Electric Power Systems Research 144 (2017) 255–267

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Interpreting and implementing IEC 61850-90-5 Routed-Sampled Value and Routed-GOOSE protocols for IEEE C37.118.2 compliant wide-area synchrophasor data transfer

Seyed Reza Firouzi[a,*], Luigi Vanfretti[a], Albert Ruiz-Alvarez[b], Hossein Hooshyar[a], Farhan Mahmood[a]

[a] SmarTS Lab, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
[b] Catalonia Institute for Energy Research (IREC), Barcelona, Spain

ARTICLE INFO

Article history:
Received 9 April 2016
Received in revised form
25 November 2016
Accepted 6 December 2016
Available online 27 December 2016

Keywords:
IEC 61850-90-5
IEEE C37.118.2
PDC
PMU
Routed-GOOSE
Routed-Sampled Value
WAMPAC

ABSTRACT

Flexibility and adaptability requirements of future electric power grids for integrating distributed energy resources (DERs) call for the development of wide-area monitoring, protection and control (WAMPAC) applications, utilizing synchrophasor measurements provided by the phasor measurement units (PMUs).

IEEE C37.118 is the most utilized protocol for real-time exchange of synchronized phasor measurement data. In order to fulfill some gaps not addressed in IEEE C37.118, and also to harmonize with the IEC 61850 power utility automation standard, the IEC 61850-90-5 technical report has been developed. IEC TR 61850-90-5 introduces a mechanism for transfer of digital states and time synchronized phasor measurement data over wide-area networks between PMUs, phasor data concentrators (PDCs) and WAMPAC applications in the context of IEC 61850.

This work interprets the IEEE C37.118.2 and IEC 61850-90-5 Routed-Sampled Value and Routed-GOOSE protocols and describes the design and implementation of a library named Khorjin with the functionality of (1) an IEEE C37.118.2 to IEC 61850-90-5 gateway and protocol converter and (2) an IEC 61850-90-5 subscriber and traffic parser.

The main contribution of this work is the development of Khorjin library using only standard C libraries (i.e. independent from any operating system). This is allowing the use of the library in different platforms.

The design requirements and functionality of the Khorjin library has been tested in the KTH SmarTS Lab real-time hardware-in-the-loop (HIL) simulation environment to assess its conformance to the functional requirements of IEEE C37.118.2 and IEC 61850-90-5 standards.

1. Introduction

Future electric power systems shall be capable of integrating distributed energy resources (DERs). Renewable sources of photovoltaics, wind and fuel cells and other DERs such as electric vehicles and energy storage, and above all market-driven behavior of the demands will make the grid towards being more dynamic, increasing its operational complexity [1].

Smart grid, as a solution to handle this complexity, requires wide-area monitoring, protection and control (WAMPAC) applications, to anticipate and respond to system disturbances in a self-healing manner [2]. A typical WAMPAC architecture utilizes the synchronized phasor or so-called synchrophasor measurements provided by phasor measurement units (PMUs) throughout the network [3].

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, e.g. global positioning system (GPS) [4].

While measurements enable the 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. Hence, the synchrophasors are described as "the MRI[1] of the bulk power system" [5].

* Corresponding author.
E-mail address: srfi@kth.se (S.R. Firouzi).

[1] Magnetic resonance imaging (MRI) is a medical imaging technique used in radiology.

http://dx.doi.org/10.1016/j.epsr.2016.12.006

1.1. Background and literature review

The concept of a phasor synchronized with the power system was introduced in the 1980s [6], and standardized for the first time in 1995 within the IEEE 1344 standard. In 2005, this standard was replaced by IEEE C37.118 standard titled as "IEEE Standard for Synchrophasors for Power Systems".

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 request since the already available IEC 61850-9-2 standard could provide similar streaming functionalities. As a consequence, a joint task force was formed between IEEE and IEC to determine how to include IEEE C37.118 in the context of IEC 61850 substation automation standard.

As a result, in 2011, the IEEE standard C37.118-2005 was separated 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 61850-90-5 technical report (TR) that was finally published in 2012.

IEC TR 61850-90-5 prepared by IEC technical committee 57 (power systems management and associated information exchange), provides a way of exchanging synchrophasor data between PMUs, phasor data concentrators (PDCs), WAMPAC and control center applications [7] in the context of IEC 61850.

The concept of synchrophasor data transfer in the context of IEC 61850 has been addressed in [6,8,9], and a few number of publications are available describing the implementation work. In [10], Lee et al. developed an IEC 61850-based PMU interface (gateway) using manufacturing message specification (MMS) and generic object oriented substation event (GOOSE) services. In [11], Yong et al. implemented synchrophasor data mapping to IEC 61850-9-2 Sampled Value service. However these works addressed the native IEC 61850-8-1 GOOSE and IEC 61850-9-2 Sampled Value running over Ethernet inside a substation, without any transport and internet protocol as presented in IEC 61850-90-5 and implemented in this work.

In [12], a project is introduced to develop an open-source implementation framework for synchrophasor measurement communications based on the IEC TR 61850-90-5, but no release has been found. In [13], Madani et al. addressed a hybrid protocol synchrophasor system implementation using the IEC 61850-90-5 Routed-Sampled Value service for data transfer and IEEE C37.118.2 for acquiring the PMU configuration, however no detailed information was provided for the necessary IEEE C37.118 to IEC 61850 mapping mechanism.

1.2. Motivation

The current available synchrophasor measurement technology installed at electric power grids throughout the world is the result of more than two decades of deployment. These infrastructures have been implemented and developed using the IEEE C37.118 standard at different hierarchy levels from PMUs and PDCs, to super PDC (SPDC) and application systems in control centers. With the introduction of IEC 61850-90-5 in 2012, there has been efforts for migration from IEEE C37.118 to IEC 61850-90-5. While new systems could easily adopt the new standard, it is a challenge to upgrade the already installed system components to support the new IEC protocol. This issue is an obstacle for migration to the new standard, and more important, interoperability. As it is shown in Fig. 1, one likely future scenario is that WAMPAC systems will be comprised by two islanded and segregated systems, one that uses the old IEEE protocol and the other one that is adapted to the new IEC standard. In order to address this challenge, one solution is to develop an IEEE C37.118 to IEC 61850-90-5 gateway,

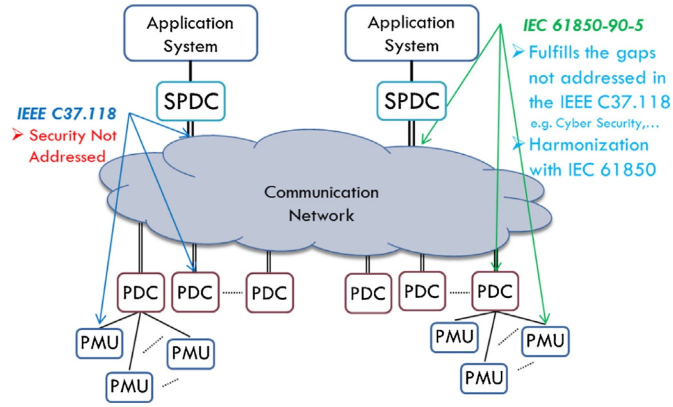

Fig. 1. Likely scenario of future WAMPAC systems' architecture.

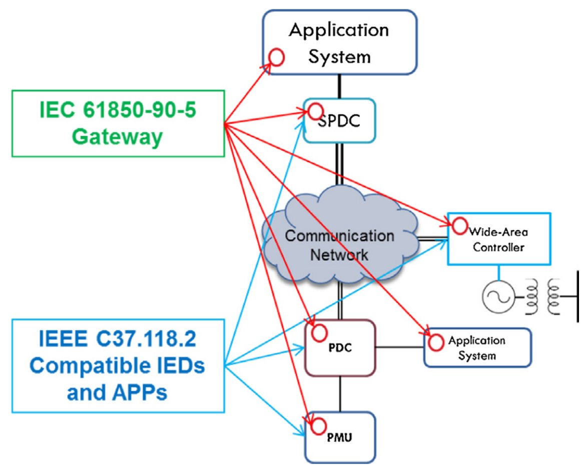

Fig. 2. IEC 61850-90-5 gateway application.

enabling the integration of C37.118 compliant intelligent electronic devices (IEDs) and application system in the context of the IEC 61850 standard.

1.3. Objective

As it is illustrated in Fig. 2, the objective of this work was to develop a software tool capable of acting as an IEC 61850-90-5 gateway for real-time integration of IEEE C37.118.2 compliant synchrophasor data, and able of running on different components and platforms within the hierarchy of the system.

1.4. Paper contributions

The main contribution of this work is the development of a software library capable of functioning as (1) an IEEE C37.118.2 to IEC 61850-90-5 gateway and (2) an IEC 61850-90-5 subscriber and traffic parser. The source code was implemented using standard C libraries (i.e. independent from any operating system). This allows the use of the library in different platforms, particularly on embedded systems with minimum hardware requirements, thereby minimizing latencies in real-time applications by enabling fast cyclic transfer of synchrophasor streams over wide-area networks. For the sake of easy referencing, this library is named as "Khorjin".[2]

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

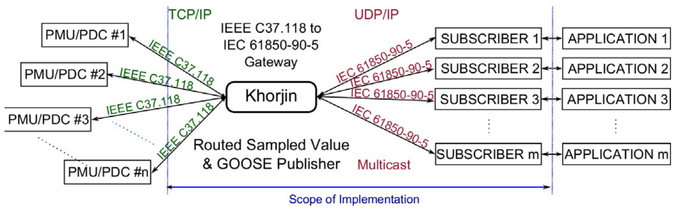

Fig. 3. Khorjin library interface overview.

As it is illustrated in Fig. 3, the Khorjin library is designed, implemented and validated to (1) receive and parse synchrophasor data originating from a PMU or PDC using the IEEE C37.118.2 protocol, (2) map it to the IEC 61850 data model, and (3) transmit the synchrophasor data through either Routed-Sampled Value or Routed-GOOSE services defined in the IEC 61850-90-5 report.

In addition to the IEEE C37.118.2 to IEC 61850-90-5 gateway functionality, the Khorjin library is capable of acting as both receiver and parser of IEC 61850-90-5 messages (i.e. Routed-Sampled Value and Routed-GOOSE), extracting raw synchrophasor data and feeding it to subscribed applications. The functionality of the Khorjin library has been validated in the KTH SmarTS Lab real-time hardware-in-the-loop (RT-HIL) simulation environment to assess its conformance to the functional requirements of the IEEE C37.118.2 and IEC 61850-90-5 standards.

This paper focuses on interpreting the synchrophasor data exchange mechanism in both IEEE C37.118.2 and IEC 61850-90-5 standards and its mapping from IEEE C37.118.2 to IEC 61850-90-5 as implemented in gateway functionality of the Khorjin library.

1.5. Paper organization

In Section 2, the mechanism for synchrophasor data exchange using the IEEE C37.118.2 protocol is explained. In Section 3, the PMU data modeling and communication based on IEC 61850-90-5 Routed-Sampled Value and Routed-GOOSE services are introduced in detail. The functional description of Khorjin library architecture is provided in Section 4. After describing the validation and performance test results in Section 5, this article is ended in Section 6 by presenting the conclusions and future works.

2. Understanding IEEE C37.118.2 synchrophasor data transfer mechanism

In the context of IEEE C37.118.2, synchrophasor data transfer is handled by exchanging four message types of: (1) data, (2) configuration, (3) header and (4) command message.

These messages are exchanged between a PMU/PDC (server) and applications receiving the synchrophasor data (client). 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 the left side of Fig. 4, a C37.118.2 message contains (i) common (highlighted in blue) and (ii) message-specific words (differentiated in green).

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) second-of-century (SOC) is a count of seconds from UTC midnight (00:00:00) of January 1, 1970, to the current second, (5) fraction-of-second (FRACSEC) contains a time quality byte and the number of fraction of seconds and (6) CHK is used to verify that the transmitted data has not been corrupted.

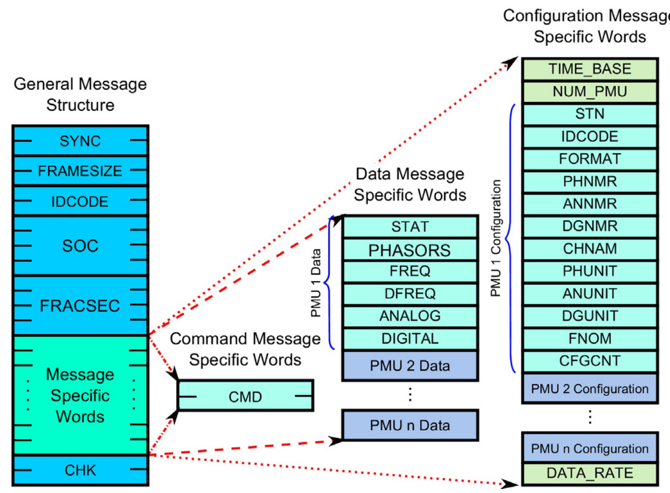

Fig. 4. IEEE C37.118.2 command, data and CFG-2 message specifications.

In C37.118.2, the measurement timestamp is defined as the sum of the number of seconds determined by SOC and fraction of a second specified by FRACSEC/TIME_BASE. In this definition, each second is divided into an integer number of subdivisions determined by the TIME_BASE parameter that is provided in the configuration message. The FRACSEC count is an integer representing the number of subdivisions out of the maximum subdivisions defined by TIME_BASE.

In the right hand side of Fig. 4, the message-specific words associated with command, data and configuration (Type 2) messages are shown. In normal operation, the PMU or PDC continuously streams the data messages containing the synchrophasor measurements made by the PMU.

The data-message specific words transmitting one PMU data includes: (1) STAT flag that gives the status of the data contained in that block, (2) PHASORS that hold the voltage and current 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 that 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 that transmits user-defined binary data such as bit mapped status or flag. When transmitting streams from multiple PMUs in one data message (i.e. data stream from PDC), these words are repeated for each PMU.

In order to enable the receiver to interpret the data messages and extract the measurement values, three types of configuration messages are defined in the standard: (i) configuration type 1 (CFG-1), (ii) configuration type 2 (CFG-2) and (iii) configuration type 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 [14]. 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 C37.118.2, CFG-3 is introduced as optional.

In this work, the configuration of a PMU or PDC is acquired by receiving and parsing the CFG-2 message, hence this message is addressed. As shown in Fig. 4, configuration-message specific words of CFG-2 messages hold the following fields:

(1) TIME_BASE: that gives the resolution of FRACSEC time stamp, (2) NUM_PMU: that defines the number of PMUs in the data message, (3) STN: 16-byte char type for station name, (4) IDCODE: that defines the data source ID number identifying source of each

data block, (5) FORMAT: that defines the data format within the data message (i.e. *Rectangular* or *Polar*, *16-bit Integer* or *32-bit IEEE floating-point*), (6) PHNMR: number of phasors, (7) ANNMR: number of analog values, (8) DGNMR: number of digital status words, (9) CHNAM: that presents the 16-byte char type phasor and channel name for each phasor, analog, and each digital channel (16 channels in each digital word), (10) PHUNIT: conversion factor for phasor channels, (11) ANUNIT: conversion factor for analog channels, (12) DGUNIT: mask words for digital status words, (13) FNOM: that identifies nominal frequency and (14) CFGCNT: as the configuration change count.

When data from multiple PMUs are within a single stream, the sequence of words from (3) STN to (14) CFGCNT are repeated in the same order for the other PMUs. The last word is then DATA_RATE indicating the rate of data transmissions.

As it is also shown in Fig. 4, the command-message specific word contains only a 2-byte word. The following hexadecimal values define the commands being sent to the PMU/PDC: (1) 0x0001: turn-off data transmission, (2) 0x0002: turn-on data transmission, (3) 0x0003: send header message, (4) 0x0004: send CFG-1 message, (5) 0x0005: send CFG-2 message and (6) 0x0006: send CFG-3 message.

The header message provides user defined descriptive information sent from the PMU or PDC.

2.1. Underlying communication mechanism

In IEEE 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 (IP)*) are suggested. The Serial communication protocol is out-dated and it is not applicable for wide-area communications, and in practice TCP/IP or UDP/IP protocols are used for synchrophasor data transmission.

3. Understanding synchrophasor data modeling and communication in IEC 61850-90-5

Unlike the IEEE C37.118.2 data transfer protocol, IEC 61850 is not only a communication standard. IEC 61850 was introduced as an international standard for substation automation, supporting interoperability of IEDs from different manufacturers. This standard has been extended beyond the substation to other domains, such as DERs, hydroelectric power plants and recently (in part 90-5) to wide-area transmission of synchrophasor information according to IEEE C37.118. Therefore, in the title of the second edition of the standard the term “Substation Automation” has been renamed to “Power Utility Automation”.

In addition to *Communication Services*, IEC 61850 introduces a standardized methodology for *Data Modeling* of the IEDs functions in the grid.

3.1. Data modeling

The IEC 61850 data model starts with a *Physical Device*, that is an IED. Each *Physical Device* may contain one or more *Logical Device*, for instance the PMU function in a protection relay. Each *Logical Device*, as an entity representing a set of typical substation functions, consists of a number of *Logical Nodes*. Each *Logical Node* that is an entity representing one typical substation function, 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).

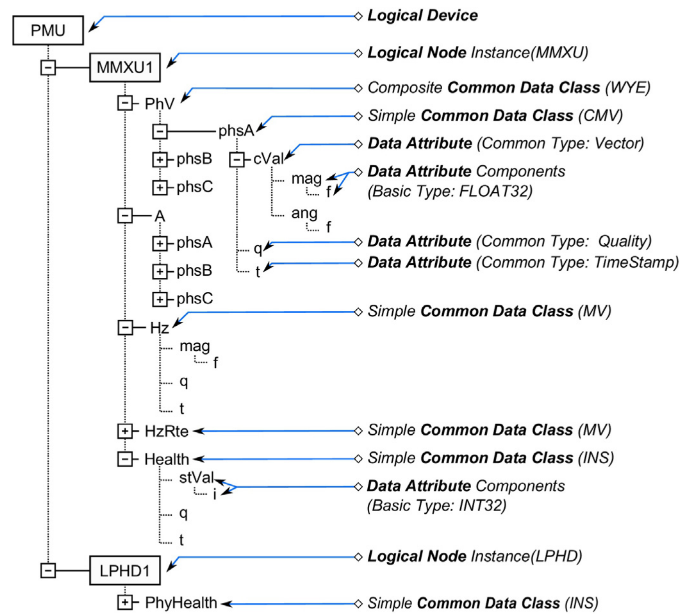

Fig. 5. PMU data model.

In this context, PMU is modeled as a Logical Device within an IED, calculating and publishing synchrophasor measurements as defined in IEEE C37.118.

In IEC 61850-7-4, the measurement Logical Node (MMXU) is defined for currents, voltages, powers and impedances in a three-phase system. In section 13.3 of [7], 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.

By introduction of IEC 61850-90-5, the new data object of HzRte was added to the MMXU Logical Node to accommodate the ROCOF data. Then the PHASORS, FREQ 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. 5.

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 semantics.

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

In this work, using the data model presented in Fig. 5, a data set is defined to transfer the entirety of the PMU data received in the IEEE C37.118.2 datagram, using either Routed-Sampled Value or Routed-GOOSE communication services.

3.2. Communication services

Among five communication mechanisms defined in IEC 61850, Sampled Value (SV), generic substation state event (GSSE) and GOOSE services are utilized for transfer of data in time-critical functions.

The SV service introduced in IEC 61850-9-2 is utilized for fast and cyclic transmission of raw data generated by measurement equipment inside the substation and the GOOSE service defined in IEC 61850-8-1 is used for event-based transfer of data. However, both GOOSE and SV services, mapped directly to the data-link layer, are not suitable for data transfer outside local networks in substation.

In IEC 61850-90-5, two options are proposed for transferring synchrophasor data in wide-area networks:

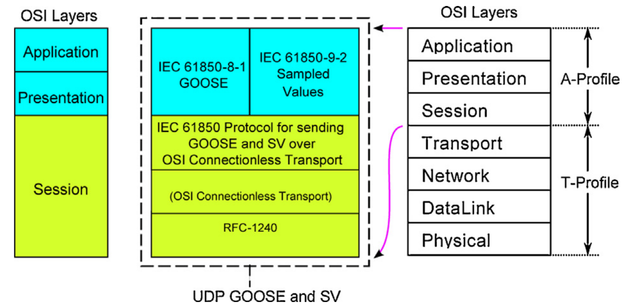

Fig. 6. OSI reference model and R-SV/R-GOOSE application profiles.

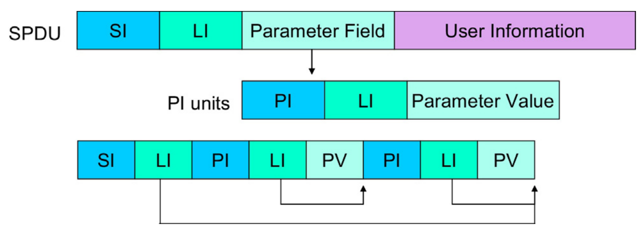

Fig. 7. Session protocol structure.

- *Tunneling* the SV and GOOSE services across some high speed communication networks like SDH or SONET.
- *Internet protocols (IP) networks*; in which, SV and GOOSE services are communicated via IP networks.

For the second option, IEC 61850 has been enhanced by mapping 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.

The new mapping of the SV and GOOSE services uses routable UDP, and are called Routed-Sample Value (R-SV) and Routed-GOOSE (R-GOOSE). The scope of this work is the design of a library and its software implementation to utilize R-SV and R-GOOSE services.

3.3. IEC 61850-90-5 session protocol specification

In IEC 61850-90-5, the application layer specifications of IEC 61850-8-1 GOOSE and IEC 61850-9-2 SV services remained unchanged, however a new protocol is introduced in session layer for sending the GOOSE and SV over OSI connectionless transport.

The seven layers of the open system interconnect (OSI) model, illustrated in Fig. 6, are divided into two profiles of *Application* (A-profile) and *Transport* (T-Profile). The R-GOOSE and R-SV services are in fact an implementation of the A-profile based on the requirements of IEC 61850-90-5.

Each data packet generated at the session layer, called session protocol data unit (SPDU), starts with a single-byte session identifier (SI), followed by a single-byte Length. As depicted in Fig. 7, this length covers the length of all of the parameter fields of the session header, but not the user data of the session protocol. The parameters field includes one or more units of parameters. Each parameter unit starts with a parameter identifier (PI) and its length identifier (LI) followed by the parameter value (PV).

The A-profile and T-profile encoding specification of IEC 61850-90-5 Routed-SV and Routed-GOOSE services, are presented in detail in Fig. 8.

Based on the RFC-1240 protocol (OSI connectionless transport services on top of UDP), the A-profile starts with the length identifier (LI) =0x01 and the transport identifier (TI)= 0x40 on top of the session layer.

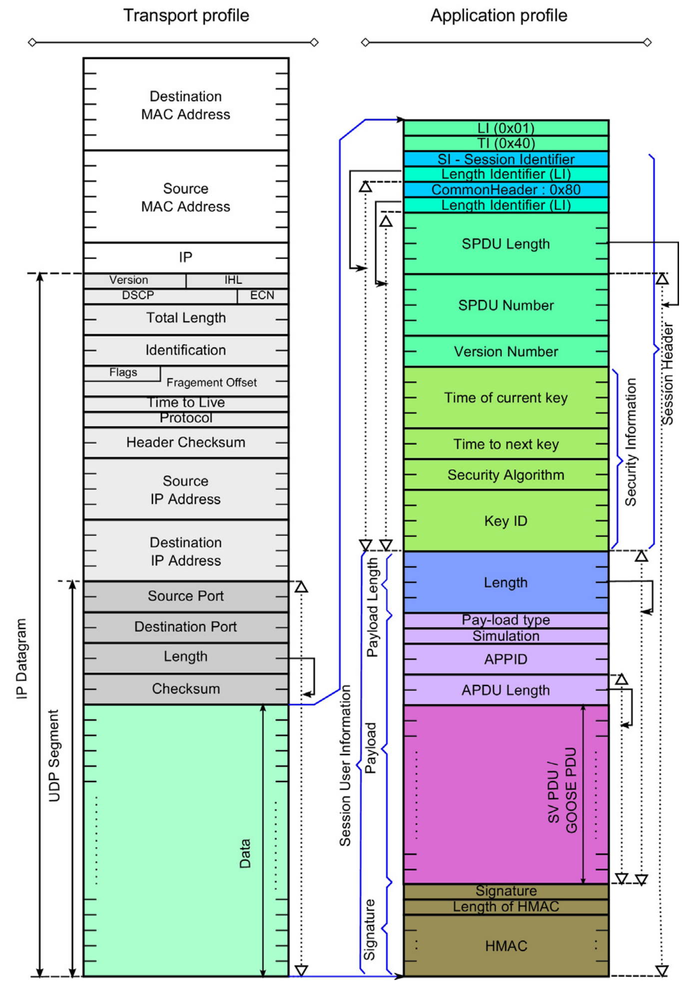

Fig. 8. IEC 61850-90-5 R-SV/R-GOOSE message specification.

Compliant with the session protocol described in Fig. 7, there are four hexadecimal values of the session identifier (SI) defined in IEC 61850-90-5; 0xA0: tunneled GOOSE and sampled value packets, 0xA1: non-tunneled GOOSE application protocol data units (APDUs), 0xA2: non-tunneled SV APDUs and 0xA3: non-tunneled management APDUs.

As shown in the right side of Fig. 8, the SI and associated LI fields are followed by the common session header as the PI with the hexadecimal value of 0x80. The length values, provided by LIs and associated respectively with SI and common header, are indicated using arrows at the left side of session header frames in the A-profile of Fig. 8. The parameter value (PV) of the session header is a sequence of the following values:

(1) *SPDU Length*: fixed size 4-byte word with maximum value of 65,517. As the indicating arrow at the right side of A-profile column in Fig. 8 shows, the value indicated by SPDU length starts from and contains the SPDU number field until the end of signature words, (2) *SPDU Number*: fixed size 4-byte unsigned integer word used by the subscriber to detect duplicate or out-of-order packet delivery, (3) *Version Number*: fixed size 2-byte unsigned integer attribute containing the session protocol version number that is assigned to 1 in this standard and (4) security information.

For the application layer data, encapsulated in the session protocol, security management is provided by the specific key distribution center (KDC) protocol of IEC 61850-90-5. The security information includes: (4.1) *Time of Current Key*: fixed size 4-byte unsigned integer value of the number of seconds since epoch, (4.2) *Time to Next Key*: fixed size 2-byte signed integer value representing the number of minutes prior to a new key being used, (4.3) *Security Algorithms*: fixed size 2-byte attribute indicating the type of encryption and hashed message authentication code (HMAC) algorithm information regarding the signature generation, (4.4) *Key ID*: fixed size 4-byte attribute assigned by the KDC.

The session header is followed by session user information, consisting of: (1) *Payload Length*, (2) *Payload* and (3) *Signature* fields.

Payload length is a fixed size 4-byte unsigned integer with maximum value of 65,399 and as its indicating arrow in Fig. 8 shows, this field covers the length of all session user information except the signature words. In R-SV and R-GOOSE messages, the payload starts with payload attributes of: (1) *Payload Type*, (2) *Simulation*, (3) *APPID* and (4) *APDU Length* words, and ends by sampled value protocol data unit (SV PDU) or GOOSE protocol data unit (GOOSE PDU).

Payload type specifies the mapping protocol of the payload. Four payload types are defined in IEC 61850-90-5; 0x81: non-tunneled GOOSE APDUs, 0x82: non-tunneled SV APDUs, 0x83: tunneled GOOSE and SV packets and 0x84: non-tunneled management APDUs.

Simulation is a one byte Boolean word. Its “true” value indicates that the payload is sent for test.

Application identification (APPID) is a 2-byte word as defined in IEC 61850-8-1 to distinguish the application association.

APDU Length is a fixed size 2-byte unsigned integer indicating the length of the SV or GOOSE PDU plus the APDU Length field itself.

The payload continues with GOOSE PDU or SV PDU. The GOOSE PDU, defined in IEC 61850-8-1, and the SV PDU, defined in IEC 61850-9-2, are explained in detail in the following subsections.

The signature production starts with a one-byte 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 byte 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, hence no signature is generated and the value of the length is considered as zero.

3.3.1. *GOOSE protocol data unit (GOOSE PDU) specification*

The data contained in GOOSE PDU is encoded using OSI’s abstract syntax notation one (ASN.1), basic encoding rules (BER).

ASN.1 is an international standard for data networks and open system communications. Based on this standard, every component of data is presented in the form of tag-length-value (T-L-V) structure where (i) *tag* indicates the type of information represented by the frame, (ii) *length* defines the length of value in the form of number of bytes and (iii) *value* contains the actual data to be specified. The data inside the value is consistent with the type specified by the tag word. In addition, each value word may contain several other data encoded in the form of other TLVs.

The GOOSE message is not encoded exactly based on the original ASN.1/BER, but it uses a version of ASN.1/BER that is adapted to manufacturing message specification (MMS) protocol.

The complete detailed structure of the GOOSE PDU, based on the ASN.1 BER, is illustrated in Fig. 9.

The whole GOOSE PDU is T-L-V encoded, with tag equal to 0x61. The tag is followed by the Length indicating the whole length of the GOOSE PDU. The value part of the GOOSE PDU contains a sequence of data as listed below:

(i) *GoCBRef* (GOOSE Control Block Reference): This visible-string field with maximum size of 65 bytes is the reference to the GOOSE control block that is controlling the GOOSE message. This field is encoded with tag equal to 0x80.

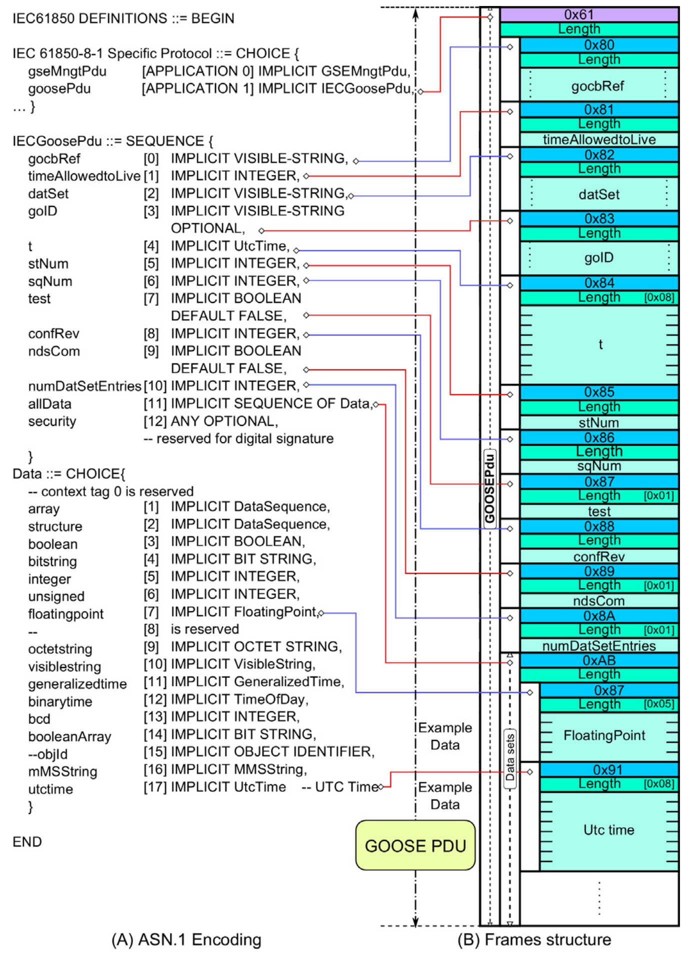

Fig. 9. IEC 61850-8-1 GOOSE PDU structure.

(ii) *TimeAllowedtoLive*: Because GOOSE messages have a re-transmission process, each message carries a TimeAllowed-ToLive 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. This Integer value indicates the number of milli seconds. This field is encoded with tag equal to 0x81.

(iii) *datSet*: This visible-string field with maximum size of 65 bytes, contains the value of the data set reference, as found in the GOOSE control block specified by GoCBRef. This field is encoded with tag equal to 0x82.

(iv) *goID*: This optional visible-string field with maximum size of 65 bytes, specifies the GOOSE ID, as found in the GOOSE control block specified by GoCRef. This field is encoded with tag equal to 0x83.

(v) *t*: This is the 8-byte UTC time stamp of the GOOSE message. This field is encoded with tag equal to 0x84.

(vi) *stNum* (State Number): This integer value contains the counter that increments each time a GOOSE message is sent and a value change is detected within the data set specified by *datSet*. This field is encoded with tag equal to 0x85.

(vii) *sqNum* (Sequence Number): For this integer field, 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. This field is encoded with tag equal to 0x86.

(viii) *test*: This Boolean value with ranges of TRUE or FALSE is encoded with tag equal to 0x87.

Table 1
Tag value in MMS data value encoding of GOOSE PDU data set.

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
13	Utctime	0x91

(ix) *ConfRev* (Configuration Revision): This integer value counts the number of times that the configuration of data set referenced by *datSet* has been changed. This field is encoded with tag equal to 0x88.
(x) *ndsCom* (Needs Commissioning): This Boolean value contains the attribute NdsCom from the GOOSE control block. This field is encoded with tag equal to 0x89.
(xi) *numDatSetEntries*:This Integer parameter specifies the number of members of data set in the GOOSE message. This field is encoded with tag equal to 0x8A.
(xii) *allData*: The last field in the sequence, is the data set that is going to be transmitted by means of GOOSE PDU. This field is encoded with tag equal to 0xAB.

Any data to be sent in the data set of the GOOSE PDU is encoded based on the MMS adapted ASN.1/BER encoding rule as listed in Table 1 [15]. Using the R-GOOSE service, the PMU data are transferred as the data set of the GOOSE PDU.

The GOOSE PDU data set contains the tag-length-value (T-L-V) encoded elements of the data. Based on the type, each component of data will be tagged according to Table 1. For instance, the floating-point type magnitude component of the complex value of phase A voltage in an MMXU LN represented as MMXU1.PhV.PhsA.cVal.mag.f is tagged with hexadecimal value of 0x87. The data set encoding structure of a GOOSE PDU holding a PMU data is shown in Fig. 11A.

3.3.2. SV protocol data unit (SV PDU) specification

Based on the ASN.1/BER encoding rules, as it is shown in Fig. 10, the SV PDU is T-L-V encoded with the tag value equals to 0x60. The tag is followed by the length indicating the length of the SV PDU. The value part of the SV PDU contains a sequence of data as listed below:

(1) *noASDU*: Each SV APDU is transmitting one or multiple consecutive samples of data in SV application service data unit (ASDU). This integer value gives the number of ASDUs which will be concatenated into one APDU and is encoded with tag value of 0x80.
In this work, one ASDU is transmitted in each SV APDU.
(2) *Sequence of ASDUs*:All sequence of ASDUs associated to the SV PDU are encoded with the tag value equal to 0xA2.

Each ASDU is encoded with tag value of 0x30. The data associated to each ASDU is described below:

(i) *MsvID*: This visible-string field, containing the system-wide unique identification of the ASDU, is encoded with tag equal to 0x80.

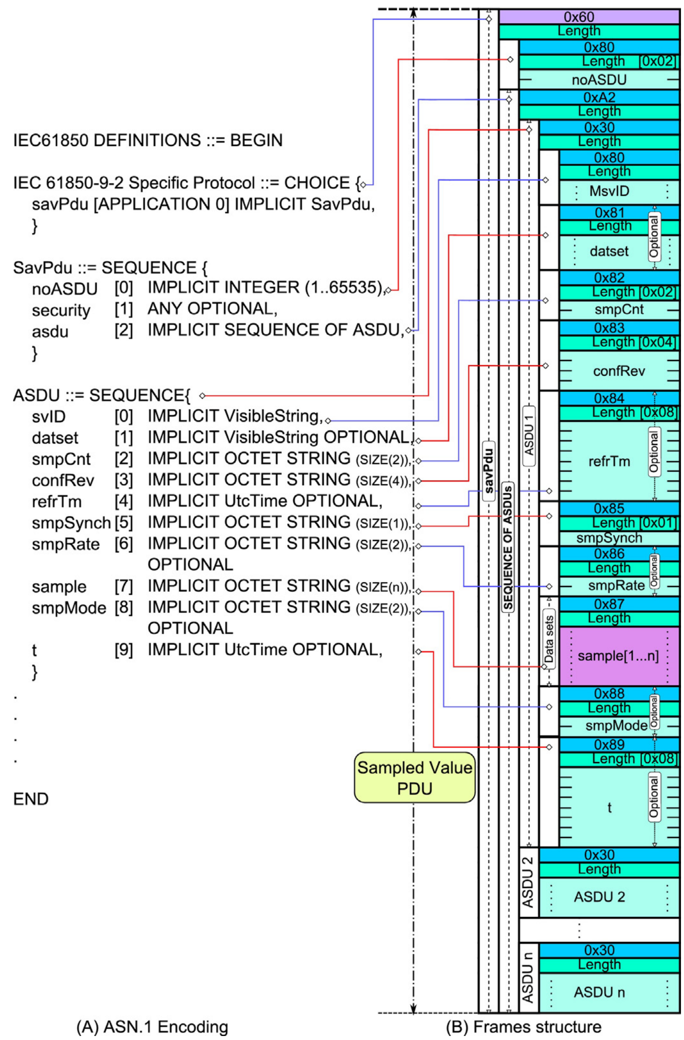

Fig. 10. IEC 61850-9-2 SV PDU structure.

(ii) *datSet*: This optional visible-string field, represents the data set reference defined in the Sampled Value Control Block. This field is encoded with tag equal to 0x81.
(iii) *smpCnt* (Sample Count): This fixed size 2-byte unsigned integer value, contains the value of sample count which will be incremented each time a new sampling value is taken. This field is encoded with tag equal to 0x82.
(iv) *ConfRev* (Configuration Revision): This 32-bit unsigned integer value indicates the count of configuration changes with regard to Sampled Value Control Block. This field is encoded with tag equal to 0x83.
(v) *refrTm* (Refresh Time): This optional field is the 8-byte UTC time stamp of the refresh time of SV Buffer. This is the measurement time of the synchrophasor data as defined in IEEE C37.118.1, encoded with tag equal to 0x84.
(vi) *smpSynch* (Samples Synchronized): This Boolean value indicates if the samples are synchronized by a clock signal and encoded with tag equal to 0x85.
(vii) *smpRate* (Sample Rate): This fixed size 2-byte unsigned integer value indicates the value of sampling rate and is encoded with tag equal to 0x86.
(viii) *Sample*: This field is the list of data values related to the data set that is going to be transmitted using the SV PDU. The whole field including the data set values is encoded with tag equal to 0x87.

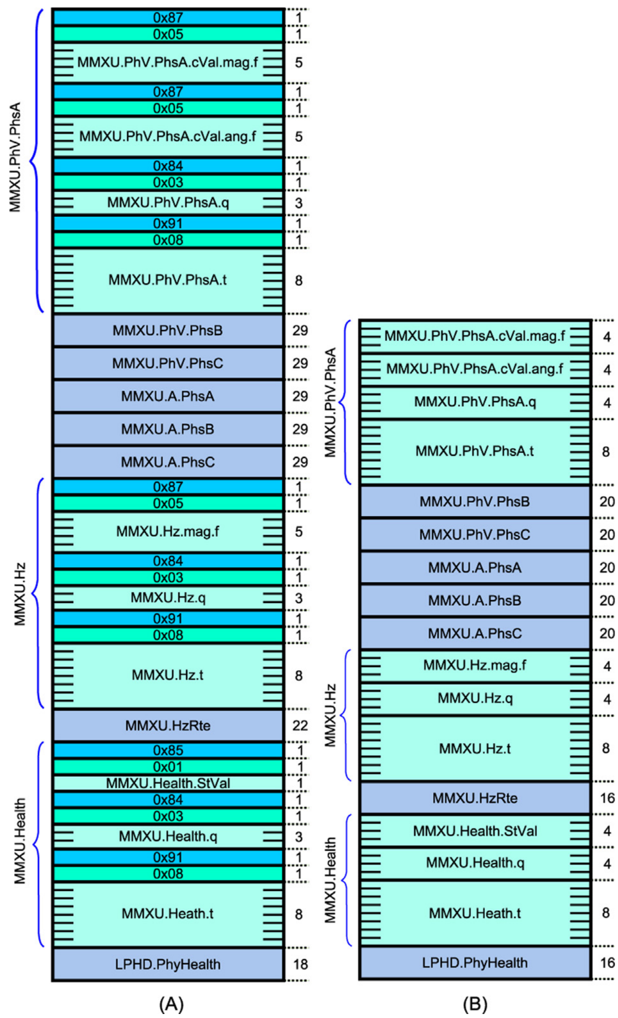

Fig. 11. Data set specification in (A) GOOSE PDU and (B) SV PDU.

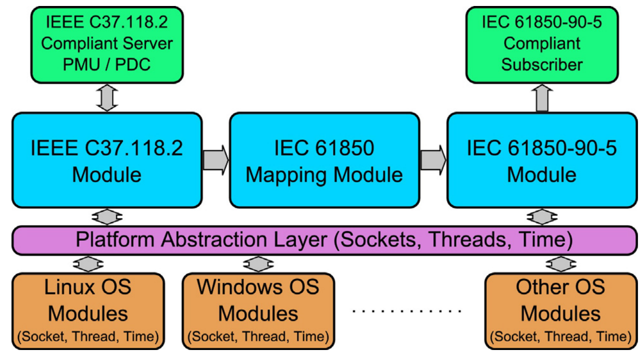

Fig. 12. Khorjin Gateway library architecture.

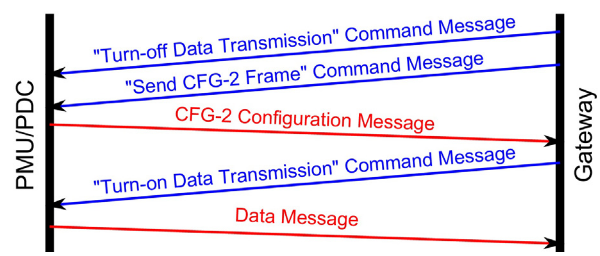

Fig. 13. Gateway communication mechanism with PMU/PDC based on IEEE C37.118.2.

Unlike the GOOSE PDU, the data set of the SV PDU is not T-L-V encoded. For encoding of the SV PDU data set values, the rules of IEC 61850-9-2 encoding is applied. In this case, based on the guidelines presented in [16], the data set components are encoded consecutively in their basic forms. The SV PDU data set specification of a PMU data is illustrated in Fig. 11B.

In IEC 61850-90-5, in order to use the SV service for transmission of synchrophasor data, the following two fields are added to the IEC 61850-9-2 SV APDU.

(ix) *SmpMod* (Sample Mode): This optional fixed size 2-byte unsigned integer value from multicast sampled value control block (MSVCB) indicates if the samples are in samples per nominal cycle, samples per second or seconds per sample. This field is encoded with tag equal to 0x88.

(x) *t*: This optional field is the 8-byte absolute UTC time stamp of transmission time of the packet. This field is encoded with tag equal to 0x89.

4. Description of IEEE-IEC gateway functionality of Khorjin library

The Khorjin library is developed using a modular architecture, enabling its easy future development. As depicted in Fig. 12, based on functional requirements, the Gateway part of Khorjin library is designed and implemented in three main components of:

1. IEEE C37.118.2 Module.
2. IEC 61850 Mapping Module.
3. IEC 61850-90-5 R-SV/R-GOOSE Publisher Module.

In order to be platform-independent, a platform abstraction layer is implemented in the library. Using this approach the platform-dependent modules (sockets, threads and time) are separated from the main modules of the library.

4.1. IEEE C37.118.2 module

In this module, the IP address, port number and IDCODE of the server PMU/PDC are the required inputs for real-time exchange of C37.118.2 compliant data over a TCP/IP connection between PMU/PDC and the gateway.

Similar to the algorithm presented in [17], as it is shown in Fig. 13, after establishing a TCP/IP connection between a PMU/PDC (Server) and the Gateway (Client), synchrophasor data transfer is performed by executing the following procedure: (1) The Gateway sends a “Turn-off data message transmission” command, then the PMU/PDC processes the received command and stops the transmission of data messages, (2) the Gateway sends a “Send CFG-2 message” command to the PMU/PDC, (3) in reply to the received command, the PMU/PDC sends the CFG-2 configuration message to the Gateway, (4) having the CFG-2 message received and parsed, the Gateway sends a “Turn-on data message transmission” command message to the PMU/PDC and (5) the PMU/PDC processes the received command message and starts streaming the data messages.

4.2. IEC 61850 mapping module

In this module, the mapping of IEEE C37.118.2 PMU data into the IEC 61850 data model is implemented for (1) synchrophasor values, (2) time stamps and (3) quality data objects.

Table 2
IEEE C37.118.2 to IEC 61850-90-5 mapping specifications of synchrophasor data.

IEEE C37.118.2		IEC 61850-90-5
CFG-2 message	Data message	
FORMAT (bits 0-1) PHNMR PHUNIT	PHASORS	Data attributes of "PhV" and "A" data objects in MMXU logical node. MMXU1.PhV.PhsA.cVal.mag.f MMXU1.PhV.PhsA.cVal.ang.f MMXU1.PhV.PhsB.cVal.mag.f MMXU1.PhV.PhsB.cVal.ang.f MMXU1.PhV.PhsC.cVal.mag.f MMXU1.PhV.PhsC.cVal.ang.f MMXU1.A.PhsA.cVal.mag.f MMXU1.A.PhsA.cVal.ang.f MMXU1.A.PhsB.cVal.mag.f MMXU1.A.PhsB.cVal.ang.f MMXU1.A.PhsC.cVal.mag.f MMXU1.A.PhsC.cVal.ang.f
FORMAT (bit 3) FNOM	FREQ	Data attribute of "Hz" data object in an instance of MMXU logical node MMXU1.Hz.mag.f
FORMAT (bit 3)	DFREQ	Data attribute of "HzRte" data object in an instance of MMXU logical node. MMXU1.HzRte.mag.f
FORMAT (bit 2) ANNMR ANUNIT	ANALOG	Appropriate data objects in relevant logical node. For example: Total active or reactive power analog values are mapped to "TotW" and "TotVAr" data objects in MMXU logical node: MMXU1.TotW.mag.f MMXU1.TotVAr.mag.f
DGNMR DGUNIT	DIGITAL	Appropriate data objects in relevant logical node. For example: Circuit Breaker status flag bits are mapped to data objects in XCBR logical node: myXCBR1.Pos.stVal

4.2.1. Synchrophasor data mapping

As mentioned before, IEEE C37.118.2 data messages holding PMU data are interpreted by parsing the configuration message type 2 (CFG-2) at the Gateway.

Using the data of FORMAT (bits 0-1), PHNMR and PHUNIT fields in the CFG-2 message, the values of voltage and current phasors in the PHASORS field of the IEEE C37.118.2 data message are mapped to data attributes of "PhV" and "A" data objects in the MMXU logical node, as the equivalent IEC 61850 data model.

For example MMXU1.A.PhsA.cVal.mag.f and MMXU1.A.PhsA.cVal.ang.f represent the floating-point (f) magnitude (mag) and angle (ang) values of the complex value (cVal) of phase A (PhsA) current (A) in an instance of the MMXU logical node (MMXU1).

Data from FORMAT (bit 3) and FNOM words in CFG-2 message, enables the frequency and ROCOF values in FREQ and DFREQ fields of IEEE C37.118.2 data message to be mapped to MMXU1.Hz.mag.f and MMXU1.HzRte.mag.f indicating the floating-point (f) magnitude (mag), data attribute of frequency (Hz) and ROCOF (HzRte) data objects in an instance of MMXU logical node (MMXU1), as shown in Table 2.

The ANALOG field of the IEEE C37.118.2 data message is parsed using the data in FORMAT (bit 2), ANNMR and ANUNIT fields in CFG-2 message and are mapped to appropriate data objects in the relevant logical node. For example when the total active and reactive power values are in the ANALOG field, they are mapped to MMXU1.TotW.mag.f and MMXU1.TotVAr.mag.f indicating the floating-point (f) magnitude (mag) data attribute of (TotW) and (TotVAr) data objects in an instance of the MMXU logical node (MMXU1).

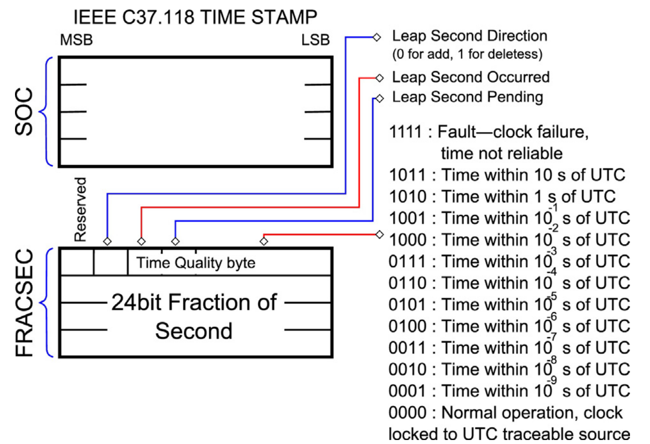

Fig. 14. IEEE C37.118.2 time stamp structure.

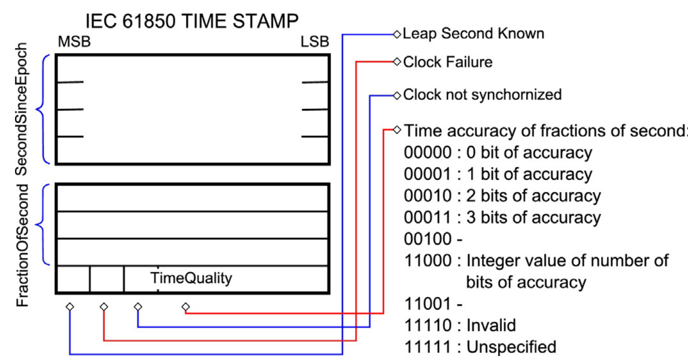

Fig. 15. IEC 61850 time stamp structure.

The digital state values in the DIGITAL field of IEEE C37.118.2 data message are extracted using the DGNMR and DGUNIT fields from the CFG-2 message and are mapped to appropriate data objects in the relevant logical node. For example a circuit breaker status flag bit is mapped to XCBR1.Pos.stVal indicating the status value (stVal) data attribute of position (Pos) data object in an instance of XCBR logical node (XCBR1).

4.2.2. Time stamp mapping

In IEEE C37.118.2, the SOC and FRACSEC fields in the data message and TIME_BASE in CFG-2 message, define the time stamp of synchrophasor data. The structure of the IEEE timestamp is depicted in Fig. 14, showing that the most significant byte in FRACSEC holds the time quality byte.

In IEC-61850-7-2, the TimeStamp is defined as a data object including *SecondSinceEpoch*, *FractionOfSecond* and *TimeQuality* data attributes. The IEC 61850-8-1 mapping specification of this data object is shown in Fig. 15. In this structure, the first 4 most significant bytes (bytes 1-4) hold the SecondSinceEpoch, bytes 5-7 specify the FractionOfSecond and the last byte is TimeQuality.

Similiar to SOC in IEEE, the SecondSinceEpoch in IEC indicates the time in seconds continuously counted from the epoch 1970-01-01 00:00:00 UTC.

However, the representation of fraction of second differs in the two standards. In IEEE, each second is divided into an integer number of subdivisions specified by the TIME_BASE field in the configuration message and the 24-bit integer FRACSEC count represents the numerator of FRACSEC with TIME_BASE as the denominator ($FRACSEC/TIME\_BASE$).

In this implementation, the TIME_BASE field in the IEEE CFG-2 message is mapped to the "TimeAccuracy" data attribute of TimeQuality in the IEC TimeStamp. It can be seen in Fig. 15 that

Table 3
IEEE C37.118.2 to IEC 61850 time stamp mapping specification.

IEEE C37.118.2		IEC 61850-90-5
CFG-2 message	Data message	
TIME_BASE	FRACSEC (bits 24-27)	"TimeAccuracy" attribute of "TimeQuality" data attribute in TimeStamp data object (bits 3-7 (Time accuracy), Maximum: 11,000. $1/2^{24} = 1/16{,}777{,}216 \simeq 60$ ns)
	SOC	"SecondSinceEpoch" data attribute in TimeStamp data object
	FRACSEC (bits 0-23)	"FractionOfSecond" data attribute in TimeStamp data object
	FRACSEC (bits 24-27 =1111)	"TimeQuality" data attribute in TimeStamp data object (bit 1 (Clock Failure))

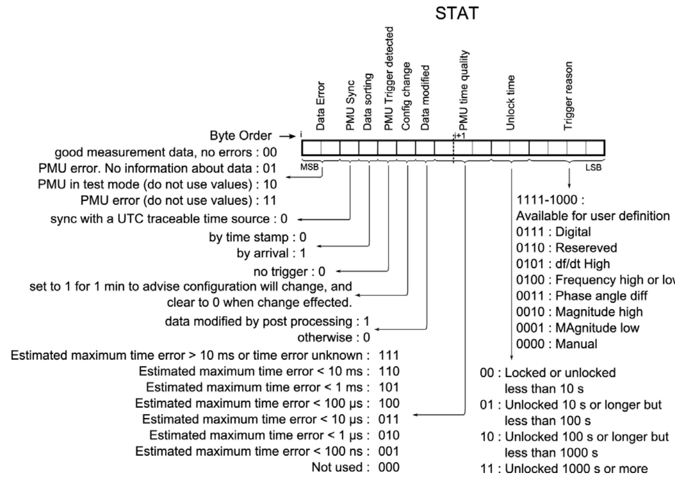

Fig. 16. STAT word in IEEE C37.118.2 data message.

TimeAccuracy is the 5 least significant bits of TimeQuality byte with maximum value of 11000 indicating that the maximum resolution of IEC 61850 TimeStamp is $1/2^{24} = 1/16{,}777{,}216 \simeq 60$ ns.

In addition, the IEC 61850 FractionOfSecond is specified by the three most significant bytes after the SecondSinceEpoch. The value of the fraction field is derived by numbering the bits of these bytes, starting with the most significant bit of the most significant byte as bit 0 and ending with the least significant bit of the least significant byte (3rd Octet) as bit 23. Each bit is assigned a numerical value of 2^{-n}, where n is the position of the bit in this numbering sequence. As it is shown in following formula, the value of fraction of second is obtained by summing the numerical values assigned to each bit for those bits which are set to one (1).

$$FractionOfSecond = \sum_{n=0}^{23} b_n . 2^{-n} \quad (1)$$

The ClockFailure data attribute of TimeQuality is also set true if the least four bits in IEEE TimeQuality become equal to 1111.

The specification of the time stamp mapping, implemented in this work, is presented in Table 3.

4.2.3. Mapping STAT word

In the IEEE C37.118.2 data message, as it is illustrated in Fig. 4, each PMU data stream contains a 16-bit STAT word specifying information about the status of the data stream. This field is described in detail in Fig. 16.

In the IEC 61850 data model, "Quality" is an attribute that contains information on the quality of the information from the server. The IEC-61850-8-1 bit-string representation of the Quality attribute is described in Table 4.

In this implementation, the information provided by bits 14-15 (data error) of the STAT word is mapped to bits 0-1 (validity) and bit 11 (test) of quality field. The detail of this mapping is provided in Table 5.

In addition, in IEC 61850, "PhyHealth" data object reflects the state of the health of the LPHD logical node related to hardware. The "stVal" data attribute of "PhyHealth" specifies status of the PMU in either of the following conditions: (1) Ok (green) no problem, normal operation, (2) Warning (yellow) minor problems, but in safe operation mode and (3) Alarm (red) severe problem, no operation possible. The specification of the mapping to this logical node is presented in Table 5.

In this work, some data fields in the STAT word are mapped to the appropriate IEC 61850 data objects, however it is not possible for all data in each STAT word to be mapped to IEC 61850 data model. This is because there is no equivalent in IEC 61850 for every information provided in the bit flags of the STAT word. In order to address this problem, in [18], an implementation agreement is specified to include the whole STAT word as a 16-bit Bit-String. Hence in this implementation the STAT word is also transferred as a 16 bit-string in the data set.

Table 4
IEC-61850-8-1 bit-string representation of quality attributes.

Bits	Attribute name	Attribute value
0–1	Validity	Good(00)/Invalid(01)/ Reserved(10)/Questionable(11)
2	Overflow	TRUE(1)/FALSE(0)
3	OutofRange	TRUE(1)/FALSE(0)
4	BadReference	TRUE(1)/FALSE(0)
5	Oscillatory	TRUE(1)/FALSE(0)
6	Failure	TRUE(1)/FALSE(0)
7	OldData	TRUE(1)/FALSE(0)
8	Inconsistent	TRUE(1)/FALSE(0)
9	Inaccurate	TRUE(1)/FALSE(0)
10	Source	Process(0)/Substituted (1)
11	Test	TRUE(1)/FALSE(0)
12	OperatorBlocked	TRUE(1)/FALSE(0)

Table 5
Mapping specification of the STAT word.

IEEE C37.118.2 Data message	IEC 61850-90-5
STAT (bits 14-15 (Data Error) =01)	Quality (bit 11(test) = FALSE, bits 0-1(Validity)= 11(Questionable)) "PhyHealth" data object in LPHD1 ("stVal" = 3) LPHD1.PhyHealth.stVal
STAT (bits 14-15 (Data Error) =10)	Quality (bit 11(test) = TRUE, bits 0-1(Validity)=01(Invalid))
STAT (bits 14-15 (Data Error) =11)	Quality (bit 11(test) = FALSE, bits 0-1(Validity)=01(Invalid)) "PhyHealth" data object in LPHD1 ("stVal" = 3) LPHD1.PhyHealth.stVal

4.3. IEC 61850-90-5 publisher module

The IEC 61850-90-5 publisher module is capable of generating the R-SV/R-GOOSE traffic over unicast or multicast UDP/IP. For this purpose, the IP address (for unicast UDP/IP), Port number (in [7] it is set to 102) and APPID are required as inputs. The implementation of the R-GOOSE publisher core of this module was started by studying the source codes of the "libiec61850" library for IEC 61850 MMS/GOOSE communication protocols [19], however the

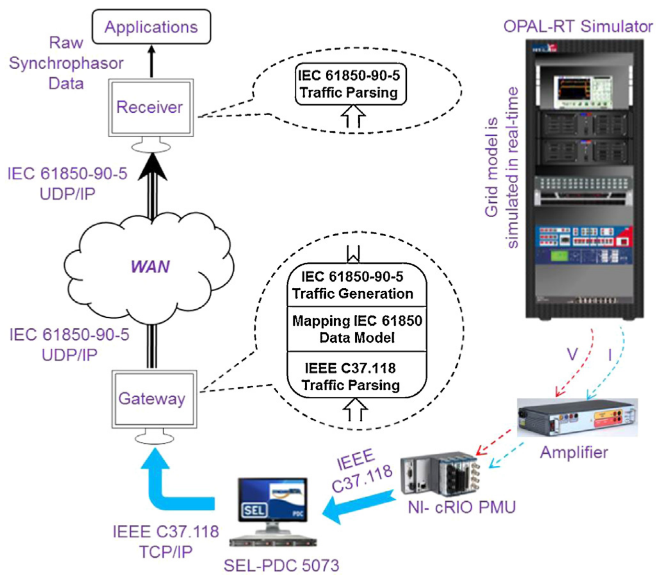

Fig. 17. Hardware-in-the-loop setup for validation experiments.

implementation has been carried out independently and without any dependency to the mentioned library.

5. Performance assessment results

The Khorjin Gateway interacts with real-time data, therefore its functionality requires validation in a real-time hardware-in-the-loop simulation environment.

The conformance of the Khorjin Gateway with the IEEE C37.118.2 standard is verified by successfully connecting and communicating with the SEL-5073 synchroWAVe PDC software (SEL-PDC 5073), compliant with C37.118 [20]. In addition, conformance validation with the requirements of the IEC 61850-90-5 Std is carried out by analyzing the frames captured by the Wireshark network protocol analyzer software [21].

5.1. Real-time hardware-in-the-loop (RT-HIL) test setup

As shown in Fig. 17, a measurement location has been specified on a grid model that is simulated using the OPAL-RT real-time simulator. The measured voltages and currents are fed to a PMU through the analogue output ports of the OPAL-RT simulator. As indicated in the figure, the PMU used in this setup is a Compact Reconfigurable IO system (CRIO) from the National Instruments Corporation [22]. As the figure shows, the signals from the RT simulator are passed through amplifiers before being fed to the PMUs. Synchrophasors are then sent to a PDC that streams the data over TCP/IP to the workstation holding IEEE-IEC Gateway. On another workstation, the receiver part of the library receives the real-time data streams in the IEC 61850-90-5 format and parses the R-SV or R-GOOSE messages.

5.2. Routed-GOOSE and Routed-SV traffic generation validation results

Figs. 18A and 19AA show the captured TCP/IP frame transmitting an IEEE C37.118 data message respectively in R-GOOSE and R-SV traffic generation tests. In these frames, the phasors are transmitted in floating-point and rectangular format as described in Section 2.

It can be seen that the data messages captured in Figs. 18A and 19A are consistent with the C37.118 data message introduced in Section 2 and its frame specification depicted in Fig. 4.

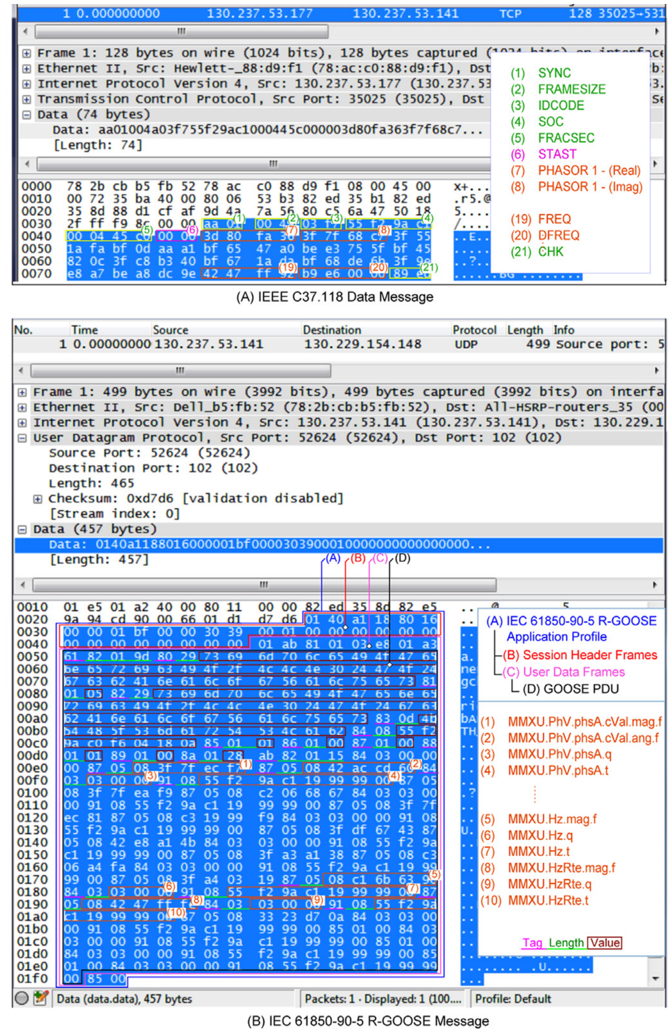

Fig. 18. Wireshark captures analysis in Routed-GOOSE traffic generation test.

The data message starts by common words of: (1) SYNC, (2) FRAMESIZE, (3) IDCODE, (4) SOC, (5) FRACSEC and ends 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 detailed analysis of R-GOOSE message shown in Fig. 18B, confirms its conformity with the specification defined in IEC 61850-90-5 standard. In this figure, the bytes in blue background color constitute the complete R-GOOSE session layer encapsulated in the UDP/IP frame. 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 GOOSE PDU bytes.

The detailed analysis of R-SV message shown in Fig. 19B, confirms its conformity with the specification defined in IEC 61850-90-5 standard. In Fig. 19B, the bytes in the blue background color constitute the complete R-SV 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 SV PDU bytes.

5.3. Performance results

The time delay imposed by the use of the Khorjin Gateway was evaluated within a test performed on a Windows OS computer through 40 experiments in both R-SV and R-GOOSE traffic

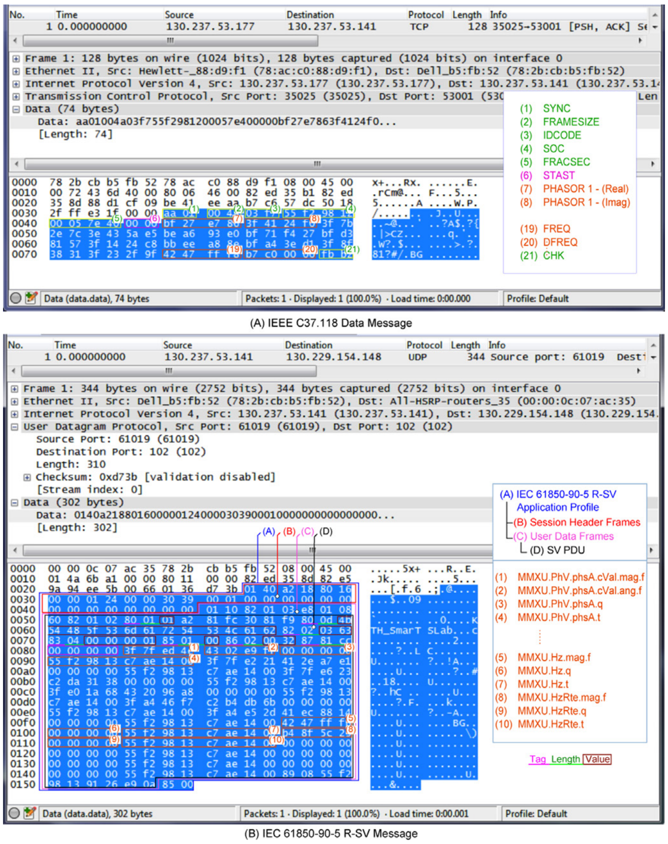

Fig. 19. Wireshark captures analysis in Routed-SV traffic generation test.

Table 6
QoS test of gateway process time performance.

Parameter	R-SV test	R-GOOSE test
Average (ms)	2.833	3.061
Minimum (ms)	2.570	2.656
Maximum (ms)	3.191	3.503
Std. deviation	0.179	0.224

generation cases. The process duration of each cycle in the Gateway was calculated using Wireshark captures. In each test, after 10 seconds of stable operation, the process time of 20 mapped messages is calculated. The results of the quality of service (QoS) test, representing the imposed latency for these 20 samples are presented in Table 6.

6. Conclusion

In this paper, the IEEE C37.118.2 standard, as the most utilized protocol for synchrophasor data transfer, and IEC 61850-90-5, as a new standard for transfer of synchronized phasor measurements over wide-area networks, were described. The functional description of the Khorjin library architecture, implemented in three main components of (1) IEEE C37.118.2, (2) IEC 61850 Mapping and (3) IEC 61850-90-5 publisher modules, were explained in detail.

The comprehensive IEEE C37.118.2 to IEC 61850 mapping specification, being implemented in the Khorjin library based on the IEC 61850-90-5 recommendations, was explained. It was understood that, in spite of mapping almost all critical information in the context of IEEE C37.118.2 to the IEC 61850-90-5 data model, there is information in the IEEE C37.118.2 protocol that remains unmodeled in the IEC 61850-90-5, i.e. there is no specification on how these data should be mapped to the IEC 61850 data model. For instance, the information provided by bit flags of the 16-bit STAT word in the IEEE C37.118 Data message does not have any equivalent in the IEC 61850 data model.

The work in this paper serves as a proof of concept of a gateway capable of translating IEEE C37.118.2 to IEC 61850-90-5. The availability of such gateway and the source code of Khorjin will be instrumental with respect to the existing and future PMU application software.

It is noted that there is a cost of adopting IEC with regard to the size of the payload. The use of R-GOOSE will result in a 6-fold increase, while the use of R-SV will result in a 4-fold increase. A careful analysis of the communication network and data system, transporting and processing this data will be necessary for existing systems. While the data system will have larger performance requirements (e.g. for parsing at the PDC), the communication network will require higher bandwidth allocation.

Concerning future work, the implementation of the security algorithms in the session layer is a necessary step to fully exploit IEC 61850-90-5 features.

Acknowledgements

This work was supported in part by the FP7 IDE4L project funded by the European Commission and by the STandUp for Energy Collaboration Initiative.

References

[1] The Future of the Grid, Evolving to Meet America's Needs, Tech. Rep., GridWise Alliance, US Department of Energy, 2014.
[2] S.F. Bush, Smart Grid: Communication-Enabled Intelligence for the Electric Power Grid, John Wiley & Sons, 2014.
[3] J. Ekanayake, K. Liyanage, J. Wu, A. Yokoyama, N. Jenkins, Smart Grid: Technology and Applications, John Wiley & Sons, 2012.
[4] A.G. Phadke, J.S. Thorp, Synchronized Phasor Measurements and Their Applications, Springer Science & Business Media, 2008.
[5] E. Schweitzer, D. Whitehead, G. Zweigle, K. Ravikumar, Synchrophasor-based power system protection and control applications, in: 2010 63rd Annual Conference for Protective Relay Engineers, 2010, pp. 1–10, http://dx.doi.org/10.1109/CPRE.2010.5469481.
[6] K. Martin, Synchrophasor standards development – IEEE C37.118 & IEC 61850, in: 2011 44th Hawaii International Conference on System Sciences (HICSS), 2011, pp. 1–8, http://dx.doi.org/10.1109/HICSS.2011.393.
[7] Communication Networks and Systems for Power Utility Automation – Part 90-5: Use of IEC 61850 to Transmit Synchrophasor Information According to IEEE C37.118 (2012-05).
[8] H. Falk, M. Adamiak, D. Baigent, V. Madani, An overview of the new IEC 61850 synchrophasor publish-subscribe profile, in: 2013 66th Annual Conference for Protective Relay Engineers, 2013, pp. 309–321, http://dx.doi.org/10.1109/CPRE.2013.6822046.
[9] M. Seewald, Building an architecture based on IP-multicast for large phasor measurement unit (PMU) networks, in: 2013 IEEE PES Innovative Smart Grid Technologies (ISGT), 2013, pp. 1–5, http://dx.doi.org/10.1109/ISGT.2013.6497794.
[10] J.-D. Lee, S.-J. Lee, J.-H. Bae, D.-Y. Kwon, The PMU interface using IEC 61850, in: 2013 International Conference on ICT Convergence (ICTC), 2013, pp. 1125–1128, http://dx.doi.org/10.1109/ICTC.2013.6675573.
[11] X. Yong, Z. Dao-Nong, Y. Yuehai, The research of synchronized phasor measurement units in smart substations, in: 2011 International Conference on Advanced Power System Automation and Protection (APAP), vol. 3, 2011, pp. 2262–2267, http://dx.doi.org/10.1109/APAP.2011.6180804.
[12] Cisco and SISCO Collaborate on Open Source Synchrophasor Framework, Tech. Rep., Cisco Systems, Inc., 2011, URL http://www.cisco.com/web/strategy/docs/energy/cisco-SISCO_factsheet.pdf.
[13] V. Madani, S. Picard, Y. Yin, M. Adamiak, Challenges and lessons learned from commissioning an IEC 61850-90-5 based synchrophasor system, in: 2015 68th Annual Conference for Protective Relay Engineers, 2015, pp. 842–849, http://dx.doi.org/10.1109/CPRE.2015.7102208, URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7102208.
[14] K. Martin, et al., An overview of the IEEE standard C37.118.2 – synchrophasor data transfer for power systems, IEEE Trans. Smart Grid (2014), http://dx.doi.org/10.1109/TSG.2014.2302016.
[15] H. Falk, M. Burns, MMS and ASN.1 Encodings, Simple Samples and Explanations, Systems Integration Specialists Company (SISCO), Inc, 2001 August.
[16] Implementation Guideline for Digital Interface to Instrument Transformers Using IEC 61850-9-2, Tech. Rep., UCA International Users Group, 2004.

[17] L. Vanfretti, I. Al Khatib, M. Almas, Real-time data mediation for synchrophasor application development compliant with IEEE C37.118.2, in: 2015 IEEE Power & Energy Society on Innovative Smart Grid Technologies Conference (ISGT), 2015, pp. 1–5, http://dx.doi.org/10.1109/ISGT.2015.7131910, URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7131910.

[18] NASPI, NASPI Synchrophasor Technical Report – Use of IEC61850-90-5 to Transmit Synchrophasor Information According to IEEE C37.118, Tech. Rep., NASPI, 2014.

[19] [link]. URL http://www.libiec61850.com/libiec61850/.

[20] [link]. URL https://selinc.com/products/5073/.

[21] [link]. URL https://www.wireshark.org/.

[22] P. Romano, M. Paolone, Enhanced Interpolated-DFT for Synchrophasor Estimation in FPGAs: Theory, Implementation, and Validation of a PMU Prototype, 2014, pp. 2824–2836, http://dx.doi.org/10.1109/TIM.2014.2321463.