended by (21) CHK frame. The frames specific to data messages are : (6) STAT, (7) PHASOR 1 (Real), (8) PHASOR 1 (Imag), ... , (19) FREQ and (20) DFREQ.

The IEC 61850-90-5 Routed-SV message specification is described in Section. 4.5. The session layer structure is illustrated in Figures .4.7, and the specification of svPdu encapsulated in session layer is shown in Figure. 4.11. The svPdu dataset holding the synchrophasor data has the structure presented in Figure. 4.12.

The detailed analysis of R-SV message shown in Figure. 5.13.B, confirms its conformity with the specification defined in IEC 61850-90-5 standard. In Figure. 5.13.B, the bytes in blue background color constitute the complete R-SV session layer. The range of bytes assigned as

```
-------------------------------------------------------------
Data Frame Time Stamp: quality ->Leap Second Occured = FALSE
Data Frame Time Stamp: quality ->Leap Second Pending = FALSE
Data Frame Time Stamp: quality ->Time is Locked =      TRUE
Data Frame Time Stamp: quality ->Time error =          0.000000

Data Frame Time Stamp: timeValueInMs =                 1441962003780
Data Time Stamp: 2015-9-11 , 11:0: 3.780000

Data of PMU No.1 :
=====================

Status of Data Frame:
---------------------

1. Data Error:                        GOOD MEASUREMENT DATA, NO ERRORS
2. Is PMU syncronised with a UTC:     TRUE
3. PMU data sorting metrice:          TIME STAMP
4. Is PMU triggered:                  FALSE
5. Is PMU configuration to change:    FALSE
6. Is PMU data modified:              FALSE
7. PMU Time Quality:                  NOT USED
8. PMU Unlocked time:                 SYNC LOCKED OR UNLOCKED <10s <BEST QUALITY>

Values of Data Frame:
---------------------

Frequency of PMU 1 = 49.999969
ROCOF     of PMU 1 = -0.000000

Phasor 1 of PMU 1 -> Channel Name = Va
Phasor 1 of PMU 1 -> Magnitude    = 0.999715
Phasor 1 of PMU 1 -> Angle        = 130.914047

Phasor 2 of PMU 1 -> Channel Name = Vb
Phasor 2 of PMU 1 -> Magnitude    = 0.999544
Phasor 2 of PMU 1 -> Angle        = 10.915986

Phasor 3 of PMU 1 -> Channel Name = Vc
Phasor 3 of PMU 1 -> Magnitude    = 0.999605
Phasor 3 of PMU 1 -> Angle        = -109.096130

Phasor 4 of PMU 1 -> Channel Name = Ia
Phasor 4 of PMU 1 -> Magnitude    = 1.750806
Phasor 4 of PMU 1 -> Angle        = 160.588501

Phasor 5 of PMU 1 -> Channel Name = Ib
Phasor 5 of PMU 1 -> Magnitude    = 1.283416
Phasor 5 of PMU 1 -> Angle        = -90.428551

Phasor 6 of PMU 1 -> Channel Name = Ic
Phasor 6 of PMU 1 -> Magnitude    = 1.288244
Phasor 6 of PMU 1 -> Angle        = 29.566444
```

(A) IEEE-IEC Gateway

```
Iec61850_90_5 Routed Sampled Value Message:
  smpCnt: 867 confRev: 1
          (1)          (2)          (3)           (4)                 (5)
{0000000000000000,0.999715,130.914047,0000000000000,20150911090003.780Z,0.999544
,10.915986,0000000000000,20150911090003.780Z,0.999605,-109.096130,0000000000000,
20150911090003.780Z,1.750806,160.588501,0000000000000,20150911090003.780Z,1.2834
16,-90.428551,0000000000000,20150911090003.780Z,1.288244,29.566444,0000000000000
,20150911090003.780Z,49.999969,0000000000000,20150911090003.780Z,-0.000000,00000
00000000,20150911090003.780Z,0,0000000000000,20150911090003.780Z,0,0000000000000
,20150911090003.780Z,0,0000000000000,20150911090003.780Z}
```

(B) Receiver

Figure 5.12: Screenshots of (A) IEEE-IEC Gateway and (B) Receiver in Routed-SV traffic generation test

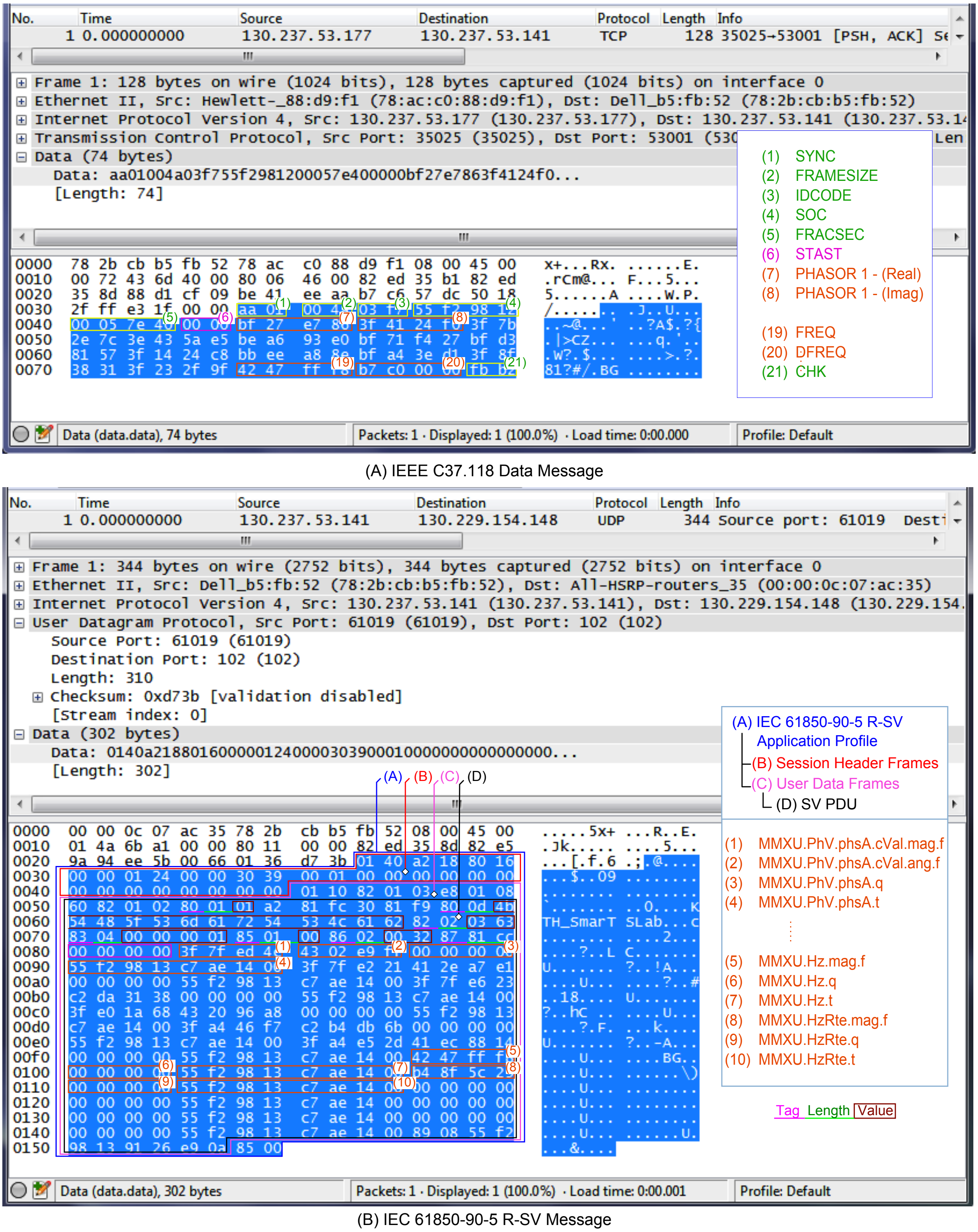


Figure 5.13: Wireshark capture analysis of (A) IEEE C37.118.2 Data message and (B) IEC 61850-90-5 R-SV message transferring the same PMU data

(B) show the session header frames and the bytes assigned as (C) are the user data of session layer. The frames marked as group (D) are the svPdu bytes.

The group of bytes marked as (B) in Figure. 5.13.B contains the session header frames of : (1) *LI* (0x01), (2) *TI* (0x40), (3) *Session Identifier* (0xA2), (4) *Length Identifier* (0x18), (5) *Common Header* (0x80), (6) *Length Identifier* (0x16), (7) *SPDU Length* (0x{0x00 00 01 24}), (8) *SPDU Number* (0x{00 00 30 39}), (9) *Version Number* (0x{0x00 01}), (10) *TimeOfCurrentKey* (0x{00 00 00 00}), (11) *TimeToNextKey* (0x{00 00}), (12) *Security Algorithms* (0x{00 00}), (13) *KeyID* (0x{00 00 00 00}).

The frames marked as group (C) in Figure. 5.13.B, starts with: (1) *User Data Length* (0x{00 00 01 10}), (2) *Payload Type* (0x{82}), (3) *Simulation* (0x{01}), (4) *APPID* (0x{03 E8}), (5) *APDU Length* (0x{01 08}) and continued with the svPdu bytes marked as group (D).

The svPdu in Figure. 5.13.B consistent with svPdu structure shown in Figure. 4.11, starts with svPdu Tag (0x60) and its Length field (0x{0x82 01 02}), then continued with (1) Number of ASDUs (noASDU) with T-L-V fields of (0x{80}-0x{01}-0x{01}) and (2) Sequence of ASDUs started with T-L fields of (0x{A2}-0x{81 FC}).

In this implementation, one ASDU is defined in each svPdu. ASDU 1 started with T-L fields of (0x{30}-0x{81 F9}), and followed by general svPdu information in following Tag-Length-Value format:

(1) *svID* (0x{80}-0x{0D}-0x{4B ...}),

(2) *smpCnt* (0x{82}-0x{02}-0x{03 63}),

(3) *ConfRev* (0x{83}-0x{04}-0x{00 00 00 01}),

(4) *smpSynch* (0x{85}-0x{01}-0x{00}),

(5) *smpRate* (0x{86}-0x{02}-0x{00 32}),

(6) *Sample* (0x{87}-0x{81 CC}-0x{00 ...}),

The *Sample* field contains the svPdu dataset. Unlike the goosePdu, the svPdu dataset elements are not encoded and transferred in their basic format. In this test, the 6 PHASORS (Va, Vb, Vc, Ia, Ib, Ic), FREQ and DFREQ are transferred within the dataset described in Figure. 4.12. For instance, the *MMXU.PhV.PhsA* and *MMXU.Hz* data objects in Figure. 5.13.B are described below:

(1) *MMXU.PhV.PhsA.cVal.mag.f* (0x{3F 7F ED 4C}),

(2) *MMXU.PhV.PhsA.cVal.ang.f* (0x{43 02 E9 F1}),

(3) *MMXU.PhV.PhsA.q* (0x{00 00 00 00}),

(4) *MMXU.PhV.PhsA.t* (0x{55 F2 98 13 C7 AE 14 00}),

(5) *MMXU.Hz.mag.f* (0x{42 47 FF F8}),

(6) *MMXU.Hz.q* (0x{00 00 00 00}),

(7) *MMXU.Hz.t* (0x{55 F2 98 13 C7 AE 14 00}),

6 Conclusion and Future works

6.1 Conclusions

Some major events such as blackout in the Northeastern part of the United States in 2003 triggered the widespread application of synchrophasor measurements in Wide-Area Measurement Systems (WAMS) and Wide-Area Control Systems (WACS) in power system networks.

The 2005 version of IEEE Std C37.118, which has been the dominant standard for synchrophasor measurement systems, included both measurement requirements and real-time data transfer requirements.

To simplify widespread adoption and facilitate the use of other communication protocols for phasor data transfer, in recent version of IEEE C37.118 standard released in 2011, the measurement and data transfer parts are segregated and published in two separate standards. This split facilitated the harmonization of IEEE Std C37.118-2005 with IEC 61850.

In 2012, the IEC TR 61850-90-5 published as a standard providing a way of exchanging synchrophasor data between PMUs, PDCs, WAMPAC (Wide Area Monitoring, Protection, and Control), and between control center applications. The data, to the extent covered in IEEE C37.118, are transported in a way that is compliant to the concepts of IEC 61850.

Some gaps not addressed in IEEE C37.118 i.e. cyber security requirements of power system network, are fulfilled in IEC TR 61850-90-5.

In this work, the main goal was to develop an IEEE-IEC gateway capable of running in an embedded system with minimum hardware requirements (no operating system, minimum memory,...), enabling fast cyclic transfer of synchrophasor streams over wide-area networks, thereby minimizing latencies in real-time applications.

In order to reach this objective, IEEE C37.118.2 and IEC TR 61850-90-5 standards studied in detail. The understandings for implementation of these two standards were presented in Chapters. 3 and 4.

The "Khorjin" library is developed to act as an IEEE-IEC Gateway. Khorjin is designed in such a way to receive synchrophasor data from PMU/PDC using the IEEE C37.118.2 protocol, map PMU data to IEC 61850 data model, and publish IEC 61850-90-5 messages in either Routed-Sampled Value or Routed-GOOSE format.

Another part in the Khorjin library is developed to receive and parse the Routed-SV and Routed-GOOSE messages and provide the extracted raw synchrophasor data to the subscribed applications.

The functionality of Khorjin is tested and validated in a Real-Time Hardware-in-the-Loop (HIL) simulation environment to assess its conformance to the functional requirements in the IEEE C37.118.2 and IEC 61850-90-5 standards.

6.2 Future works

IEC TR 61850-90-5 is a comprehensive standard, and its complete implementation is far beyond the scope of a master thesis.

As described in Section. 4.4, in this work, the two communication mechanisms of Routed-Sampled Value and Routed-GOOSE using UDP/IP are implemented for transfer of IEEE C37.118.2 compliant PMU/PDC data. Implementation of control and configuration services illustrated in Figure. 4.3, are the main tasks to be followed as the next steps of this project.

In addition, implementation of the security algorithm introduced in session protocol of IEC TR 61850-90-5, is another important task to be developed in future works.

Annex A Cyclic Redundancy Code (CRC)

At the Gateway side, using following code introduced in Annex B of [18], the CRC is calculated and transmitted in the CHK frame of each of four types of messages. At the receiver side, using the same algorithm, the CRC is calculated for the received message and compared with CHK word inside the exchanged message. In case of mismatch of the two CHK words the received message is discarded.

```
uint16_t
ComputeCRC(const uint8_t *Message, uint32_t MessLen)
{
        uint16_t crc = 0xFFFF;
        uint16_t temp;
        uint16_t quick;
        int i;
        for (i = 0; i < MessLen; i++)
        {
                temp = (crc >> 8) ^ Message[i];
                crc <<= 8;
                quick = temp ^ (temp >> 4);
                crc ^= quick;
                quick <<= 5;
                crc ^= quick;
                quick <<= 7;
                crc ^= quick;
        }
        return crc;
}
```

In this code, `*Message` is the pointer to the first byte in the message and `MessLen` is the number of characters in the message, not including the CHK on the end.

Annex B Khorjin library API functions - Gateway

B.1 IEEE C37.118.2 traffic generation and parsing

Following functions in Khorjin library are used to communicate with PMU/PDC using the IEEE C37.118.2 protocol.

```
C37_118client
c37_118client_connect(PMUConnectionData* parameters, uint32_t timeoutInMs);

void c37_118client_stopTransmission(C37_118client self);

void c37_118client_startReceiveDataStram(C37_118client self);

void c37_118client_startTransmission(C37_118client self);

int c37_118client_getConfiguration(C37_118client self);

int c37_118client_getDataStream(C37_118client self);

void c37_118client_printDataStream(C37_118client self);

void c37_118client_printConfiguration(C37_118client self);

void c37_118client_destroy(C37_118client self);
```

B.2 Routed-SV traffic generation

Following functions are developed in Gateway part of Khorjin for the purpose of setting the svPdu parameters.

```
void sampledValuePDU_enableDataSet(Iec61850_90_5Message self, bool status);

void sampledValuePDU_setDataSetRefernece(Iec61850_90_5Message self,
const char* dataSetReference);

void sampledValuePDU_setConfRev(Iec61850_90_5Message self, int confRev);

void sampledValuePDU_setSampleCount(Iec61850_90_5Message self,
uint16_t smpCount);

void sampledValuePDU_setRefreshTime(Iec61850_90_5Message self,
uint64_t timeValueinMs);

void sampledValuePDU_enableRefreshTime(Iec61850_90_5Message self, bool status);

void sampledValuePDU_setSmpSynch(Iec61850_90_5Message self, uint8_t smpSynch);

void sampledValuePDU_setSmpRate(Iec61850_90_5Message self, uint16_t smpRate);

void sampledValuePDU_enableSmpRate(Iec61850_90_5Message self, bool status);

void sampledValuePDU_setSmpMod(Iec61850_90_5Message self, uint16_t smpMode);

void sampledValuePDU_enableSmpMod(Iec61850_90_5Message self, bool status);

void sampledValuePDU_setAbsoluteTime(Iec61850_90_5Message self,
uint64_t timeValueinMs);

void sampledValuePDU_enableAbsoluteTime(Iec61850_90_5Message self,
bool status);
```

B.3 Routed-GOOSE traffic generation

Using the below listed functions the goosePdu parameters will be set.

```
void goosePDU_setGoCbRef(Iec61850_90_5Message self, const char* goCbRef);

void goosePDU_setTimeAllowedToLive(Iec61850_90_5Message self,
uint32_t timeAllowedToLive);

void goosePDU_setDatSetReference(Iec61850_90_5Message self,
const char* dataSetRef);

void goosePDU_setTimeStamp(Iec61850_90_5Message self, uint64_t timeValueinMs);

void goosePDU_setStNum(Iec61850_90_5Message self, uint32_t stNum);

void goosePDU_setSqNum(Iec61850_90_5Message self, uint32_t sqNum);

void goosePDU_setTest(Iec61850_90_5Message self, bool status);

void goosePDU_setConfRev(Iec61850_90_5Message self, uint32_t confRev);

void goosePDU_setNdsCom(Iec61850_90_5Message self, bool status);
```

Bibliography

[1] K. Martin, "Synchrophasor standards development - IEEE c37.118 & IEC 61850," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, 2011, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5718521

[2] H. Falk, M. Adamiak, D. Baigent, and V. Madani, "An overview of the new IEC 61850 synchrophasor publish-subscribe profile," in *Protective Relay Engineers, 2013 66th Annual Conference for*, 2013, pp. 309–321. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6822046

[3] *Communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit Synchrophasor information according to IEEE C37.118*, IEC Std. 61 850-90-5 (Edition 1.0), 2012-05.

[4] M. Seewald, "Building an architecture based on IP-multicast for large phasor measurement unit (pmu) networks," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*, 2013, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6497794

[5] Cisco, "Cisco and sisco collaborate on open source synchrophasor framework," Cisco Systems, Inc., Tech. Rep., 2011. [Online]. Available: http://www.cisco.com/web/strategy/docs/energy/cisco-SISCO_factsheet.pdf

[6] "Unified field area network architecture for distribution automation," Tech. Rep., January 16, 2014. [Online]. Available: http://www.cisco.com/c/dam/en/us/products/collateral/routers/500-series-wpan-industrial-routers/ida_wp.pdf

[7] J.-D. Lee, S.-J. Lee, J.-H. Bae, and D.-Y. Kwon, "The pmu interface using IEC 61850," in *ICT Convergence (ICTC), 2013 International Conference on*, 2013, pp. 1125–1128. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6675573

[8] X. Yong, Z. Dao-Nong, and Y. Yuehai, "The research of synchronized phasor measurement units in smart substations," in *Advanced Power System Automation and Protection (APAP), 2011 International Conference on*, vol. 3, 2011, pp. 2262–2267. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6180804

[9] [Online]. Available: http://www.salzburgresearch.at/en/projekt/otitos-open-test-implementation-iec-61850-90-5-based-transmission-synchrophasor-information/

[10] J. Ekanayake, K. Liyanage, J. Wu, A. Yokoyama, and N. Jenkins, *Smart Grid: Technology and Applications*. John Wiley & Sons, 2012.

[11] "Report to nist on the smart grid interoperability standards roadmap," Electric Power Research Institute (EPRI), Tech. Rep., August 2009.

[12] D. Mah, P. Hills, V. O. Li, and R. Balme, *Smart Grid Applications and Developments*. Springer, 2014.

[13] S. F. Bush, *Smart Grid: Communication-enabled Intelligence for the Electric Power Grid*. John wiley & sons, 2014.

[14] L. Vanfretti, M. Chenine, M. Almas, R. Leelaruji, L. Angquist, and L. Nordstrom, "Smarts lab — a laboratory for developing applications for wampac systems," in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6344839

[15] A. G. Phadke and J. S. Thorp, *Synchronized phasor measurements and their applications*. Springer Science & Business Media, 2008.

[16] E. Schweitzer, D. Whitehead, G. Zweigle, and K. Ravikumar, "Synchrophasor-based power system protection and control applications," in *Protective Relay Engineers, 2010 63rd Annual Conference for*, 2010, pp. 1–10. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5469481

[17] *IEEE Standard for Synchrophasor Measurements for Power Systems*, IEEE Std. C37.118.1-2011 (Revision of C37.118-2005), 2011. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6111219

[18] *IEEE Standard for Synchrophasor Data Transfer for Power Systems*, IEEE Std. C37.118.2-2011 (Revision of C37.118-2005), 2011. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6111222

[19] R. Mackiewicz, "Overview of IEC 61850 and benefits," in *Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES*, 2006, pp. 376–383. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1668522

[20] *Communication networks and systems in substations - Part 1: Introduction and overview*, IEC Std. 61 850-1 (First edition), 2003-04.

[21] "Abb review special report iec 61850," ABB Asea Brown Boveri Ltd, Switzerland, Tech. Rep., August 2010. [Online]. Available: http://www.abb.com/abbreview

[22] International electrotechnical commission webstore. (Accessed 19/06/2015). [Online]. Available: https://webstore.iec.ch/publication/20083

[23] A. Elgargouri, M. Elfituri, and M. Elmusrati, "IEC 61850 and smart grids," in *Electric Power and Energy Conversion Systems (EPECS), 2013 3rd International Conference on*, 2013, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6713080

[24] *Communication networks and systems in substations - Part 7-1: Basic communication structure for substation and feeder equipment - Principles and models*, IEC Std. 61 850-7-1 (First edition), 2003-07.

[25] *Communication networks and systems in substations - Part 7-2: Basic communication structure for substation and feeder equipment - Abstract communication service interface (ACSI)*, IEC Std. 61 850-7-2 (First edition), 2003-05.

[26] *Communication networks and systems in substations - Part 7-3: Basic communication structure for substation and feeder equipment - Common data classes*, IEC Std. 61 850-7-3 (First edition), 2003-05.

[27] *Communication networks and systems in substations - Part 7-4: Basic communication structure for substation and feeder equipment - Compatible logical node classes and data classes*, IEC Std. 61 850-7-4 (First edition), 2003-05.

[28] C. Kriger, S. Behardien, and J.-C. Retonda-Modiya, "A detailed analysis of the goose message structure in an iec 61850 standard-based substation automation system," 2013.

[29] J. Konka, C. Arthur, F. Garcia, and R. Atkinson, "Traffic generation of IEC 61850 sampled values," in *Smart Grid Modeling and Simulation (SGMS), 2011 IEEE First International Workshop on*, 2011, pp. 43–48. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6089025

[30] K. Martin, G. Brunello, M. Adamiak, G. Antonova, M. Begovic, G. Benmouyal, P. Bui, H. Falk, V. Gharpure, A. Goldstein, Y. Hu, C. Huntley, T. Kase, M. Kezunovic, A. Kulshrestha, Y. Lu, R. Midence, J. Murphy, M. Patel, F. Rahmatian, V. Skendzic, B. Vandiver, and A. Zahid, "An overview of the IEEE standard c37.118.2—synchrophasor data transfer for power systems," pp. 1980–1984, 2014, smart Grid, IEEE Transactions on. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6839218

[31] H. Falk and M. Burns, "Mms and asn.1 encodings, simple samples and explanations," Systems Integration Specialists Company (SISCO), Inc, August 2001.

[32] "Implementation guideline for digital interface to instrument transformers using iec 61850-9-2," UCA International Users Group, Tech. Rep., 2004. [Online]. Available: http://iec61850.ucaiug.org/Implementation%20Guidelines/DigIF_spec_9-2LE_R2-1_040707-CB.pdf

[33] [Online]. Available: http://www.libiec61850.com/libiec61850/

[34] P. Romano and M. Paolone, "Enhanced interpolated-DFT for synchrophasor estimation in FPGAs: Theory, implementation, and validation of a pmu prototype," vol. 63, no. 12, pp. 2824–2836, 2014. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6818415

[35] [Online]. Available: https://www.wireshark.org/

[36] [Online]. Available: http://www.h-schmidt.net/FloatConverter/IEEE754.html

List of publications

[1] Firouzi S.R., Vanfretti L., Ruiz-Alvarez A., Mahmood F., Hooshyar H. and Cairo I., "An IEC 61850-90-5 gateway for IEEE C37.118.2 synchrophasor data transfer", Submitted to IEEE Power Engineering Society General Meeting, Boston, USA, 2016.

An IEC 61850-90-5 Gateway for IEEE C37.118.2 Synchrophasor Data Transfer

Seyed Reza Firouzi*, Luigi Vanfretti*†, Albert Ruiz-Alvarez‡, Farhan Mahmood*,
Hossein Hooshyar* and Ignasi Cairo‡
*KTH Royal Institute of Technology, Stockholm, Sweden
†Statnett SF, Oslo, Norway
‡IREC Catalonia Institute for Energy Research, Barcelona, Spain
Email: srfi@kth.se

***Abstract*—This work describes the implementation and validation of a library named Khorjin to receive and parse synchrophasor data from a PMU/PDC based on IEEE C37.118 protocol, map it to the IEC 61850 data model and transmit the synchrophasor data through either Routed-Sampled Value or Routed-GOOSE services defined in the IEC 61850-90-5 protocol. In addition, the library is capable of acting as the receiver and parser of IEC 61850-90-5 messages, extracting raw synchrophasor data and feeding to subscriber applications. The functionality of the Khorjin library is validated in a Real-Time Hardware-in-the-Loop (HIL) simulation environment to asses its conformance to the functional requirements of the IEC 61850-90-5 standard.**

***Index Terms*—IEC 61850-90-5, IEEE C37.118.2, PDC, PMU, Routed-GOOSE, Routed-Sampled Value, Synchrophasor measurment**

I. Nomenclature

IP	Internet Protocols
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
PMU	Phasor Measurment Unit
PDC	Phasor Data Concentrator
ROCOF	Rate Of Change Of Frequency
(R-)SV	(Routed-) Sampled Value
(R-)GOOSE	(Routed-) Generic Object-Oriented Substation Event
WAMPAC	Wide-Area Monitoring, Protection And Control

II. Introduction

The concept of synchronized phasors was first introduced in the 1980s[1] and standardized for the first time in 1995 with the IEEE 1344 standard. During the post mortem analysis of the Northeast blackout in 2003, it was understood that widespread synchrophasor measurements could help in preventing such great disturbances in the grid [2]. Then, in 2005, the C37.118 standard with the title of "IEEE Standard for Synchrophasors for Power Systems" was approved and published.

In 2009, IEEE proposed dual logo standard and requested IEC to accept IEEE C37.118 as a part of IEC. However, IEC rejected the request because the IEC 61850-9-2 could provide similar streaming functionalities. In consequence, a joint task force was formed between IEEE and IEC. The objective of this task force was to determine how C37.118 could be harmonized with IEC 61850.

In 2011, as the result of the joint work, IEEE C37.118-2005 split into two parts. C37.118.1: that standardized how to measure synchrophasors and C37.118.2: that specified the data transfer requirements. The formation of this task force was the formal start of the development of IEC TR 61850-90-5 that was finally published in 2012.

IEC TR 61850-90-5, a technical report prepared by IEC technical committee 57-Power systems management and associated information exchange-, introduces a protocol for exchange of synchrophasor data between PMUs, PDCs, WAMPAC systems and control center applications in the context of the IEC 61850 standard [3].

While concept of synchrophasor data transfer using IEC 61850, has been addressed in [1], [2] and [4], a few publications are available describing the implementation works. In [5], Lee *et al* developed a IEC 61850-based PMU interface using MMS and GOOSE services. In [6], Yong *et al* implemented synchrophasor data mapping to IEC 61850-9-2 Sampled Value service. In [7], a joint work introduced for an open source implementation framework of synchronized phasor measurement communications based on the IEC TR 61805-90-5, but no release is available yet. However, none of these works addressed the concept of an IEEE-IEC Gateway implementing the Routed-SV and Routed-GOOSE services introduced in [3].

The main goal of this work is to develop IEEE-IEC gateway capable of running in an embedded system with minimum hardware requirements, enabling fast cyclic transfer of synchrophasor streams over wide-area networks, thereby minimizing latencies in real-time applications. As depicted in Fig.1, this work required to design and develop a library to (1) Communicate with a PMU/PDC using the C37.118 protocol, (2) map synchrophasor data to the IEC 61850 data model and (3) publish the IEC 61850-90-5 messages transmitting synchrophasor using either R-SV or R-GOOSE services defined in [3].

In addition, another part in the library fulfills the requirements of receiving and parsing the R-SV and R-GOOSE frames and provide the raw synchrophasor data to subscribed applications.

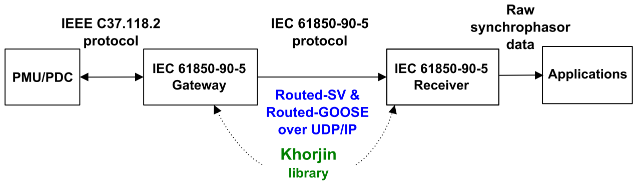


Fig. 1. Khorjin library functional diagram

For the sake of simplicity, the library is named "Khorjin[1]".

III. Understanding IEEE C37.118.2 for Implementation

In the context of C37.118.2, synchrophasor data transmission is handled by exchanging four message types:

1) Data Message
2) Configuration Message
3) Header Message
4) Command Message

The first three message types are transmitted from the data source (PMUs/PDCs), and the last one (Command) is received by the PMU/PDC.

As shown in Fig.2, a C37.118 message contains (1) Common and (2) Message-specific frames.

In the common part: (1) SYNC provides synchronization and frame identification, (2) FRAMSIZE indicates the total number of bytes in the frame (including CHK), (3) IDCODE identifies the source of a data, header, or configuration message, or the destination of a command message, (4) SOC (Second-Of-Century) is a count of seconds from UTC midnight (00:00:00) of January 1, 1970, to the current second, (5) FRACSEC (FRACtion-of-SECond) contains a time quality byte and the number of fraction of seconds, while (6) CHK is used to verify (or at least indicate) that a set of data has not been corrupted.

Considering that in C37.118 the measurement timestamp is defined as $SOC + FRACSEC/TIME\_BASE$, each second is divided into an integer number of subdivisions by the TIME_BASE parameter that is defined in the configuration frame. The FRACSEC count is an integer representing the numerator of FRACSEC with TIME_BASE as the denominator.

A. Data Message

Data message contains the synchrophasor measurements made by the PMU. In normal operation, the PMU continuously streams the data frames.

Following the transmission order shown in Fig.3, the Data-frame specific part of a C37.118 message contains: (1) STAT flag that gives the status of the data contained in that block, (2) PHASORS that holds the phasor data in either *16-bit Integer* or *32-bit IEEE floating-point* format, and in each case, the complex value of phasors represented in *Rectangular* or *Polar* format; (3) FREQ, (4) DFREQ, (5) ANALOG give the value of system frequency, ROCOF and user defined analog values respectively in either *16-bit Integer* or *32-bit IEEE floating-point* format; and (6) DIGITAL transmits user-defined binary data such as bit mapped status or flag.

[1]In the Persian language, KHORJIN, is a special bag placed on the two sides of a horse, which was used for transfering of parcels.

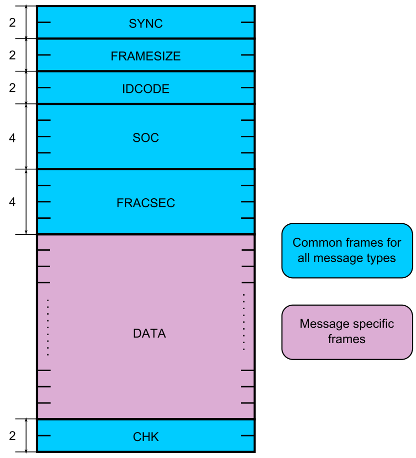


Fig. 2. Common and Message-specific frame structure

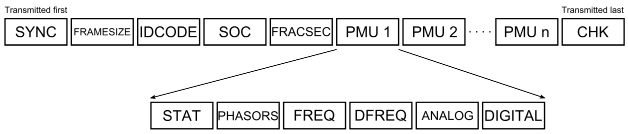


Fig. 3. Data frame transmission order

B. Configuration Message

In order to enable the receiver to interpret the Data messages and extract the measurement values, the *Configuration message* defines three types of frames:

- Configuration Frame 1 (CFG-1)
- Configuration Frame 2 (CFG-2)
- Configuration Frame 3 (CFG-3)

CFG-1 provides information about the PMU/PDC capability. It defines the set of data that the PMU/PDC is capable of reporting [8]. CFG-2 gives information about the synchrophasor data being transmitted in the data frame. The CFG-3 frame is similar to other configuration frames, but with some additional and modified data words in comparison to CFG-1 and CFG-2. In [9], CFG-3 is introduced as optional.

For each PMU, the following words are transmitted in the CFG-1 and CFG-2 messages: TIME_BASE, NUM_PMU, STN, IDCODE, FORMAT, PHNMR, ANNMR, DGNMR, CHNAM, PHUNIT, DIGUNIT, FNOM and CFGCNT. The last word in configuration frames is DATA_RATE. A detailed description of these words is available in [9].

C. Header & Command Messages

The *header message* is human readable, it provides user defined descriptive information sent from the PMU/PDC.

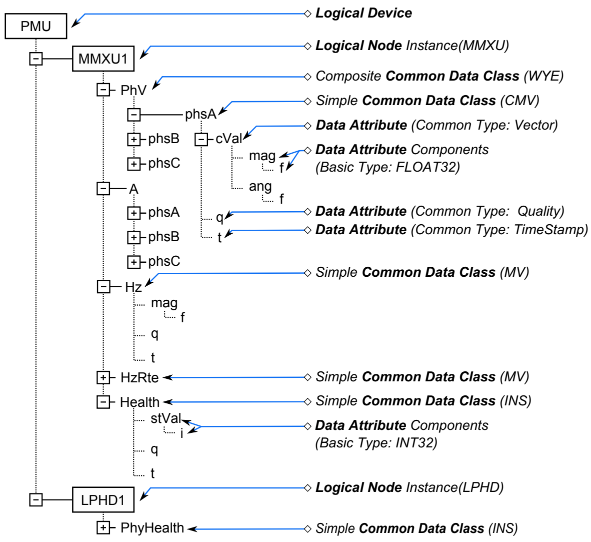


Fig. 4. IEC 61850 PMU Data Model

Command messages are machine-readable codes sent to the PMU/PDC for control or configuration. The following functions are defined to be sent as commands to PMU/PDC.

- Turn off transmission of data frames
- Turn on transmission of data frames
- Send Header frame
- Send CFG-1 frame
- Send CFG-2 frame
- Send CFG-3 frame

D. Underlying Communication Mechanism

In IEEE Std C37.118.2, no specific underlying communication protocol is introduced and just four frame types and their specifications are defined. However, in Annex E and F of the standard, two protocols (*Serial* and *Internet protocol*) are suggested. The Serial communication protocol is out-dated, and in practice TCP/IP or UDP/IP protocols are used for synchrophasor data transmission.

IV. Understanding IEC TR 61850-90-5 for Implementation

IEC 61850, was introduced as an international standard for substation automation, supporting interoperability of IEDs from different manufacturers. However, it has been extended to other domains, such as DR, Hydroelectric Power Plants and recently in part 90-5 to wide-area transmission of synchrophasor information according to IEEE C37.118. Therefore, the title in the second edition of this standard has been renamed to Communication Networks and Systems for Power Utility Automation. This comprehensive standard covers three aspects of: *Data Modeling*, *Communication* and *Engineering*.

A. Data Modeling

The IEC 61850 data model starts with a *Physical Device*, that is an Intelligent Electronic Device (IED) and in this case could be a protection relay or a standalone PMU. Each *Physical Device* may contain one or more *Logical Device*, for instance the PMU function in a protection relay is a *Logical Device*. In IEC 61850-7-2, *Logical Device* is defined as an ”entity that represents a set of typical substation functions”. Every *Logical Device* consists of a minimum of three *Logical Nodes*. In IEC 61850-7-2, *Logical Node* is defined as an ”entity that represents one typical substation function”. Each *Logical Node* includes one or more data elements. Each data element is of one of the *Common Data Class* (CDC) standardized in IEC 61850-7-3. Each CDC has a standard name describing the type and structure of the data within Logical node. Each CDC contains several individual *Data Attributes* which are categorized based on their *Functional Constraints* (FC).

In IEC 61850-7-4, the measurement Logical Node (MMXU) is defined for calculation of currents, voltages, powers and impedances in a three-phase system. In section 13.3 of [3], it is mentioned that the phasor data and the frequency contained in the C37.118 telegram, can be mapped directly to MMXU data objects of IEC 61850.

The unspecified analog and digital data in a C37.118 telegram should be mapped to any IEC 61850 data object fitting to the appropriate data type and carrying the needed semantic.

By introduction of IEC 61850-90-5, the new data object of HzRte is added to the MMXU logical node to accommodate the ROCOF data. Then the Phasors, Frequency and ROCOF data in the context of IEEE C37.118 are modeled as the data objects of the MMXU logical node, as depicted in Fig.4.

In addition to phasor data, the information about the status of the PMU is transmitted using the common data class named ”PhyHealth” in an instance of the LPHD Logical Node.

B. Communication Mechanism

In IEC 61850, the SV service is introduced for cyclic transfer of data inside substations and the GOOSE service is defined for event-based transfer of data. However, these communication protocols are over Ethernet inside a substation. For the purpose of data transfer outside of the substation, there are two options introduced in [3]:

- *Tunneling;* SV and GOOSE services across some high speed communication networks like SDH or SONET.
- Communicating SV and GOOSE services via *Internet Protocol (IP) networks*.

In the second option, IEC 61850 has been enhanced by mapping of SV and GOOSE messages onto an IP based protocol. Based on the cyclic nature of these services, UDP with multicast addressing is the transport protocol chosen in the standard. In this regard, the new mapping of the SV and GOOSE services uses routable UDP, and are called Routed-Sample Value and Routed-GOOSE.

While both tunneling and IP protocol mechanisms are addressed in [3], the scope of this work is limited to implementation of R-SV and R-GOOSE services.

In the routed versions of SV and GOOSE introduced in [3], the application layer specifications of ”native” SV and

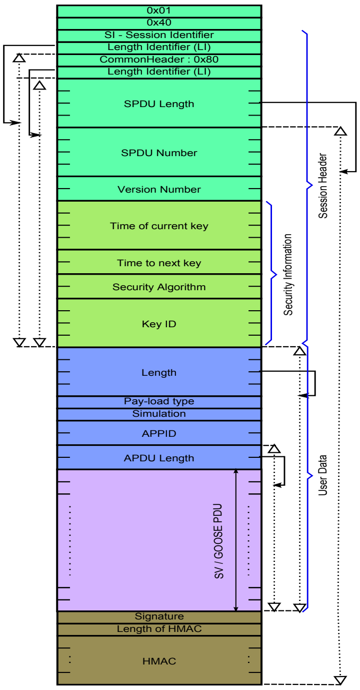


Fig. 5. IEC 61850-90-5 application profile specification

GOOSE services defined respectively in part 9-2 and part 8-1 of IEC 61850, have remained unchanged and a new protocol is introduced in the session layer for sending the SV and GOOSE messages over UDP/IP.

The complete Application Profile specification (including session layer) of IEC 61850-90-5 R-SV and R-GOOSE messages is illustrated in Figure 5.

The Session header starts with a single-byte `Session Identifier (SI)`, followed by the single-byte`Length Identifier (LI)`. This length covers the length of all of the parameter fields for the session header, but not the user data of the session protocol.

There are four `Session Identifier` defined in [3] among which 0xA1 is for R-GOOSE and 0xA2 is for R-SV. The `SI` and associated length fields are followed by `CommonHeader` with the hexadecimal value of 0x80. The Common session header includes following data: `SPDU Length`, `SPDU Number`, `Version Number` and `Security Information`. The User data part contains: `Length` of user data, `PayLoad Type`, `Simulation`, `APPID`, `APDU Length` and `SV / GOOSE PDU`. Finally, the frames are ended with `Signature` frames.

V. Implementation and Results

The Khorjin library was developed in C language. It is utilized on two separate Gateway and Receiver platforms. The Receiver output, i.e. raw synchrophasor data, is fed to the subscribed applications.

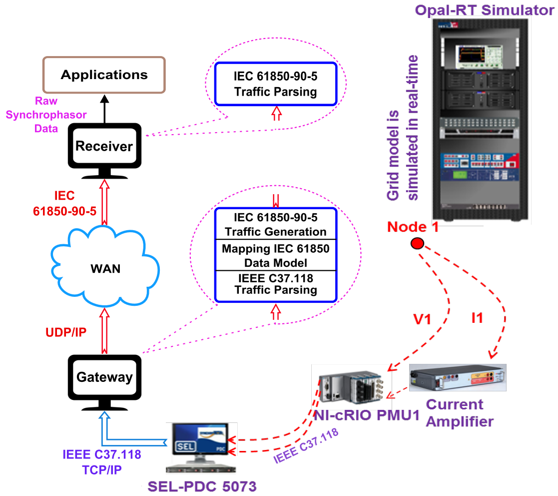


Fig. 6. Hardware-in-the-loop setup

A. Design of Khorjin Library

In the Gateway of the Khorjin, two sets of connection data are required as input: (1) IP address, Port number and IDCODE of the PMU/PDC and (2) IP address (for Unicast UDP/IP), Port number (in [3] it is set to 102) and APPID for IEC61850-90-5 traffic generation part of the gateway.

Having the connection data group (1), upon a successful TCP/IP connection between Gateway and PMU/PDC, Configuration message is received. Having the configuration of the PMU/PDC, Data streams are started to be received and parsed. Using connection data group (2) the received synchrophasor data are mapped to either R-SV or R-GOOSE protocol and transmitted over UDP/IP.

In the Receiver of Khorjin, the Port number and the APPID are required as input. Upon receiving a UDP frame, the `APPID` in the received message will be checked to be consistent with APPID of the Receiver, otherwise the received message will be discarded.

After parsing the Session Header and User Data frames, dataset of SV/GOOSE PDU is decoded and synchrophasor data is extracted.

B. RT-HIL Validation of the Functionality of Khorjin

In order to validate the functionality of Khorjin, testing was performed using the HIL setup in Fig6, and two separate computers where Khorjin is used either as Gateway or Receiver.

As shown in Fig. 6, a measurement location has been specified on a grid model that is simulated by the OPAL-RT real-time simulator. The measured voltages and currents are fed to PMU through the analogue output ports of the OPAL-RT simulator. As indicated in the figure, the PMU used in this setup is Compact Reconfigurable IO systems (CRIO) from National Instruments Corporation [10]. As the figure shows, the signals from RT simulator are passed through current amplifiers before being fed to the PMUs. Syncrophasors are

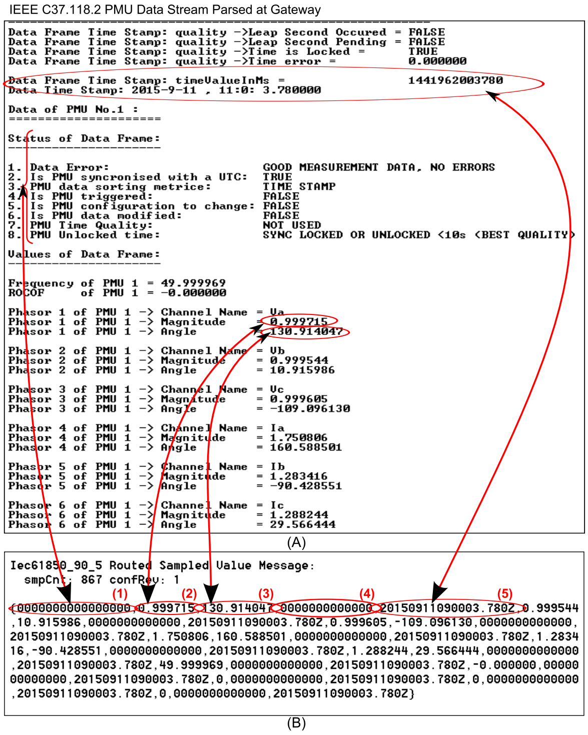


Fig. 7. IEEE-IEC Gateway and Receiver screenshots in R-SV traffic generation test

then sent to a PDC which streams the data over TCP/IP to the workstation holding IEEE-IEC Gateway. On another workstation the Receiver part of library receives the real-time streams of data in IEC 61850-90-5 format and parses the R-SV or R-GOOSE messages.

Figures 7.A and 7.B, show screenshots of Gateway and Receiver, respectively. It can be seen that the dataset of R-SV message, transferring synchrophasor data, starts with: (1) 16-bit STAT word, as introduced in [11] (2) Magnitude, (3) Angle, (4) Quality and (5) Timestamp of phasor 1 and ends by data objects of Health and PhyHealth CDCs. In Fig.8, the wireshark capture of the UDP/IP message transfering R-SV APDU is shown. In this figure, hexadecimal values of the R-SV message containing the Session Header and User Data described in Fig.5 are shown. As part of the User Data of the Session header, the SV PDU is pointed in the figure. In addition, the hexadecimal values of data (1) to (5) described in Fig.7 can be seen as the dataset of the SV PDU.

VI. Conclusion and Future Works

In this paper, IEEE Std C37.118.2, as the dominant protocol for synchrophasor data transfer, and IEC TR 61850-90-5, as a new standard for transfer of digital states and synchronized phasor measurements over wide-area networks, are described. The implementation of the Khorjin library was described. Khorjin acts (1) as a Gateway, transmitting IEEE C37.118.2

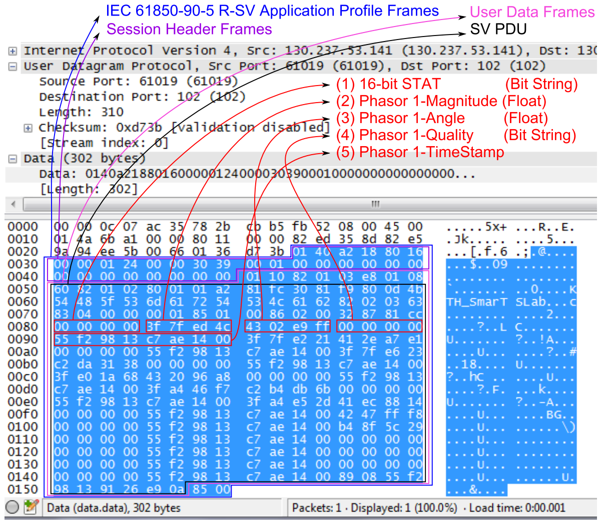


Fig. 8. IEC 61850-90-5 wireshark capture analysis in R-SV traffic generation test

compliant synchrophasor data in IEC 61850 using R-SV or R-GOOSE services defined in IEC 61850-90-5; and (2) as a Reciever, receiving and parsing IEC 61850-90-5 messages in either formats of R-SV or R-GOOSE. Implementation of the security algorithms in the session layer is in process as one of the most relevant future works of the project.

References

[1] K. Martin, "Synchrophasor standards development - IEEE c37.118 & IEC 61850," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, 2011, pp. 1–8.
[2] H. Falk, M. Adamiak, D. Baigent, and V. Madani, "An overview of the new IEC 61850 synchrophasor publish-subscribe profile," in *Protective Relay Engineers, 2013 66th Annual Conference for*, 2013, pp. 309–321.
[3] *Communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit Synchrophasor information according to IEEE C37.118*, IEC Std. 61 850-90-5, 2012-05.
[4] M. Seewald, "Building an architecture based on IP-multicast for large phasor measurement unit (pmu) networks," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*, 2013, pp. 1–5.
[5] J.-D. Lee, S.-J. Lee, J.-H. Bae, and D.-Y. Kwon, "The pmu interface using IEC 61850," in *ICT Convergence (ICTC), 2013 International Conference on*, 2013, pp. 1125–1128.
[6] X. Yong, Z. Dao-Nong, and Y. Yuehai, "The research of synchronized phasor measurement units in smart substations," in *Advanced Power System Automation and Protection (APAP), 2011 International Conference on*, vol. 3, 2011, pp. 2262–2267.
[7] "Cisco and sisco collaborate on open source synchrophasor framework," Cisco Systems, Inc., Tech. Rep., 2011. [Online]. Available: http://www.cisco.com/web/strategy/docs/energy/cisco-SISCO_factsheet.pdf
[8] K. Martin *et al*, "An overview of the IEEE standard c37.118.2—synchrophasor data transfer for power systems," pp. 1980–1984, 2014, smart Grid, IEEE Transactions on.
[9] *IEEE Standard for Synchrophasor Data Transfer for Power Systems*, IEEE Std. C37.118.2-2011 (Revision of C37.118-2005), 2011.
[10] P. Romano and M. Paolone, "Enhanced interpolated-DFT for synchrophasor estimation in FPGAs: Theory, implementation, and validation of a pmu prototype," vol. 63, no. 12, pp. 2824–2836, 2014.
[11] NASPI, "Naspi synchrophasor technical report - use of iec61850-90-5 to transmit synchrophasor information according to ieee c37.118," NASPI, Tech. Rep., 2014.