بسم الله الرحمن الرحيم

Sudan University of Science and Technology

College of Graduate Studies

Implementation of IEC 61850 Based on Substation Automation Systems

تطبيق البروتوكول 61850 على أنظمة التحكم في المحطات

A thesis Submitted in Fulfillment of the Requirements for the degree of M.Sc. in Computer Engineering

By:

Hind Saeed Ali Saeed

Supervisor:

Dr. Rashid Abd Alhaleem

Date December, 2015

الآيه

بسم الله الرحمن الرحيم

خَلْقِ السَّمَاوَاتِ وَالأَرْضِ وَاخْتِلافِ اللَّيْلِ وَالنَّهَارِ وَالْفُلْكِ الَّتِي تَجْرِي
فِي الْبَحْرِ بِمَا يَنفَعُ النَّاسَ وَمَا أَنزَلَ اللَّهُ مِنَ السَّمَاءِ مِن مَّاءٍ فَأَحْيَا بِهِ الأَرْضَ
دَ مَوْتِهَا وَبَثَّ فِيهَا مِن كُلِّ دَابَّةٍ وَتَصْرِيفِ الرِّيَاحِ وَالسَّحَابِ الْمُسَخَّرِ بَيْنَ
السَّمَاءِ وَالأَرْضِ لآيَاتٍ لِّقَوْمٍ يَعْقِلُونَ)

صدق الله العظيم

البقرة (164)

Dedication

I dedicate this thesis to ....

To my father soul ...

To my mother…

To My sisters and my brothers...

To My husband…….

To My Daughter (Rovan)….

Acknowledgement

First and forever, I thank Allah the Owner of the World; I would like to express my appreciation to my supervisor Dr. ***Rashid Abd Alhaleem*** for his guidance, support, and suggestions throughout this research project. I am also grateful for the assistance and encouragement provided by Engineer ***Khaled Alabass***, especial thank to all of my colleagues at SETCO .

Abstract

The standard IEC61850 is the first standard that considers all the communication needs within a substation. The main goal of this standard is to support interoperability of IEDs, to allow IEDs from different manufacturers to operate on the same network sharing information and commands.

This thesis discusses challenges for implementing the new communication architecture of IEC61850. The aim of this thesis is to demonstrate how an IEC 61850 source code application running at station level can be configured to acquire data from IED installed at bay level.

Within this scope, a laboratory setup has been developed for experimental practice of IEC 61850 using a communication stack depends on open source library (PIS 10). An Ethernet communication based on server /client architecture was used to exchange monitoring and control information among developed IEC61850 server and a commercial client.

Combined with the implementation, this thesis giving a realization method for protocol conversion (or mapping) using a programming techniques to integrate between Modbus master and IEC61850 server.

As a result of that a traffic generation systems for IEC 61850 MMS message has been achieved which will help in evaluation of the data acquisition, data integration and data distribution between the various devices use this standard in Smart Grid systems.

المستخلص

البروتوكول IEC61850 هو أول بروتوكول يشمل جميع أنواع الاتصالات المطلوبة بين الأجهزة التى تعمل فى محطات نقل وتوزيع الكهرباء , حيث يسمح لها بتبادل المعلومات فى ما بينها حتى إذا كانت مختلفة التصنيع . هذا البحث يهدف إلى تطبيق هذا البرتوكول عمليا, بحيث يصف كيفيه تبادل البيانات بين جهازين احدهما (خادم) يقوم بطلب القراءات وإرسال أوامر التشغيل الخاصة بالمفاتيح والأخر (زبون) يقوم بتنفيذ تلك الأوامر . تم كتابه الكود الخاص بالجهازين عن طريق استخدام مكتبه بيانات خاصة بالبرتوكول. بالإضافة الى تطبيق البروتوكول .هذا البحث يقدم طريقه لتحويل البيانات المرسلة بواسطة البرتوكول (Modbus) الى بيانات تستخدم بنيه البرتوكول IEC61850 . ونتيجة لذلك تم تبادل البيانات في صورة MMS messages بين الجهازين حسب البروتوكول , وتم تحليلها للتأكد من تطبيق البرتوكول بصوره صحيحة .

Table of Contents

LIST of FIGURES

LIST OF TABLES

GLOSSARY

Terms	Definition
IEC61850	It’s a communication standard used for the realization of automation in the substation. It is a part of the International Electro-Technical Commission’s (IEC) Technical Committee 57(TC57).
Abstract Communication Service Interface (ACSI)	Virtual interface to an IED providing abstract communication services
Interchangeability	Ability to replace a device supplied by one manufacturer with a device supplied by another.
Interoperability	Ability of two or more IEDs from the same vendor, or from different vendors, to exchange information and use that information for correct execution of specified functions
TCP/IP	Transmission Control Protocol and the Internet Protocol is the computer networking model and set of communications protocols used on the Internet and similar computer networks.
UCA	Utility Communications Architecture
UDP	User Datagram Protocol
GOOSE	Generic Object Oriented Substation Events (GOOSE) where any format of data, such as status, value, etc. is grouped into a dataset and transmitted within a time period of a few milliseconds.
IED	Intelligent Electronic Device describes microprocessor-based controllers used in power system equipment
HMI	Human machine interface

SAS	Substation Automation System, Described as a system which makes it possible for power utilities to monitor, coordinate and operate distribution equipment remotely, usually from a control center.
S/S	A substation is a part of the electrical network system.
MODBUS	Modbus is a serial communications protocol
DNP3	Distributed Network Protocol is a set of communications protocols
SCADA	Supervisory Control and Data Acquisition is described as a system that operates with coded signals that control equipment remotely over a communication medium
SCL	Substation Configuration description Language is the language defined by the IEC 61850 standard used for the configuration of substation devices **ICD-**IED Capability Description **SSD-**System Specification Description **SCD-**Distribution Substation Configuration Description **CID-**Configured IED Description Power System
RTU	Remote Terminal Unit is a device that interfaces electrical equipment objects to a SCADA (system control and data acquisition) system. It is used to transmit data to a master system and control the connected electrical equipment remotely.
CT	Current Transformer is a type of instrument transformer that delivers a proportional reduced secondary current which can be used by protective devices, measuring devices and recording instruments.

VT	Voltage transformer is a type of instrument transformer, which produces a proportional reduced secondary voltage that can be used by metering and protection devices.
MU	Merging Unit is a device used to publish Sampled Values for currents and or voltages.
Logical Node (LN)	Smallest part of a function that exchanges data. A LN is an object defined by its data and methods.
Data object	Parts of a logical node object representing specific information, for example, status or measurement
IEEE	Institute of Electrical and Electronic Engineers

Chapter One

Introduction

Chapter One

Introduction

1. Preface

Substation Automation (SA) is a system to enable an electric utility to remotely monitor, control and coordinate the distribution components installed in the substation. High speed microprocessor based Remote Terminals Units (RTUs) or Intelligent Electronic Devices (IEDs) are used for substation automation and protection. These IEDs are installed in strategic locations for collection of system data and automatic protection of substation equipments [1].

A Communication plays a vital role in automation & control in a substation. We need to collect information from the utility and exchange that information with the Master station in order to execute the control & monitoring actions [2].

Data communication between the control centre and substation devices (IEDs) in remote locations and among the IEDs becomes an important issue to realize the substation automation functions [1].

Various protocols are used for tele-control purpose for example Modbus, DNP 3.0, and IEC60870, but none of them is fully support the interoperability among IEDs supplied by different vendors in the substation.[1]

The communication protocol allows two devices to communicate with each other which use the same protocol otherwise error will occur. So the utility is always restricted to one vendor. With the arrival of open systems concept, it is desired that devices from one vendor be able to communicate with those of other vendors i.e. devices should interoperate. Using open protocol gives the following advantages [3]:

- Open system connectivity
- Supplier independence

1.2 IEC61850

IEC61850 is the international standard for substation automation systems (SAS). It is designed to support the communication of all functions being performed in the substation. It is covering design aspects, defines guidelines for protection, monitoring, control and automation. Also raises interoperability and free allocation of functions to devices (IEDs), therefore supports any kind system philosophy covering different approaches in function integration, function distribution, and SA architecture [4].

IEC 61850 has been a very popular standard for the last decade and it has been developing to include more functionalities. A detailed discussion about IEC 61850 is provided in chapter three.

1.2.1 The advantage of IEC 61850[4]:

- Apply the request an open protocol for protection, monitoring, control at least inside substations.
- This is higher flexibility solution, have the possibility to make extension without being dependant on the manufacturer having delivered the previous parts of the substation equipment.
- Full application and all requests for substation automation, and data interoperability within the substations.

To obtain a complete advantage of this standard, it is important to consider all the major issues related to practical implementation.

1.3 Research problem

How to use IEC 61850 standards in practice to exchange data between two IEDs (server and client) based on communication structure of the substation automation system.

This problem can be sub-divided into the following challenges:

1) There are no details for understanding and implementing of the IEC 61850 standard in academic institutions. Knowledge of implementation methods usually reside by vendors and are generally not available in the public domain.

2) Difficulty to find devices that support IEC61850 capabilities on the local market.

1.4 Research Objectives

IEC 61850 has become a very important topic for researchers as the power system automation needs are rapidly increasing. Currently the situation is that detailed knowledge of implementation of the standard is not available in the public domain as industrial vendors ensure that their methods and implementation techniques remain confidential for economic leverage and also as justification for the high costs of the field devices. So this thesis aims to develop methods, software coding, hard interfacing, protocol conversion techniques and testing methodology that will allow low cost implementing of IEC61850. This thesis will describe how to implement the protocol IEC 61850 to provide a basic acknowledge of engineering applications at substation.

The overall goal of this research is:

- To implement IEC61850 standard. In other words, it to shows how the new standard IEC 61850 can be used by build a (server), and try an example of data monitoring and control to communicate with a client as a simulation of a substation automation system (SAS) with different topologies .
- To provide insight and overview of how the standard is structured and how it works.
- To be familiar with IEC 61850 standard protocol.

1.5 Methodology

The purpose of this research is to achieve the special communication Requirements of the IEC 61850 standard (Client/server communication model).

Software implementation of the IEC 61850 was carried out by using open source library (PIS -010) to build a server by developing of specific Application Programmed Interface (API), and mapping the 61850 objects to User Application objects using the Data Attribute ID. This server will communicate with commercial simulation software (IED SCOUT) as a client running in different PC.

The specific task to achieve a successful completion of this research is to use Packet analyzer software to show the data exchange and have a confirmation for all operation.

1.6 THESIS OUTLINE

The first chapter will describe introduction, Thesis Objectives and scope of work. Chapter Two literature reviews, chapter three introduce the substation and automation system principles. Chapter four will represent IEC 61850 structure and component. Chapter five will focus on implementing IEC 61850 based on substation automation application, with hardware and software description. Finally, Chapter sex will be allocated for results and discussion, chapter seven for conclusion and recommendation.

Chapter Two

Literature Review

Chapter Two

Literature Review

2.1Preface

The work presented in this research is targeted at the first phase on implementation of the new standard IEC 61850 .During initial phase of any product; the emphasis is on the development of standards that support new technologies while leaving the scope of implementations as open as possible, to allow the engineers to freely create and invent tools that can quickly help them to simulate or emulate their ideas.

2.2 Literature search

In order to gain this knowledge a literature search on the trends and methodologies of IEC61850 implementation based on substation automation environment has been done, the following are some of key papers discussing that:

For educational purpose and to raise awareness of the benefits of the IEC 61850 (Bai-gent et al., 2004)[5] teaches about the standardization data names, creation of services and the implementation of IEC 61850 to provide an overview of IEC 61850 protocol and its device model. He explains that IEC 61850 can be easily interfaced with existing legacy protocols OPC.

Also (Apostolov, 2006) [6] describes the different communications related to imple-mentation of the new IEC 61850 standard. He shows that different applications and re-quirements can be met by proper selection of the communications type in IEC 61850. The communications in the substation automation system can be between devices at the same level of the functional hierarchy or between bay and process levels of the system.

The work of (Kostic and Frei, 2007) [7] debates the use of the Abstract Communication Services Interface (ACSI) as an Application Programming Interface (API). Their meth-od was to build Client/Server application using an Application Programmed Interface based on the ACSI core. They use one PC and two IEDs to simulate this. The software of API is written in C++, It has been tested using iec61850 browser, which allow one to connect to any number of devices running an MMS server through core ASCI API.

The use of ACSI as a single API for the developer could accelerate the development work. That led to development of a library to generate and parse IEC61850 messages and using it in implementing a control / protection scheme. The use of the specific IEC 61850 libraries in MMS traffic simulation reduces the user's development time. This thesis based on the same idea of the above related work.

At 2008 Liang and Campbell [8] concentrated on implementing and simulating ACSI in J-Sim open-source simulator. Their ultimate goal was to explore security aspects of the standard and in particular of IEC 61850-7.

Kanabar et al.[9] presented results of modeling communication networks for distributed automation systems with DERs in the OPNET Modeler. In their investigation, SV protocol was mapped directly onto Ethernet layer, and included VLAN tags. However, no information was provided whether SV-specific header, ASDUs and APDU were also appended. Similarly,Ali and Thomas [10] did not specify whether any SV-specific fields were attached to the packet while modeling various communication scenarios within a substation in the OPNET Modeler.

(Eunkyu In ,2011)[11] Deals with implementation of a MMS communication stack library and the development of an IEC61850 one-chip communication stack processor of an IED. The single chip solution is composed of two processors with an efficient VHDL architecture for the development of IEC 61850 compliant devices.

The hardware was PC platform, Uses4independent processor modules for IEC 61850 communication, microprocessor for logic process control, processor for CT/VT measurement and Protection algorithm. Software was written by C++ Supports only MMS (no support for GOOSE). A prototype IED board has been built that makes use of the one-chip solution for logic control and IEC 61850 communications.

A method composed of two processors embedded into one single chip is described in order to increase reliability and accuracy of signals between the communication processor and the logic processor. The one-chip solution ensures low cost, low power consumption and helps reduce the size of IEDs.

(Shujian et al., 2011)[12] Emphasis is on the development of an Intelligent Terminal Unit. His method was using of ARM+ DSP dual-CPU structure. The hardware was ARM7 microprocessor and CPLD and Software written by C++.Requirement details for the design of an intelligent electronic device are given. Structure and function of different hardware modules are also explained. Hardware- software modularization is suitable for IEC 61850 embedded development and enhancement.

(Park et al., 2012)[13] Describes the development of an MMS communication stack library using Object Oriented Programming techniques. Hardware was PC Platform and Software was Microsoft Windows OS and C++ language. Paper shows the development of an MMS communication stack library based on IEC 61850 using C++ Programming language. Using the proposed method, developers can understand and deal with applications more easily and have the advantage of maintenance, and module reuse.

(Yang et al.,2011) [14]aims to design and simulate IEC 61850 communication network, by using a PC platform running NS2 and Wireshark .A topology that satisfies the timing requirement of IEC 61850 proposed . Design and simulate the station bus and process bus was achieved.

Finally In December 2013**,** Mini S. Thomas, Ikbal Ali, Nitin Gupta[15] wrote a paper , the aim of this paper is to demonstrate how an IEC 61850 source code applications running at station level can be configured to acquire data from the IEDs installed at bay level. The paper also describes the role of Manufacturing Message Specification (MMS) in establishing a Communication between the client and the server (IED). By using their implemented steps, the practicing engineers can use the source code to fetch data from IEC 61850 IEDs without writing the whole program from scratch that can save lot of time and engineering efforts. That could enhance the knowledge of researchers towards using it for communicating with IEC 61850 based IEDs leading to reliable automated substations.

Several research investigations looked into the topic of simulation modeling of communication services defined in the IEC 61850 standard. Sidhu and Yin produced two publications [16], [17] showing how the OPNET Modeler network simulator can be used to model the substation communication network and analyze the network's dynamic performance (i.e., packet delay characteristics of time-critical services). Their model of SV traffic generator mapped packets directly onto Ethernet and, in their latter publication, included handling of IEEE 802.1Q VLAN tags. Moreover, in [17], the model additionally appended SV-specific header, application service data units (ASDUs), and application protocol data unit (APDU) according to IEC 61850-9-1.

(Konka et al., 2012)[18] describe an accurate and realistic model of SV traffic generator by using PC platform , NS3 and Wireshark .Packet traces generated using the model was shown to be realistic as they included the full byte structure of Sampled Values

APDU and ASDU. The advantage of cheap testing tool for emulation of sampled value messages was shown.

In [19] the authors think that Smart substations need to solve the problem of protocol conversion between the conventional monitoring equipment and the standardized monitoring system firstly. This paper proposed a realization method of conversion method between Modbus and IEC61850. Object oriented technology is used for information model on Modbus. A protocol conversion method is given by QT technology and MMS-EASE LITE tools package. This is the only paper discuss the protocol conversion between IEC61850 and Modbus. This thesis describes a different method for these protocols mapping.

2.3 Summaries

Therefore, to the best of our knowledge, all previous efforts have investigations into implementation of communication services defined in IEC 61850 standard. The standard documents allowed developers a lot of freedom to actual implementation, this present a serious challenge for them, so some possible solutions to several major implementation issues are suggested.

This thesis discusses challenges of implementing IEC61850 communication architecture and IEC61850 protocol mapping. The intent is to provide the foundation and framework for a real-time implementation.

Chapter Three

Background

Chapter Three

Background

3.1 **Preface**

The stated scope of IEC 61850 was communications within the substation so we need background acknowledge about substation and communication function in the substation.

3.2 **Substation**

A substation is a node in a power transmission and distribution system. The purpose of the substation is to convert the electricity into a suitable form before leaving the station. A common conversion is to increase or decrease the voltage level for it to fit the specifications of a topologically lower level of distribution. The substation also serves as a monitoring unit and regulates the electricity by switching it (connecting and disconnecting lines). The protective measures exist to prevent damage to the station equipment as well as the entire grid connected. In this research a typical substation will be considered. [20].

Fig(3.1) AIS Substation

3.3 Substation Equipments

In order for the substation to fulfill its tasks, it uses some common equipment. These consist of controlling, measuring and protecting units. This section will cover the most basic primary equipment found in substations to allow the reader to interpret the stations described in this report [20]:

3.3.1 Circuit Breakers

A circuit breaker, CB, is a mechanical device that closes and interrupts an electric circuit between contacts under specified faults or abnormal load conditions [20].

3.3.2 Disconnectors

A mechanical switching device, which provides an isolating distance when open. Used to facilitate maintenance and other operations and provides a visual assurance of disconnection .The disconnector allows opening or closing of a circuit if only an insignificant current is switched or if there is no change in voltage between the poles[20].

3.3.3 Earthing switches

Mechanical device used for earthing/grounding and short-circuiting de-energized substation components. These switches are designed to withstand currents over a specified time under abnormal conditions, but not to carry continuous currents [20].

3.3.4 Power transformers

The transformer is the main part of the substation, its main task is to transfer electric energy between two different voltage levels (e.g. to a distribution grid or a sub transmission grid). Higher voltages are used to transmit energy from power plant to consumption areas, and lower voltages are used in distributing the energy [20].

3.3.5 Voltage transformers

VTs are transformers flow output that is proportional to and in phase with the primary voltage. It is used in measurement and protection [20].

3.3.6 Current transformer

A transformer has its primary winding in the high voltage line and its secondary winding outside. The task is to provide a current proportional to that of the one in the primary winding. Current transformers are used in measurement and protective purposes due to the fact that the measurement equipment is isolated from the higher voltages [20].

3.3.7 Surge arrester

In the protection against harmful transient voltage overloads a surge arrester is the basic protective device. When a transient overload appears at the arrester it becomes conducting and grounds the overload. After the over voltage is reduced the arrester stops conducting and the circuit is returned back to normal [20].

3.4. Notation

All equipment described in the section above has specially designated symbols assigned when studying plans and schematics. Variations occur regarding symbols, but these portray the concept of the symbols used in this report. Below a table of these notations is given[20]:

Table 3: Schematics notation

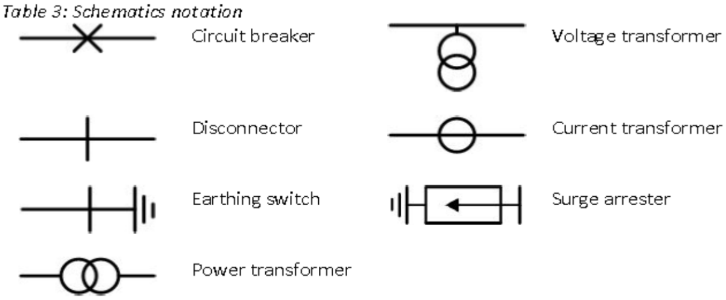


Fig (3.2) Substation Equipments Notation

3.4.1 The Single Line Diagram (SLD)

As an example, part of a single line diagram (SLD) of a substation with two transformer bays is shown in Figure3.3. The SLD shows all power equipment to be controlled and protected [20].

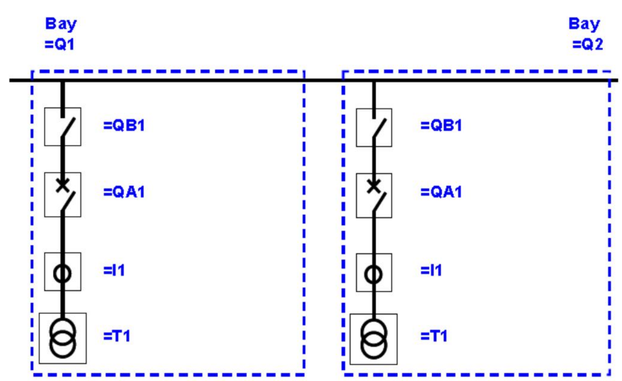


Fig (3.3) Part of substation single line diagram

3.5 Substation functions

A part from the purely electrical functions of a substation, there are also a large number of other operative functions such as monitoring and supervision as well as automation/control and protection .Knowledge of these functions are important when implementing the communication protocols [21].

3.5.1 Monitoring and supervision

In order to be able to supervise the substation, functions need to be implemented that allow acquisition and processing of data from the station. This enables monitoring of the process, either locally or remotely. The main purpose of these functions is to show the state of the process, inform about possible hazards and to archive data. Some of these functions are [21]:

- Event and alarm management
- Disturbance recorder/fault data retrieval

3.5.2 Automation/control

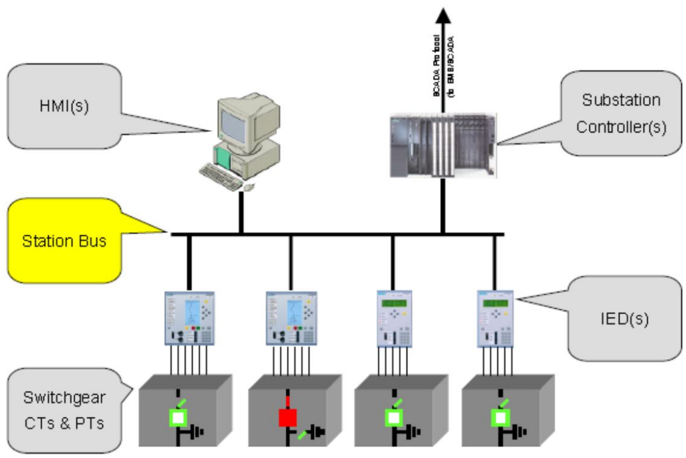
In managing the station, control functions are implemented. These are either used for directing the power flow during normal operation, or for maintenance of primary equipment (e.g. control switchgear, tap changers) [21].

3.5.3 Protection

A typical substation contains a variety of different functions for protection and safety related functions of the substation. These protection functions can be classified by three distinctions (In this research just Monitoring and supervision function will be considered):

- Protection: An active safety level that continuously supervises a give process for harmful states and clearing these by tripping the right CBs.
- Interlocking: A more passive state of protection that supervises the commands issued to a process and blocking harmful commands.
- Automatics: Sequences of actions performed automatically after a given trigger action. Automatics can be triggered by an operator or by another automatic process [21].

3.6 Substation Automation System Architecture (SAS)



Fig(3.4) Substation Automation System Architecture (SAS)

The figure above represents a “typical” SA system:

- switchgear and associated CTs and PTs
- Intelligent Electronic Devices (IEDs)
- Substation Human Machine Interface (HMI)
- Substation Controller.

The last two are optional and either, one or both may be present. The Substation Controller may be (RTU, PLC, Data Concentrator, or a hybrid) [21].

3.7 Telecommunication Protocols

Communications protocol is a list of rules or methods for exchanging messages and data between two different systems or two different networks; it also represents exchanging messages and communicates between the devices within the same system or the same network.

There are many standards and protocols such as Profibus, MODBUS, and distribution network protocol (DNP3), etc., available for the substation communication. Next three parts explain briefly the protocols that IEC 61850 is mapped over them [3].

3.7.1 **Profibus**

In automation technology, Profibus is a standard for field-bus communication developed by German companies in 1987 and some of its key features are listed below [3]:

- Vendor-independent and open field bus standard for automation technology
- support for high-speed time critical and complex communication applications

3.7.2MODBUS

MODBUS is a serial communication standard published by Modicon in 1979 and it is mainly used for the communication between RTUs and control centre devices. It enables millions of automation devices to communicate based on request/reply (client/server) protocol (layer 7 protocol) . There are number of variants of Modbus protocols, such as Modbus remote terminal unit (RTU), Modbus ASCII, Modbus TCP/IP [23].

3.7.2.1 Serial Transmission Modes of MODBUS networks

The transmission mode defines the bit contents of the message bytes transmitted along the network, and how the message information is to be packed into the message stream and decoded. Standard MODBUS networks employ one of two types of transmission modes [23]:

- ASCII Mode
- RTU Mode.

The mode of transmission is usually selected along with other serial port communication parameters (baud rate, parity, etc.) as part of the device configuration [23].

3.7.2.2 RTU (Remote Terminal Unit) Transmission Mode

In RTU (Remote Terminal Unit) Mode, each 8-bit message byte contains two 4-bit Hexadecimal characters and the message is transmitted in a continuous stream. Thus, every 8 bits of an RTU message is effectively 11 bits when accounting for the start, stop, and parity bits of the data frame. RTU mode messages start with a silent interval of at least 3.5 character times implemented as a multiple of character times at the baud rate being used on the network. The first field transmitted is the device address. The allowable characters transmitted for all fields are hexadecimal values 0-9, A-F[23].

3.7.2.3MODBUS Functions

The function code field of the message frame will contain 8 binary bits (in RTU Mode) that tell the slave what kind of action to take. Valid function codes are from 1-255, but not all codes will apply to a module and some codes are reserved for future use[23].

3.7.2.4MODBUS Data Field

The data field provides the slave with any additional information required by the slave to complete the action specified by the function code. The data is formed from a multiple of two hex digits in RTU mode, in range 00H-FFH[23].

The data field typically includes register addresses; count values, and written data. If no error occurs, the data field of a response from a slave will return the requested data. If an error occurs, the data field returns an exception code that the master's application software can use to determine the next action to take [23].

3.7.2.5 Parity and Redundancy Checking

A MODBUS device can be configured for even or odd parity, or for no parity checking. This determines how the parity bit of the character's data frame is set. If even or odd parity checking is selected, the number of 1 bit in the data portion of each character frame is counted. Each character in RTU mode contains 8 bits. The parity bit will then be set to a 0 or a 1, to result in an even (even parity), or odd (odd parity) total number of 1 bits [23].

Also RTU Mode message frames include an error checking method that is based on a Cyclical Redundancy Check (CRC). The error-checking field of a message frame contains a 16-bit value (two 8-bit bytes) that contains the result of a (CRC) calculation [23].

3.7.3 DNP3

DNP3 is a set of protocols (layer 2 protocol) used between different types of data acquisition control equipments. Its primary uses are in control centers, RTUs and IEDs and include the following features [3]:

• Robust, efficient, compatible and secure protocol

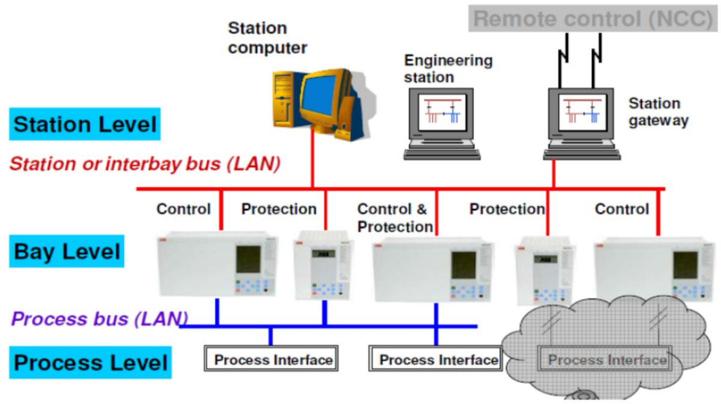
• Time synchronization with RTU

• Event oriented data reporting

3.8 Communication in SCADA and SA

Substation Automation is a method of controlling power system automatically via IEDs (Intelligent Electronic Devices) using control commands from remote users. The communication in SCADA (Supervisory Control and Data Acquisition) and substation automation is more and more often TCP/IP based LAN communication.

The typical architecture of a modern SA or SCADA consists of three levels; station level where is the database, which is represented as the station computer in Figure (2.5) as well as the engineering station, which represent HMI (Human Machine Interface) and the station gateway. Bay level represents the IEDs with the LAN connection between them. Finally, the process level where the event messages are captured and controlled by the station level .The communication between IEDs is horizontal whereas it is vertical communication between two different levels (e.g. station level and bay level)[24].



Fig(3.5) SCADA and SA[24].

From the previous figures, it's clear that SCADA needs a protocol that can achieve both vertical and horizontal communication.

IEC 61850 supports vertical and horizontal communication services beside that it is used for time synchronization and file transfer. IEDs at the bay level communicate together horizontally by sending and receiving GOOSE messages whereas the communication between station level and bay level is done by sending data and receiving SCL files at the station level. All previous services refer to the part 7-2 Abstract Communication Services (ACSI). GOOSE, SCL, ACSI and IEC parts will be explained later [24].

3.9 Manufacturing Message Specification (MMS)

MMS is an application layer used for exchanging real-time data and supervisory control information. The basic component of MMS is defined by VMD (Virtual Manufacturing Device) model, which is represented in Figure (3.6)[25].

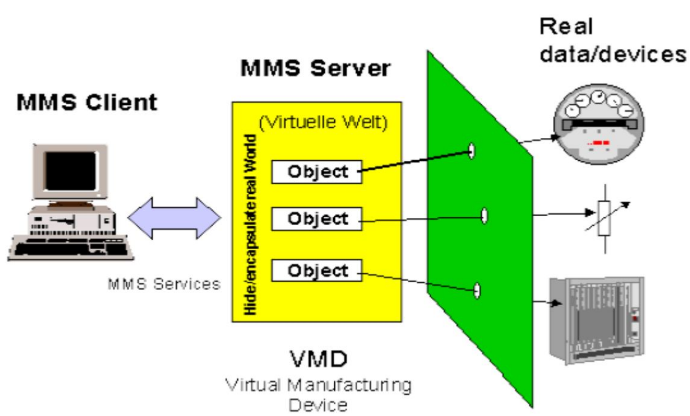


Fig (3.6) Virtual Manufacturing Device (VMD) Architecture[25].

The basic components show the behavior of transferring data between MMS servers and an external MMS client.

UCA and IEC working groups have adopted MMS application layer middleware for the fact of the high technical advantages that MMS provides. The two most important advantages are interoperability, which means the ability of the network layer applications to exchange the data among themselves, generating a communication environment is not needed. The other important advantage is the independency. It makes the interoperability independent of the developer application, connectivity and type of function being performed. Main criticism of MMS is complexity, poor performance and ISO protocol stacks high cost [25].

However, MMS protocol stack is required when using IEC 61850 because GOOSE and SCL files mapping requirements need such a protocol, this comes from the fact that one of the main aims of using IEC 61850 is the virtualization, which makes VMD model in MMS components highly valuable[25].

3.9.1 MMS protocol stack

The MMS protocol stack for the MMS client and MMS server model are based the modules that shown in Figure 3.7 [26].

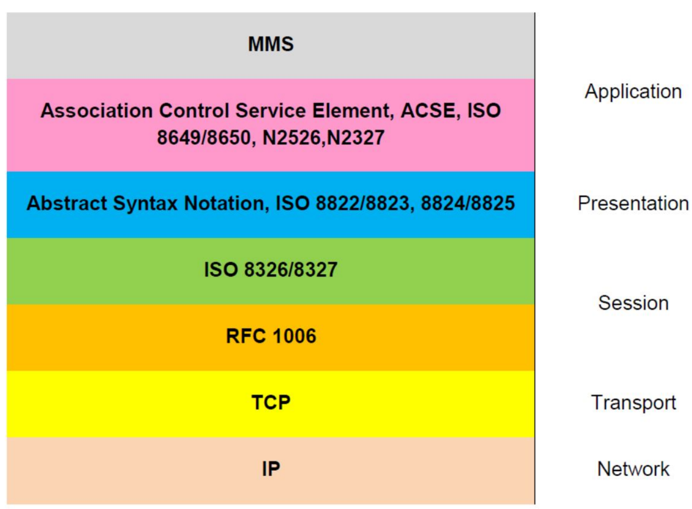


Fig (3.7) Layer 3-7 of the current MMS communication stack[26].

MMS services are placed on top of the stack in the Application layer. Through the services, the MMS client can interact with the MMS server for specific functions such as reading or writing local variables. Association Control Service Element (ACSE) protocol is also situated in the Application layer and is used to establish associate and release Application Associations (AA) between the two communication parties by means of the A-ASSOCIATE and A-RELEASE services and to determine the identity and application context of that association[26].

The presentation layer exists to ensure that the information content of presentation data values is preserved during transfer and to add structure to the units of data that are exchanged. Presentation: ASN.1 and BER. MMS uses ASN.1 as abstract syntax notation at the presentation layer. An abstract syntax notation is the notation used in defining data structures or set of values for messages and applications. The abstract syntax notation is then encoded with a set of encoding rules before transmission [27].

```
Field size:  <      8 bits    ><     8 bits     ><              16 bits              >

             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Field name:  |      vrsn      |    reserved    |          packet length         |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig (3.8) TPTK header format, copied from [27]

MMS is an ISO protocol which requires the transport protocol exchanges information between peers to be in discrete units of information called transport protocol data units (TPDUs). Therefore RFC 1006 [18] describes that all TPDUs shall be encapsulated in discrete units called TPKTs. The TPKT layer is used to provide these discrete packets to the OSI Connection Oriented Transport Protocol (COTP) on top of TCP [27].

```
                <           header           ><            body            >

Byte number       1        2           3          4       5             ... end
                +----+-----------+-----------+----------- - - - - -------+
Field name      | LI |  T    CDT |  TPDU-NR  | User Data                 |
                |    | 1111 0000 |  and EOT  |                           |
                +----+-----------+-----------+----------- - - - - -------+
```

Fig (3.9) COTP PDU format [27]

3.10 Abstract Communication Service Interface (ACSI) and Communication Service Mappings (SCSM)

Abstract Communication Service Interface (ACSI) is a subject of IEC 61850 part 7 and its subparts. Abstract means the definition of the data and information to describe what the services provide. The implementation is done through the Specific Communication Service Mappings (SCSM) by mapping to e.g. MMS, TCP/IP and Ethernet [28][1].

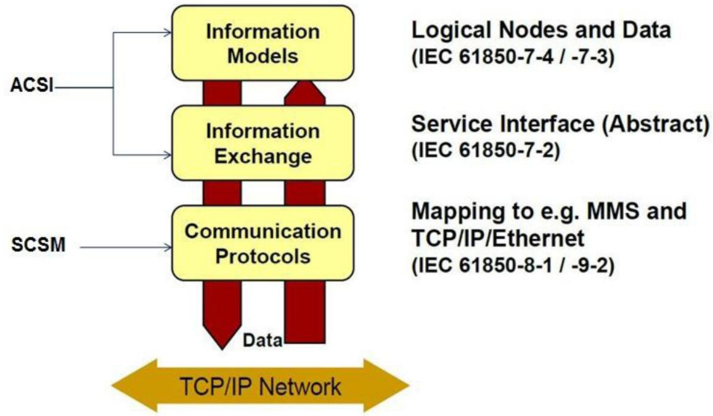


Fig (3.10) ASCI and SCSM.

ACSI represents the application view of IEC 61850, when SCSM which is subject to IEC 61850 parts 8 and 9 represents the communication view [28]. Both Application and communications views of IEC 61850 will be explained in chapter 4.

Chapter Four

Analysis of IEC 61850 Standard

Chapter Four

Analysis of IEC 61850 standard

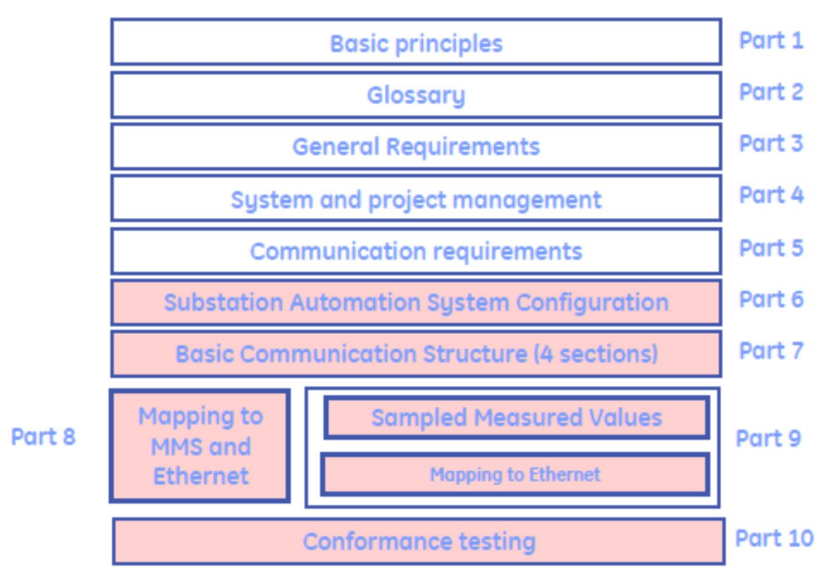
4.1 Background

In 1996, EPRI/IEEE (Electric Power Research Institute/ Institute of Electrical and Electronics Engineers) started a group called Utility Communication Architecture (UCA) This group defined a standard for SA, which known as UCA 2.0. Two years later, IEC TC57 began to work on IEC 61850 to define it as an International standard for SA. This led to a combined effort in 2001 to define an international standard that would combine the work of both groups. Both groups worked on this task and in 2003 the result was the current IEC 61850 specifications [29].

4.2 Overview and Scope of IEC 61850

The IEC 61850 standard series named "Communication networks and systems in substations" comprises in total 10 major sections, which cover different aspects of the substation communication network. The IEC 61850 standard (Edition 1) consists of the following fourteen parts presented in Table 3.1 and published between 2003 and 2005.

Table 4.1 parts of The IEC 61850 standard documents



The parts 1 and 2 are meant to provide an introduction to the standard series IEC 61850 and contain also the glossary of terms used along with definitions regarding the power utility automation systems context.

The general requirements for communication in substations are presented in part 3, with the focus on quality requirements, environmental conditions and auxiliary services. The specific functional communication requirements are described in the 4^{th} and 5th part where all known functions are identified and described in detail.

In part 6, the system configuration language (SCL) used by the IEDs to communicate is explained, allowing devices of different manufactures to exchange information in a compatible way [30].

The most important part of the IEC 61850 standard is 7-xx series named “Basic communication structure for substation and feeder equipment”:

4.2.1 IEC61850-7-1 Principles and models

This part scope is to provide an introduction to the modeling methods, communication principles and information models used in the 7-xx series. Information is given regarding the dependency of this part with the requirements from IEC 61850-5 [30].

4.2.2 IEC61850-7-2 Abstract communication service interface (ACSI)

The abstract services definitions are found in this section together with the methodology of client-server communication. The modeling process of IEDs such as information model and exchange are possible by accessing the pre-defined functions [31].

4.2.3 IEC61850-7-3 Common data classes

The part 7-3 describes the common data classes (CDCs) such as status information, measured and controllable status information, controllable analogue set point information, status and analogue settings and the common attribute types associated with the substation modeling [32].

4.2.4 IEC61850-7-4 Compatible logical node classes and data classes.

The information model and functions of the real devices used in substation applications are specified in this part. The logical nodes (LNs) name and data object necessary for developing communication between IEDs are described in particular depending on their class of origin [33].

In the section 8.1 of the standard, the procedure of mapping the abstract data object and services defined in parts 7-2,7-3 and 7-4 into the MMS is described [35]. In the parts 9-1&9-2 the service communication mapping (SCM) which is necessary for the transmission of sample values is specified, while in section 10 the conformance testing methods of devices and engineering tools are defined [36,37].

4.3 IEC 61850 Basic Approach

The functions performed by Substation Automation (SA) system are in, general, switch control, data monitoring, protection etc. In IEC-61850, these functions are broken into low-level functions called sub-functions. Each sub-function is performed by the IED installed in the substation. Each IED can perform one or many sub functions.

A set of sub-functions is integrated together to realize a substation automation function. These communicate with each other through Local area network in the substation. Specific syntax and semantics are defined for communication between sub-functions. All the possible sub-functions have been standardized in IEC-61850. Information produced and required by each substation is given in the IEC-61850 standard. The sub functions are assigned at three levels as shown in fig (4.1) (i) Process level (ii) Bay level (iii) Station level.[1]

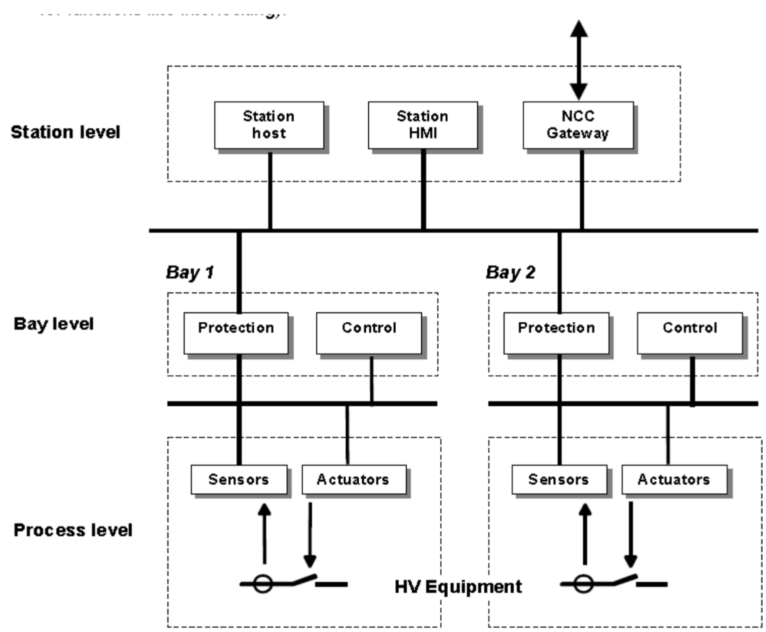


Fig (4.1) IEC61850 levels

4.4 IEC 61850 communication stack

The communication services mapping of IEC 61850 has to meet several communication requirements defined in IEC 61850-5. As a consequence, the different message types which belong to different performance classes are mapped to different communication protocols in order to support the respective requirement for specific message types. These message types will be described in more details in this section [37].

The IEC 61850 communication services mapping is depicted in Figure (4.2). The message types which have similar performance requirements are grouped together and mapped to the same protocol.

For example the type 1 and 1A messages are very time critical so that they are mapped to Generic Object Oriented Substation Events – GOOSE and directly mapped to Ethernet to reduce processing time caused by overhead of transport and network layer protocols. The raw message type 4 is mapped to Sampled Value (SV) which is a protocol designed to carry raw data and is also directly mapped to Ethernet to achieve time-critical performance [37].

Type 6 message represents the message used for time synchronization and is mapped to the Simple Network Time Protocol (SNTP). Message type 2, 3 and 5 which can be used to support core IEC 61850 services are mapped to the Manufacturing Message Specification (MMS) [37].

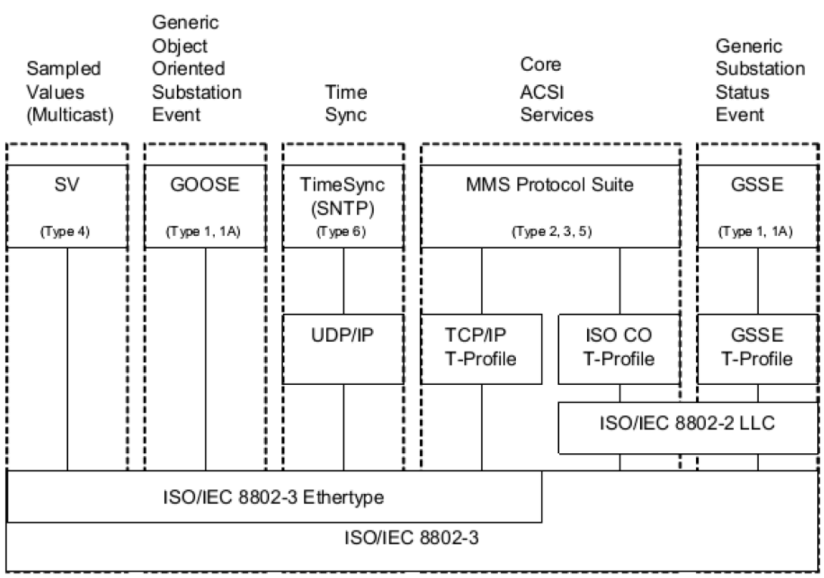


Fig 4.2 –IEC61850 Overview of functionality and profile [37]

Next figure (4.3) describes the mapping of IEC 61850 data models over the previous mentioned communication protocols.

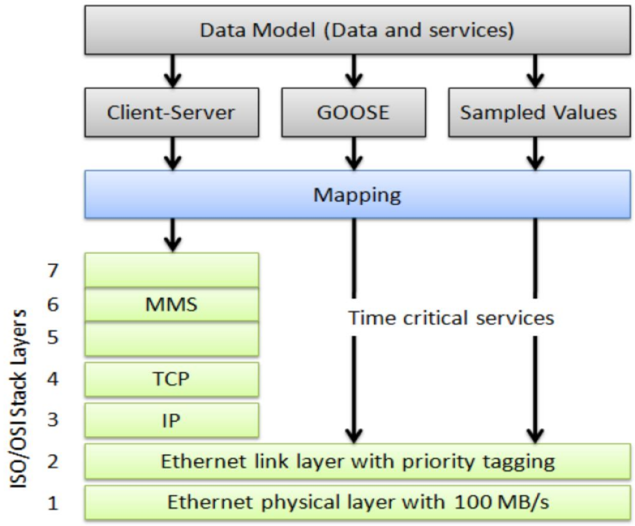


Fig (4.3) Mapping to stack (OSI and IEC 61850)[37]

The IEC 61850 standard was founded to avoid problems between different manufacturers' communication solutions. To reach interoperability, the approach taken in the IEC 61850 standard is to separate the domain related model for both data and communication services from the protocol.

The OSI seven-layer stack is used to code and decode information into bit strings for communications over a serial link. The stack consists of Ethernet, TCP/IP and MMS layers. Only time-critical services are mapped directly to the Ethernet link layer. All other services are mapped in the MMS application layer, as shown in figure (4.3).

IEC 61850 offers three kinds of communication schemes and services. These are:

- Client-Server communication
- GOOSE messages
- Sampled Value

The purpose of this thesis is to implement IEC618500 client – server communication scheme and deals with the generation of MMS messages, other one (Goose message and sampled value are out of scope)

4.5 Virtualization Model

The core of IEC 61850 is the interoperability between IEDs from different vendors .In other word, interoperability between different functions that are performed by different Physical (real) devices. This is done by using data models, data exchange bases on these models [38].

Virtualization means that every physical device can be represented in a Virtual world and only aspects of a real device that are of interest for the information exchange with other devices are virtualized. This method is called distributed functionality and the involved devices in data exchanging are called distributed devices. Achieving the virtualized model when using IEC 61850 comes from the fact of mapping the standard over MMS where VMD model is used figure (4.4) [38].

In IEC 61850 series, one of the core functionalities of the standard is to decompose the real device comprise into smallest entities. These entities are called logical nodes.

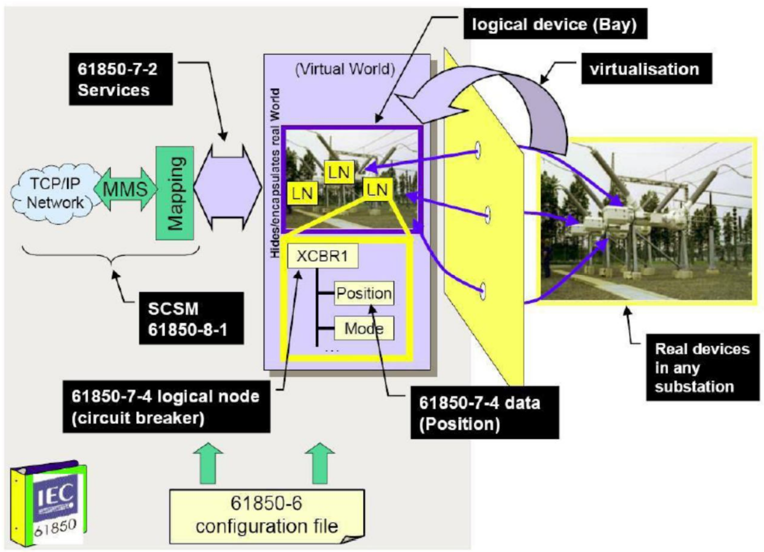


Fig (4.4) Virtual world vs. real world [38]

Similar functionalities performed by different devices build a logical node and several logical nodes performed by different devices or by the same device build a logical device. Logical device is represented in virtualized model it does not usually represent one real device; it mostly represents a different aspects or different logical nodes from different real devices. A logical device is always implemented in one IED even though it is built by logical nodes from different real devices, which means that logical devices are not distributed [38].

4.6 IEC61850 Data Model

The main substation automation (SA) function consists of several sub-functions which are appropriately interfaced .These sub-functions are known as Logical Nodes (LN). Logical nodes reside in the IED which is also called a Logical Device (LD). One logical device (IED1) holds one or multiple Logical Nodes (LN1 and LN2) as shown in fig (4.5) [1].

A logical node is realized through an object class, called logical node class. For example, XCBR is a logical node class for monitoring and operation of circuit breaker. A logical node class consists of a set of data belonging to different class. For example data “Pos” of data class type “DPC” is a member of logical node class “XCBR”. Further each data class inside a logical node class consists of a set of data attributes. For example, data class “DPC” consists of data attributes given by Control value “ctlVal”, Operating time “operTime”, etc.

These data attributes are used as necessary parameters to perform the functions. For example, XCBR1.Pos.stVal represents the close/open status of circuit breaker “XCBR1”. [1]

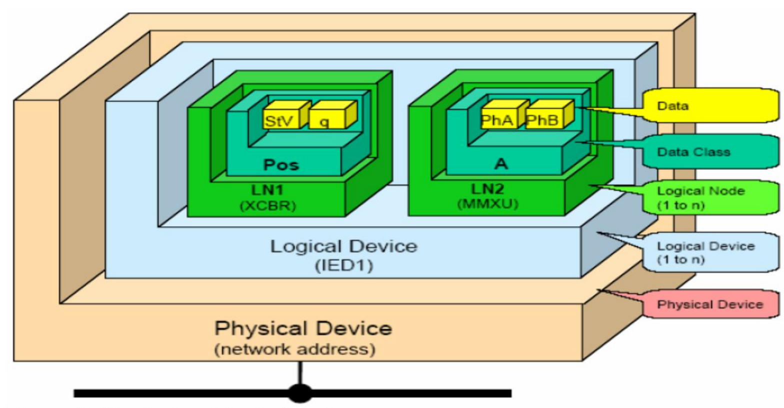


Figure 5: Physical and logical device.

Fig(4.5) Physical and logical devices [1]

Each logical node contains a pre-defined set of data classes. Every data class contains many data attributes (status value, quality etc.). The logical devices, logical nodes and data objects are virtual parameters, they merely seem to exist. The logical nodes data contained in the logical devices are fundamental for the description and information exchange inside the power station automation systems to reach interoperability [1].

4.6.1 Physical Device (IED)

The top tree structure of the data model is represented by the physical device, which is also defined as server. An electrical network can contain one or several IEDs which have the capability to connect to exchange information by using a unique IP Address [30].

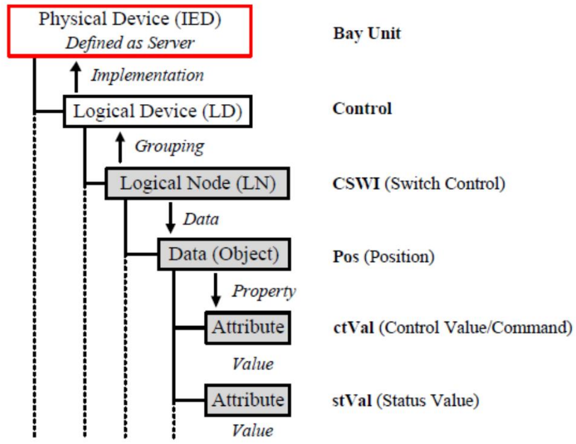


Figure 4.6 Object model of IEC 61850 [11]

4.6.2 Logical device (LD)

The logical device (LD) inside the server is defined as the main entity of the data model. It contains a group of LNs depending on the functions required by the particular device from the network. It is important to know that an IED contains only one LD and it does not include LNs from other devices. Also it is mandatory that each device contains at least three logical nodes as it is shown in Figure3.7 where the relationship between the common LNs is exposed, Logical Node Zero (LN0) and LPHD, which correspond to the logical device and the physical device, alternatively [30].

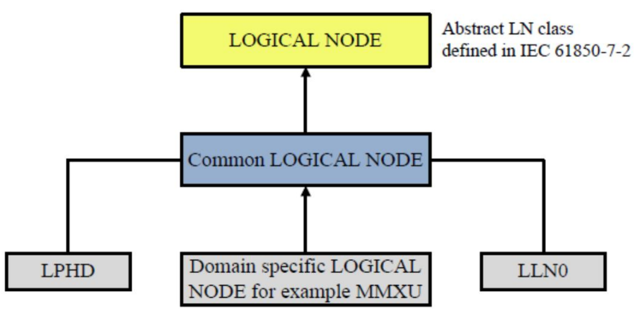


Figure 4.7 Logical node relationships [6]

4.6.3 Logical Node (LN)

The logical node (LN) represents a combination of data and services related to a particular function within an IED. As it was previous mentioned, at least three LNs must be contained in a LD, such as LPHD, LLN0 and a domain specific LN (e.g. MMXU). The LLN0 contains the common information for the logical device as "health, mode, beh and NamePlt” while the LPHD represents the common information related to the physical device [33].

Logical nodes are grouped into 13 main groups; each group contains a specific number LNs. 92 LNs are covering the most common applications of power stations and feeder equipment. The names of logical nodes begin with the character representing the group to which the logical node belongs, depending on their functionality and they are described in detail in part 7-4 [33].

IEC 61850 store for the future clear rules relating to extensions of the information models. Table (4.2) shows the LN groups according to the last updated information models in 2006.

Table 4.2 LN groups**.**

Group Indicator	LNs groups	Number
A	Automatic Control	8
C	Supervisory Control	5
G	Generic Function References	3
I	Interfacing and Archiving	4
L	System Logical Nodes	3
M	Metering and Measurement	8
P	Protection Functions	28
R	Protection Related Functions	10
S	Sensors and Monitoring	4
T	Instrument Transformer	2
X	Switchgear	2
Y	Power Transformer	4
Z	Further (power system) Equipment	15
TOTAL		92

In order to provide further information of how a logical node is modeled, we show the logical node of measurements that we will use it later in chapter five. The voltage to ground phase A measurement value from the Metering and Measurement group (LN MMXU) is shown in Table 3.3.

Table 4.3 The logical node MMXU[33]

Object Reference	Type	Remark
MMXU1	LN	Measurement LN
MMXU1.PhV	DATA	Phase to ground voltages
MMXU1.PhV.phsA	DATA	Value of Phase A
MMXU1.PhV.phsA.cVal	DataAttribute	Complex Value
MMXU1.PhV.phsA.cVal.mag	DataAttribute	Magnitude of complex number
MMXU1.PhV.phsA.cVal.mag.f	DataAttribute	Floating point number

4.6.4 Data Objects

The data class represents the information contained by a LN and which is intended to be accessed in the real device. The information can be represented by currents, voltages, power, temperatures, status, quality, timestamps etc. In Figure 4.7 the anatomy of an object name is presented using the example of logical node MMXU. The PD and LD can take any name in the IEC 61850 world, while the other parts as LNs, DOs and DAs are predefined names in the standard [34].

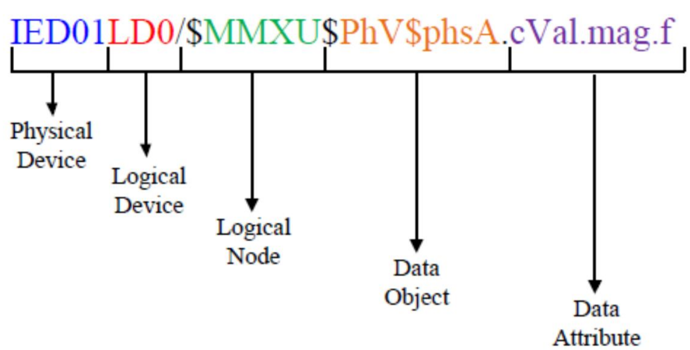
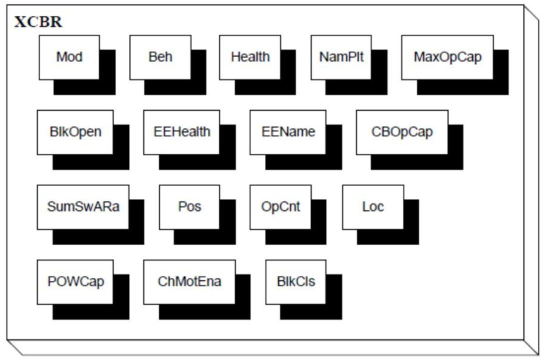


Figure 4.8 object name according to IEC 61850-8-1 [34]

LNs are a kind of “Folders” which contain data that can be used or exchanged. For example, The XCBR LN implements the functionality of CB (Circuit Breaker) by grouping 16 data classes as shown in Figure 4.9. XCBR data set contains characters correspond to the used commands for the CB such as BlkOpn (block open operation), Beh (Behavior), etc [32].



4.9 XCBR (Circuit Breaker) Logical Node [32]

One example is given in Fig (4.10) shows how to use the data object to read the state value of a switch position in a protection relay and how by following the contents of the data object the state value can be reached and be read. A decimal number describes each attribute status, 0 means that the switch is in the intermediate state, 1 shows that it is off, 2 is on and 3 warns that it is in a bad state. Data objects and data sets are defined in IEC 61850-7-3[32] but their analysis during the real-time applications is done in IEC 61850-8-1 [34].

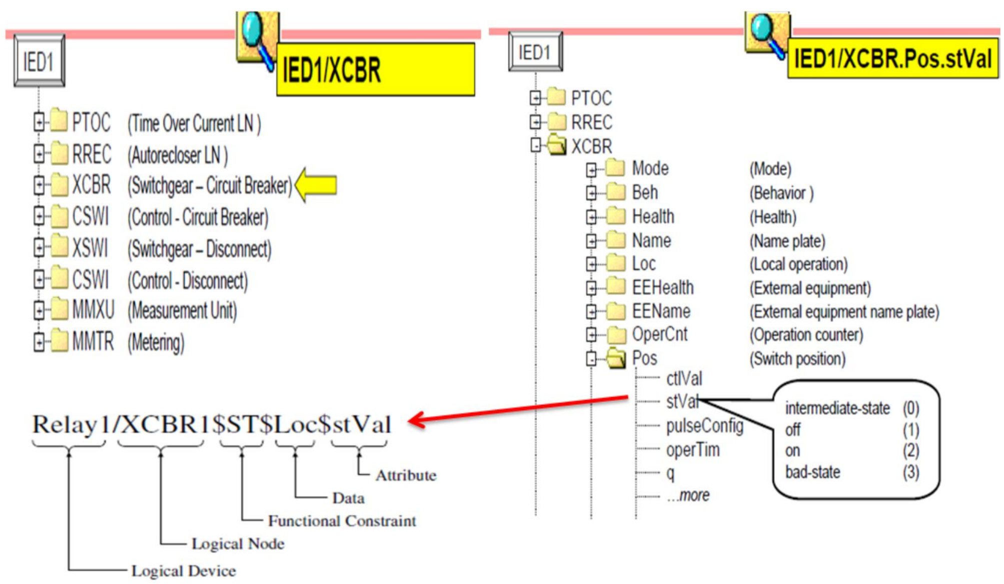


Fig (4.10) Analysis of a data object

4.7 Services model

After the information has been specified by means of LNs, DAs and service parameters, the data has to be transmitted over the communication network from the IEDs towards the control center or between interconnected devices. The information exchange is defined by the services categorized in part 7-2 [31] of the IEC 61850 which are presented in Fig (4.11).

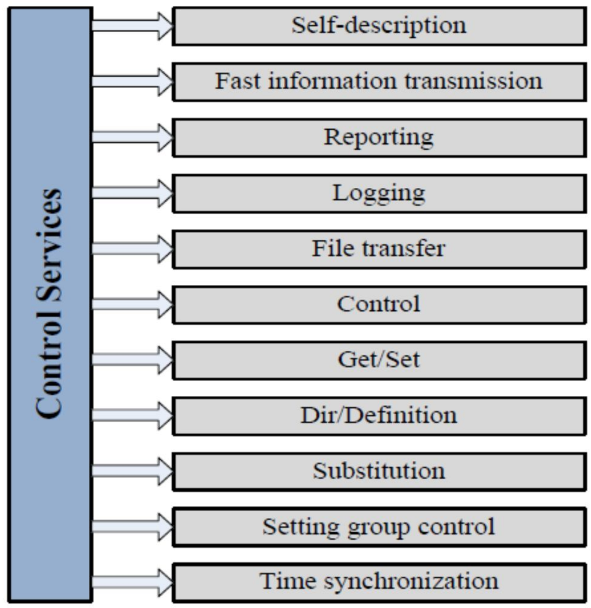


Fig (4.11) IEC61850 Services model [31][39].

4.7.1 ***Self-description*** – the client is using this service to access the hierarchy of the information model and its subclasses, by using the command GetDirectory.

4.7.2 ***Peer-to-peer information*** exchange provided by:

- Generic Object Oriented Substation Events (GOOSE) – transmission of information regarding the status of the devices (e.g. start, stop, trip, etc.)
- Sample Value (SV) – uses the publisher-subscriber mechanism for fast transmission of multicast synchronized sample values (the data is written into the transmission buffer and its being accessed by the receiving side).

4.7.3 ***Reporting*** – it contains two report data classes such as buffered-report-control-block(BRCB) and unbuffered-report-control-block (URCB). Ensures the report of the data object values from the logical node to client within a specific configured time.

4.7.4 ***Logging*** – it is used to store the events occurred and gives the possibility to access them at any time from the internal storage.

4.7.5 ***File transfer*** – for configuration, disturbance recording or historical data

4.7.6 ***Control*** – controls the state of internal and external processes by a client. Applies to CDCs and DAs which have enabled this function.

4.7.7 ***Data-Set*** – Retrieve / Write/ Create/ Delete the data object set values.

4.7.8 ***Dir/Definition*** – to access the directory information and its data definition

4.7.9 ***Substitution*** – allow to overwrite manually the value of a DA by using the substitution associated DAs "subEna" and "subVal". In case of reporting, the new value will be transmitted and not the original one.

4.7.10 ***Setting group control*** –allows a data Instance to take multiple values which are used one at a time.

4.7.11***Time/Time synchronization*** – provides accurate synchronization to applications in the server and client substation IEDs.

The services are described in an abstract way, meaning that only the information that is required to perform a desired action will be outlined for more information see part 7-2 of the IEC 61850 [31].

4.8 IEC61850 Communication Topology

Communication according to IEC 61850 takes place between LNs. In any implementation, physical communication takes place between IEDs. Multiple communication ports may exist. IEC 61850 is based on Ethernet, and Ethernet allows different physical variants. Since the standard and Ethernet is supporting both client-server relations and peer-to-peer communication, any communication topology connecting all related IEDs fulfills the functional requirements [21].

The IEC61850 communication stack supports three modes of communication called Generic Object Oriented Substation Event (GOOSE), Sampled Values, and MMS for exchanging data between a client and server device in substation automation [21].

4.8.1 Client – server communications:-

For this vertical relationship IEC 61850 is using the client – server concept. The server is the process or bay level IED, which provides all data to the client at station or any remote level. Client server communication relies on the full seven layer stack using a confirmed transmission layer and is, as consequence, very reliable but relatively time consuming. Therefore, the client server communication is not suited for time critical data transmission but very well for the communication with an operator [21]

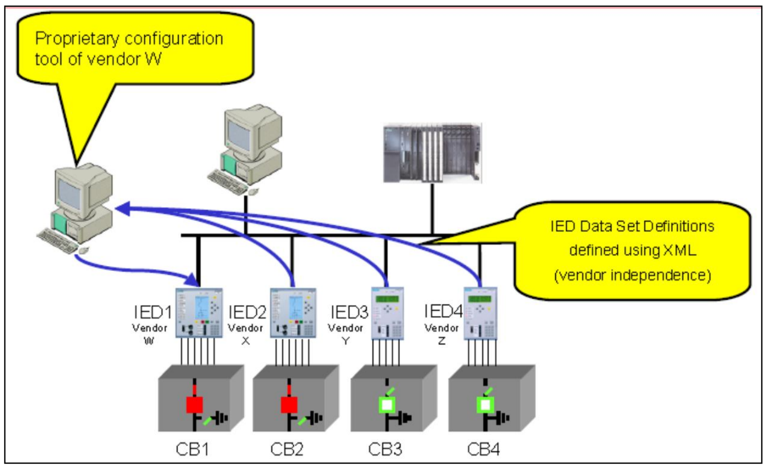


Fig (4.12) Client – server communications [21]

4.8.2 Peer to peer communications:-

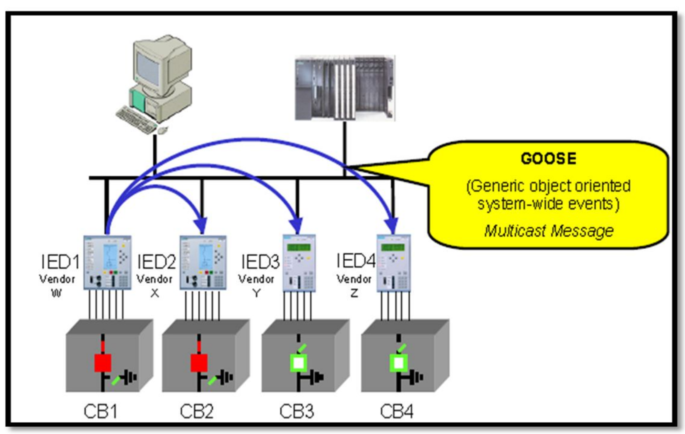


Fig (4.13) Peer to peer communications [21]

IEC61850 introduces a specific information exchange service called GOOSE (Generic Object Oriented Substation Event) based on the peer to peer concept [21].

A GOOSE message is used to exchange data between IED's and is only one part of the new standard IEC61850. GOOSE communication represents the horizontal communication in the substation automation system [21].

The GOOSE message is sent as a multicast message over the communication network. It is used for time critical communication like transmission of a trip signal from the protection relay to the circuit breaker. By broadcasting the same GOOSE message to one or several IEDs at the same time that allows faster communication and lower cost [21].

4.9 GOOSE and Sampled Values

IEC 61850 presents two real-time communication methods that can be used successfully in protection engineering: Generic Substation Event (GSE) and Sampled Values (SV) messaging [30].

GSE messages are divided into two types: Generic Substation Status Event (GSSE) and to Generic Object Oriented Substation Status Event (GOOSE). The main difference between GSSE and GOOSE is the fact that GSSE is an older message type, which only supports data in form of binary-only. GOOSE is more flexible, supporting both analog and binary data. All new substation automation systems use GOOSE only instead of GSSE for horizontal communication. GSSE and GOOSE can both exist in a system, but are not compatible with each other [30].

GOOSE, as mentioned before, is described as rapid horizontal communication between IEDs. GOOSE messages are mapped straight to Ethernet layer (layer 2), thus providing fast transmission of time-critical data. The messages are transmitted over LAN as a multicast see figure (3.14), so the same substation event message is delivered simultaneously to multiple IEDs. The IEDs that are configured to receive the message can subscribe it [30].

However, due to nature of the multicast and the design of the Ethernet, the messages are connectionless. This means that we cannot know which IEDs will receive the message, the message delivery is not ensured, and the acknowledgement of the successful receiving of the message is not sent by the IED. Because of this, IEC 61850 specifies a retransmission scheme, which increases the probability of successful reception in all subscribing IEDs. Furthermore, GOOSE uses periodic heartbeat messages to enable detection of link or device failure [30].

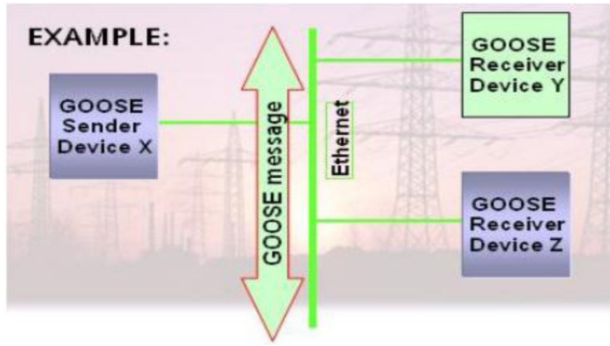


Fig (4.14) Example of GOOSE messages exchange [30].

The digital information exchange between IEDs and next generation voltage and current sensors is becoming possible. IEC 61850 defines Sampled Values (SV) for this purpose. Sampled Values are also mapped to the Ethernet layer (layer 2) being time-critical data. SV messages are used for transferring digitalized measurement values of current and voltage from switchyard to IEDs inside substation. The data collection (from current and voltage sensors) and digitization is made by a Merging Unit (MU), which sample the signals at an appropriate, synchronized rate. Like GOOSE messages, SV messages are also transmitted via LAN as multicast to any number of subscribing IEDs in the Ethernet network [30][35].

4.10 Time Synchronization

Time synchronization is used for synchronizing all devices within the system. The time source is usually external (satellite or radio clock). IEC 61850 presents five different requirement levels of time accuracy for time synchronization, ranging from 1 millisecond to 1 microsecond against real time. It also presents the protocol SNTP (Simple Network Time Protocol) for time synchronization accomplished via LAN communication [30].

SNTP is, as its name states, a simpler modification of NTP (Network Time Protocol). These two protocols differ in the areas of error checking and time correction. In addition, the SNTP uses only one time server at a time, while NTP uses multiple ones. They both provide synchronization over LAN. With SNTP, the system is capable to reach time accuracy of 1 millisecond. However, this is not precise enough for Sampled Values (of voltages and currents) needed for protection, which require an accuracy of 1 microsecond[30]. Therefore, more precise time synchronization methods must be used.

4.11 Substation Configuration Language (SCL)

SCL is introduced in Part 6 of the IEC 61850 standard. The main duty of SCL is to guarantee the interoperability by exchanging information between IEDs from different vendors and the station computer. It comes from the fact that each IED is configured by independent configuration tool developed by its manufacturer. This is done by using four types of SCL common files. These files are the IED Capability Description (ICD), Configured IED Description (CID), Substation Configuration Description (SCD) and System Specification Description (SSD) files [1].

Table(4.4) SCl Files

SCD	Describing the complete substation description including topology, communication, all IEDs and logical nodes.
ICD	Describing IEDs capability, specific IED configuration and functional and engineering capabilities.
SSD	Describing the single line diagram, topology, functions of the substation and (LN's...) without concrete IEDs to run the functionality.
CID	Describing the communication related part of an IED.

All ICD files get imported into the IEC 61850 system configuration, which allows the configuration of GOOSE messages by specifying the senders (publishers) and the receivers (subscribers) of messages. The system configuration tool creates the SCD file, which includes the one line diagram of the station and the description of the GOOSE messages. Each IED Configuration tool must be able to import an SCD file and extract the information needed for the IED, and then join all information in one SCL file to be sent to another IED or to the station computer and the control unit in the SA [1].

4.12 IEC 61850 Benefits

Benefits of IEC 61850 is measured according to the high requirements of IEDs inside the SA, such as high-speed IED to IED communication, guaranteed delivery times, multi-vendor interoperability, etc . IEC was mainly designed to satisfy these requirements with help of GOOSE and SCL files, which adds more advantages when using the IEC standard [41].

The key benefit of IEC 61850 is use of the virtualized model. The virtualized model of logical devices, logical nodes, ACSI, and CDCs enables definition of the data, services, and behavior of devices to be defined besides the protocols that are used to define how the data is transmitted over the network. In addition, every element of IEC 61850 data is named using descriptive strings to describe the data. Devices are self-describing. Client applications that communicate with IEC 61850 devices are able to download the description of all the data supported by the device from the device without any manual configuration of data objects or names [41].

Another key benefit of IEC 61850 is the high-level services offered because of the use of ACSI model, which supports a wide variety of services that exceeds what is available in the typical legacy protocol. GOOSE, GSSE, SMV, and logs are few examples of the unique capabilities of the IEC standard besides more capabilities gained when including SCL. It enables the configuration of a device and its role in the power system to be accurately defined using XML files. The use of SCL also eliminates procurement ambiguity. [41]

The major benefits of using IEC 61850 are [40]:

4.12.1Implementation of New Capabilities: The radical services and unique features of IEC 61850 enable new capabilities that are not viable with most legacy protocols. Wide area protection schemes that would normally be cost prohibitive become much more feasible. Because devices are already connected to the substation LAN, the incremental cost for accessing or sharing more device data becomes insignificant enabling new applications that would be too expensive to yield.

4.12.2Less Connection Cost: IEC 61850 allows devices to exchange data and status using GOOSE and GSSE (Generic Substation Status Event) over the station LAN rapidly without having to wire separate links for each relay. This reduces wiring costs by utilizing the station LAN bandwidth for these signals.

4.12.3Less Transducer Costs: instead of separate transducers for each device needing a particular signal, a single merging unit supporting SMV can deliver these signals to many devices using a single transducer lowering transducer, wiring, calibration, and maintenance costs.

There are several other costs reduced by using IEC 61850 due to the fact that devices don't require as much manual configuration as legacy devices and that reduces the Commissioning cost. Because IEC 61850 defines more of the externally visible aspects of the devices besides just the encoding of data on the wire, the cost for equipment migrations is minimized. Because IEC 61850 devices don't have to be configured to expose data, new extensions are easily added into the substation without having to reconfigure devices to expose data that was previously not accessed.

This results a lowering in the extension costs. In addition, integration costs are reduced by utilizing the same networking technology that is being widely used across the utility enterprise (UCA 2.0). As it was mentioned previously, IEC 61850 is UCA 2.0 plus new added functionalities.

4.12.4 High Performance of GOOSE: The increment of IEC 61850 overall benefits is also related to the use of GOOSE messages, which brings a number of highly performed functions. One of the essential preconditions for using GOOSE is that it performs adequately compared with a hardwired solution. In addition, due to the non-deterministic nature of Ethernet, reliability is guaranteed under difficult communication load conditions.

By using GOOSE messages, a high operational speed (e.g. less than 4ms for protective relaying) can be achieved. However, GOOSE messages can be delayed using certain delay commands when it is needed. Operational speed is 30-50% faster when comparing with the operating speed of classical hardwired. Moreover, Three LAN configurations (10 MB switched hub, 100 MB shared hub, and 100 MB switched hub) are able to deliver 100 messages within these 4ms.

Chapter Five

Methodology

Chapter Five

Methodology

5.1 Introduction

The research work focuses on the experimental development of IEC 61850 client / server architecture, this is achieved by using Application Programming Interface (API) required to create this custom.

Substation physical devices are accessed via Ethernet, from the IEC 61850 viewpoint the actual devices are a collection of logical devices made up of logical nodes, and these logical devices are mapped to the specific communication services GOOSE, Sample Value, SNTP, or (MMS) service over TCP/IP. The use of MMS messages to demonstrate the monitoring & control characteristics using a client-server scheme is the main concern.

5.2 IEC 61850 Implementation Method

As we mention in research methodology we use IEC 61850 Application Programming Interface (API) and a communication stack (PIS-010.lib) that developed by System-Corp.

The protocol integration stack PIS-010 implementation allows users to integrate a Server or Client applications easily. It uses the Substation Configuration Language (SCL) XML file format using “Private Elements” for mapping the 61850 objects to User Application objects using the Data Attribute ID. The call-back functions must be provided to the PIS-010 for converting the mapped objects to 61850 objects. The user application objects are read, written or updated using the call-back functions with Data Attribute (DA) ID [42]. See Figure 5.1 below.

The PIS-010 stack library is ANSI-Standard C compatible and can be configured via SCL file. It is an event driven stack (Interrupt based), so there is no need to poll for data [42].

The function of the IEC61850 API and the PIS-010 stack is to provide any IEC61850 application programmer with features that enable them to create -from a communication point of view- a functional IEC61850 application within a short period of time (demo version).

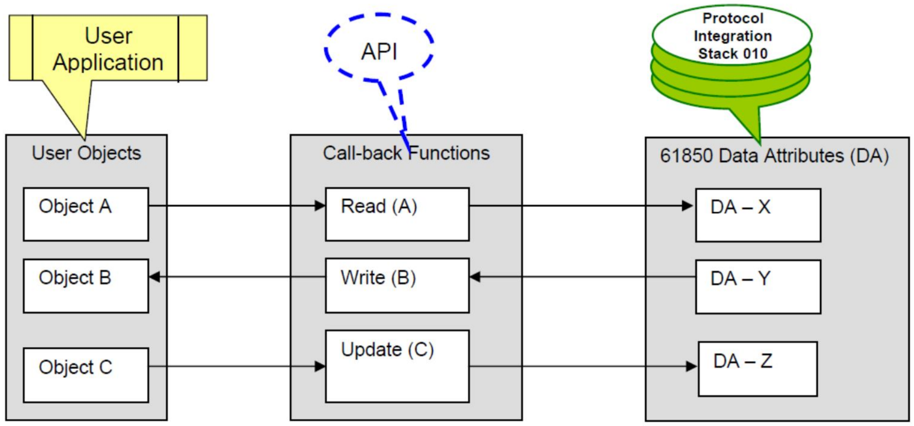


Fig (5.1) Overview of the PIS-010 communication stack and its IEC 61850 API [42]

The API provides the programmer with more flexibility when integration of the system to the main software core specific protocol of communication (GOOSE, SV, MMS, SNTP…) and other specific communication services using the IEC 61850 stack (PIS-010) is considered [42].

Therefore the software development time is reduced and implementation is accelerated since the required application protocol is already provided within the PIS-10 stack. The IEC 61850 protocol Integration stack (PIS-10), implements the ASCI layer and the SCSM on top of the OSI model. Fig (5.2) presents the communication protocol contained in the PIS-010 stack [42].

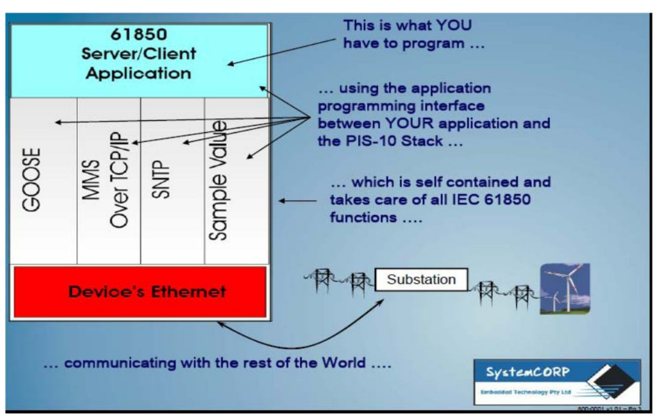


Fig (5.2) PIS-010 stack [17]

5.3 Server /Client architecture

The purpose of a server/client application is to exchange real time information within the electrical network between the interconnected IEDs. The information travels from the server (IED) towards client (Controller) which performs the calculations based on the information received. The data is then transmitted back to the server which execute the desired operations (for e.g., Measurements, control, Start/Stop, Reset etc.).

It should be mentioned that after the server transmits the information, its exchange capabilities remain blocked until the further reply from the client.

In figure (5.3) the configuration of a client-server application is illustrated, by using the Application Programming Interface (API) offered by SystemCorp . The TCP/IP protocol together with the Data Report Service is used to broadcast the information between client and server within the power utility network.

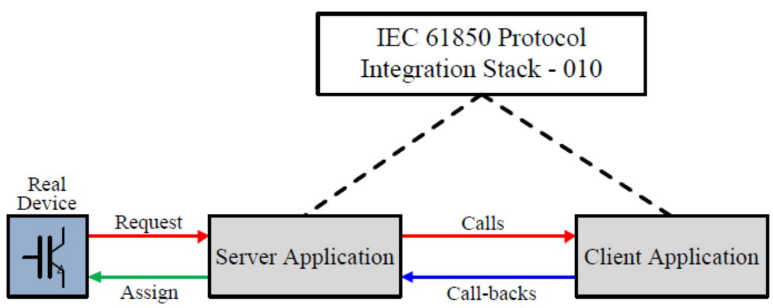


Fig (5.3) Mapping of the Server/Client application to IEC 61850

The integration of a client/server application is possible by using the Protocol Integration Stack (PIS) which includes separately the IED capability description file (ICD), containing the individual data and services that are modeled in a device.

The file contains also the LNs field addresses and unique TCP/IP and MAC address required for the information exchange. The data is accessed between client and server using the special Call and Call-back functions also defined in the Protocol Stack [17]. The general steps in the configuration of server/client required to exchange real time information are shown in Fig (5.4).

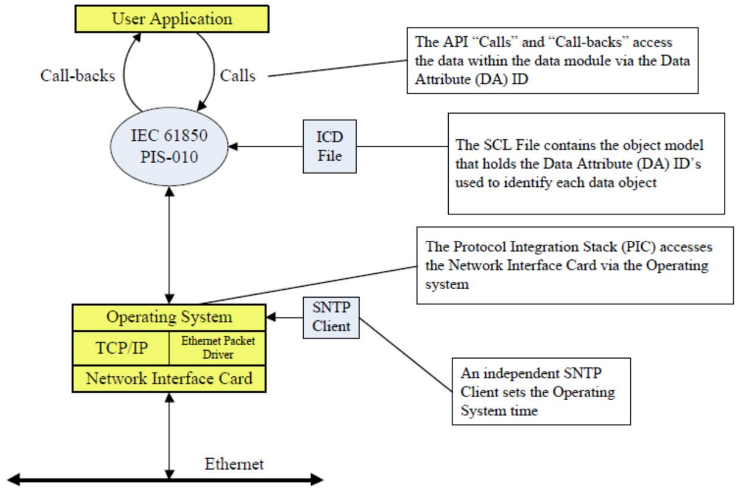
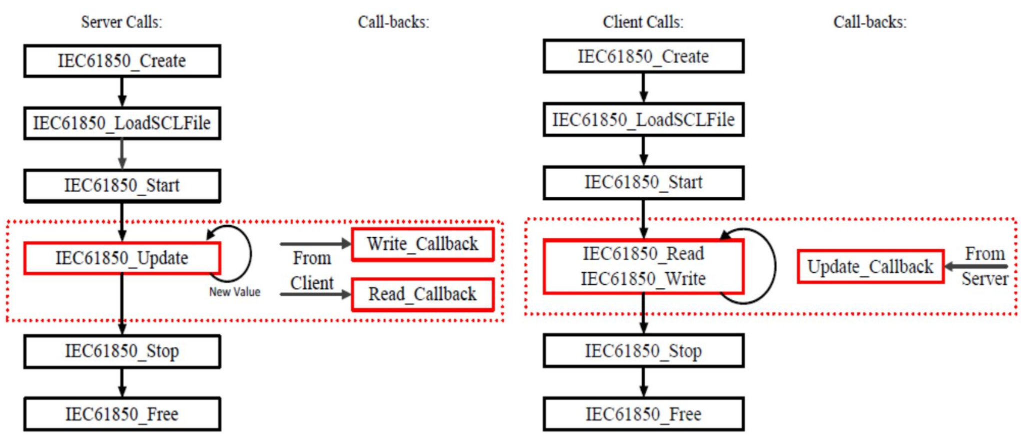


Fig (5.4) Real Time Information Exchange Diagram [42]

In figure 5.5 the functions used in the client/server application is illustrated, presented as they have been used in the modeling process of the application.



Fig(5.5) Server/client management – API functions[42]

Common functions for client and server application [42]:

- IEC61850_Create – create a new IEC 61850 object as client or server.
- IEC61850_LoadSCLFile – load the configured ICD file into the IEC 61850 application.
- IEC61850_Start – start the IEC 61850 console application.
- IEC61850_Stop – stop the IEC 61850 console application.
- IEC61850_Free – free the memory used by IEC 61850 objects.

The specific functions of the Server are [42]:

- IEC61850_Update – used to send the information to the client

- IEC61850_ReadCallback – used to read the information from the client
- IEC61850_WriteCallback – used to write the information from the client

The specific functions of the Client are [42]:

- IEC61850_UpdateCallback – used to receive the information from the server
- IEC61850_Read – used to send readable information to the Server
- IEC61850_Write – used to send writable information to the Server

By using the above described functions, the client and server source codes can be developed. The scope of this research is to development an IEC61850 application based on a hardware platform that supports IEC61850 functionality. So we build an IEC 61850 server, that communicate with a commercial simulation software in another pc as a client (for example IED SCOUT, TEST HARNESS or PCVUE), that mean we will not create a client we will just configure it.

5.4 IEC61850 laboratory setup

To experimentally validate the communication protocol IEC 61850 a laboratory setup was developed as shown in Fig (5.6).

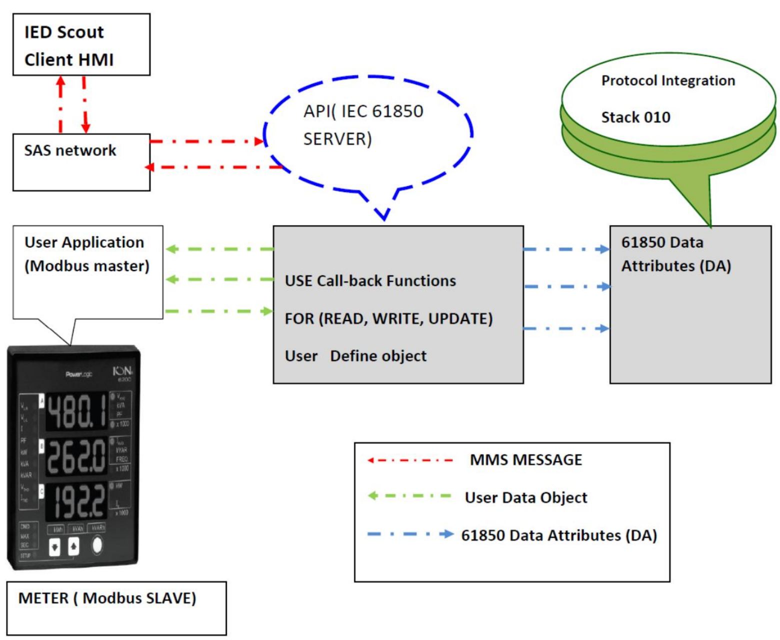


Fig (5.6) IEC61850 laboratory setup

The objective of the laboratory setup is to establish a real time bi-directional data communication between the IEC 61850 Client and IED (server) further show that the communication concept can be successfully applied in power substations. To achieve this we do the following steps:

1. Build Modbus master –slave using meter (ION 6200) as slave.
2. Build SCL file using ICD Designer provided by company SystemCORP
3. Build an IEC61850 Server using (PIS -010) as we mention in section 4.3.
4. Build SCL file using ICD Designer provided by company SystemCORP
5. Use IED Scout as a client HMI running separately on the IED computer.
6. Use Wire shark to capture MMS & GOOSE messages

Also the main hardware components are:

1. Use (ION 6200) power logic meter as iec61850 IED
2. Power supply (110v) DC and serial communication for meter.
3. Two laptops to simulate client/ server communicated together using Ethernet cards.
4. Three phase voltage and current injection device(Omicron)
5. Serial to serial convertor (RS 234/ RS 485) to connect meter with laptop.

5.4.1 Build a Modbus master – slave

One of the challenges that we face in this research is how to test this protocol without using a hardware that support the protocol. To solve this problem we used a Power Logic ION6200 Meter from Schneider electrical company [43]. See figure (5.7)

Fig (5.7)ION6200 Meter [43]

This meter support Modbus communications protocol only to allow measured data and setup information to be efficiently transferred between a Modbus master station and a meter [23]. This includes:

- Interrogation of all data measured by the meter
- Interrogation and control of the meter's digital outputs

The meter used as a Modbus slave device that communicates with a Modbus master which we create it using (NModbus API library version 1.2)[44], C code written by following the steps that shown in the figure (5.8) bellow. For more information about code See appendix(A) .

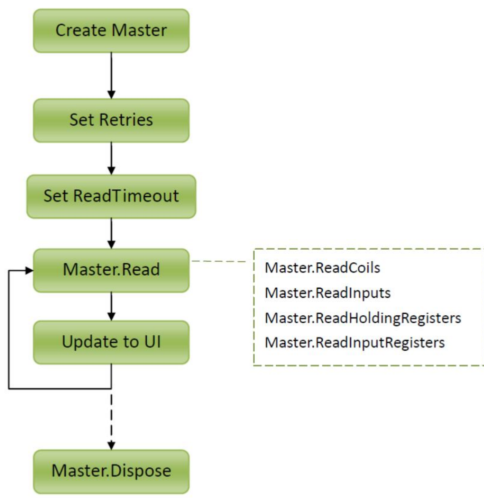


Fig (5.8) N Modbus API library flow chart [44]

5.4.1.1Modbus Protocol:

It's used to establish master-slave/client-server communication between intelligent devices. It truly open and most widely used in the industrial manufacturing environment [23].

5.4.1.2Communication between MODBUS devices

MODBUS devices communicate using a master-slave technique in which only one device (the master) can initiate transactions (called queries). The other devices (slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device (I/O transducer or other measuring device), which processes information and sends its output to the master using MODBUS.Masters can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually, but do not respond to broadcast queries. Modbus serial has two mode RTU and ASCI mode, in this research we use Modbus RTU mode [23]. For more information see chapter three.

5.4.1.3MODBUS REGISTER MAP

MODBUS devices usually include a Register Map. MODBUS functions operate on registers, map registers to monitor, configure, and control module I/O. You should refer to the register map for our device (ION 6200) to gain a better understanding of its operation [45]. Or see Appendix (B).

5.4.2Building IEC61850 Server

By following the steps of building of IEC 61850 server applications which is previously explained in section 4.3, a script code has been developed using the API library provided by SystemCorp. The script flowchart for server application is shown in Figure (5.9).

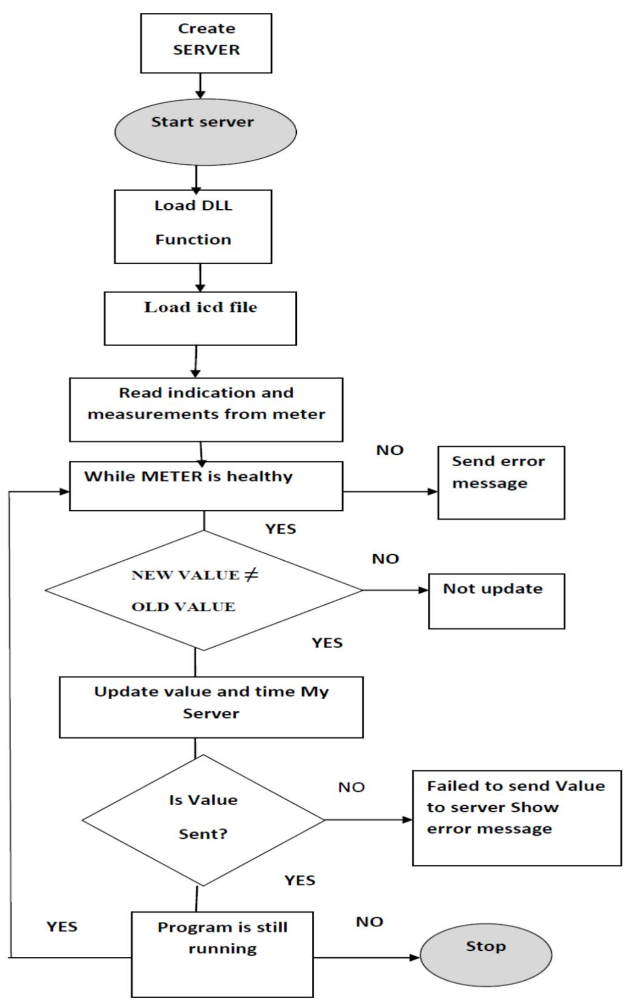


Figure (5.9) Flowchart for server application

The server and client applications are running separately on different computers and exchange information using the Ethernet protocol and the Get/Set functions from the report services.

The application is loading first the DLL functions contained by the PIS10 library where the services are defined, and which are different for both client and server. Afterwards, the preconfigured ICD files containing the data (LDs, LNs and DOs), IP Addresses and activated report services are imported [47].

The server is reading the grid voltage, current and frequency every (1000) sec and if the values change within the prescribed limits (threshold value=0.01*)* then it is transmitted to the client. Also the data is accessed by the receiving end (client) to control the two digital outputs switches.

After every function that has been not successfully executed in the script, a failure message will prompt on the (GUI) informing the user that an error occurred. The errors list is also provided by the SystemCorp API Library and can be found in [42].

An important aspect is represented by the fact that the Callback functions (Update and Write) used to access the information from the dispatcher are updating permanently and are not limited by the iteration number. See code in appendix (A).

5.4.3 Configure (SCL) file

The configuration of the information model and the binding of the model to the real values are described by the System Configuration Language (SCL). The chosen tool used to create the ICD configuration file of the server and client is ICD Designer provided by company SystemCORP . The SCL file is described in a XML format so it can be easily interpreted and transformed. The functions contained by the LNs which are going to be used have to be identified first. The server application contains the following functions:

- AC Voltage, Current and Frequency Measurements.
- Healthy /error and local /remote indications status for IED (ION6200 meter).
- Digital outputs (2) ON/OFF used to simulate CB status (controllable locally only remotely from client)

Figure (5.10) presents the hierarchical structure of the server ICD file. The configuration contains two main sub headers such as “Communication” where the IP Addresses, Gateway and Subnet Mask are included for every particular device connected, and "IED" sub header which comprises the LNs with their specific functionality.

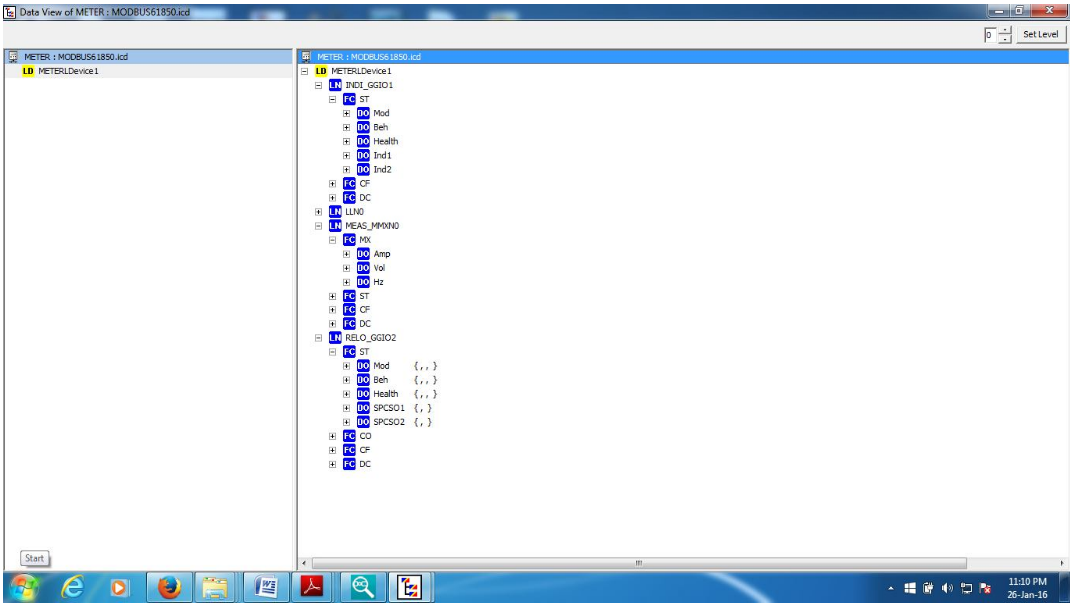


Fig (5.10) Server ICD file

After we build it we need to load it to server (see figure 5.9), for more information about the SCL file see appendix (E).

5.5 Data Modeling

IEC 61850 provides standardized designations for this kind of information. The GGIO LN belongs to the group G (Logical nodes for generic references) and shall be used for modeling inputs and outputs signals as shown in figure 5.10.

The following figure (5.11) of the LN **GGIO** depicts three kinds of information (each having a Data Object (DO): Measured Values (**AnIn**), Controls (SPCSO) and Status Information (**Ind**). Each has a Data Object has a suffix of "1" (e.g., **Ind1**) indicating that a LN **GGIO** have two Data Objects (**Ind1**, **Ind2**). These two Data Objects will represent the local /remote status, meter healthy/error status, the two o/p switches are represented by two Data Objects SPCSO1 and SPCSO2 [33].

GGIO class			
Measured Values			
AnIn1	instMag	AnalogueValue	Analogue input
	q	Quality	
	t	TimeStamp	
	units	Unit	
Controls			
SPCSO1	ctlVal	BOOLEAN	Single point controllable status output
	stVal	BOOLEAN	
	q	Quality	
	t	TimeStamp	
Status information			
Ind1	stVal	BOOLEAN	General indication (binary input)
	q	Quality	
	t	TimeStamp	
IntIn1	stVal	INT32	Integer status input
	q	Quality	
	t	TimeStamp	

Fig (5.11) GGIO logical node[33]

As shown in figure 5.12 bellow each of the three data attributes will be communicated when a client reads the Data Object **Ind1**. The value **stVal** represents the status of single point signal. The other two attributes **q** and **t** are also required to be specified and they need also be stored together with the status value [33].

Ind1	stVal	BOOLEAN
	q	Quality
	t	TimeStamp

Fig (5.12) Indication Data Attribute[33]

5.6 Data Mapping

When the communication stack receives a (GetDataValues) request for the Data Object **INDI_GGIO1.ST.Ind1**, then it needs to get the corresponding reference to the memory locations for the **stVal**, **q** and **t**.

The API use a hierarchy of three fields: Field1 selects the position in the corresponding list between 1 and 3 to determine if it's indication or digital O/P or measurements; Field2 to identify if the value is related to the indication 1 or 2 also it's identify the D O/P number if it's 1 or 2; Field3 references to the values for **stVal**, **q** and **t**. This mapping is unique in order to refer to the correct memory location for any of the values.

For more information see the table (5.1) below: The following figure shows the formal syntax based on the three Fields: Field1 (indications or outputs or measurements), Field2 (number of indication, number of outputs, type of measurements), and Field3 (value, quality or timestamp).

Table (5.1) Data Mapping

Signal	Type	Field 1	Field 2	Field 3
Indication (SP)	Boolean	1	1= local/ remote	1= VALUE 2 = STATUS 3= TIME
			2= Healthy	1= VALUE 2 = STATUS 3= TIME
D O/P (SPCSO)	Boolean	2	1= Digital O/P (1)	1= VALUE 2 = STATUS 3= TIME
			2= Digital O/P (2)	1= VALUE 2 = STATUS 3= TIME
Measurements (MMXU)	float	3	1=Voltage	1= VALUE 2 = STATUS 3= TIME
			2=Current	1= VALUE 2 = STATUS 3= TIME
			3= Frequency	1= VALUE 2 = STATUS 3= TIME

5.7 Data Acquisition or Data Fetch

The communication stack uses the above hierarchical reference (table 5.1) to get the value from the application. The following figure (5.13) shows how the service GetDataValues for **INDI_GGIO1.Ind1.stVal**> is mapped the corresponding memory location (that is bound to the real digital input 1).

The stack uses the local Read (at the API between the communication software and the real application) with the three fields: Field1=1 (means indication), Field2=1 (means Digital indication 1), and Field3=1 (means Value).

The returned value needs to be encoded according to the type of the Logical Node GGIO and **Ind1.stVal**. The type is a **BOOLEAN**. The Value TRUE or FALSE will be returned to the client that issued the GetDataValues request.

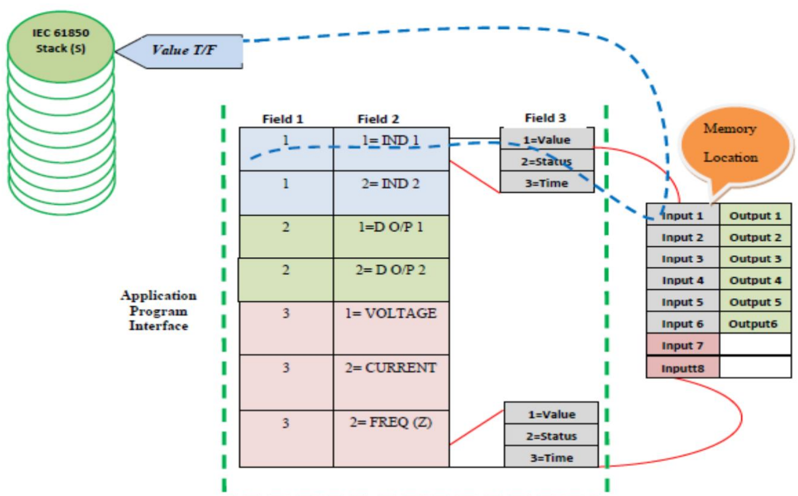


Fig (5.13) Data Acquisition

The mapping table is defined by the system designer of the API between the IEC 61850 communication stack and the real application. It is up to the system designer how to organize the internal communication between the stack and the application. From a communication point of view it is not visible during the exchange of messages with that particular server device.

As shown in the next figure, the client just needs to know the name of the Data Object **INDI_GGIO1.Ind1. stVal**. The name is “translated” into an internal access mechanism using a table to map between the standard name and the memory location of the value. The following figures (5.14) show the relation between the physical I/Os, the internal memory and the name used in the communication. Its maps the physical I/Os and measurement to a list, this list represents the memory location in which values may be stored.

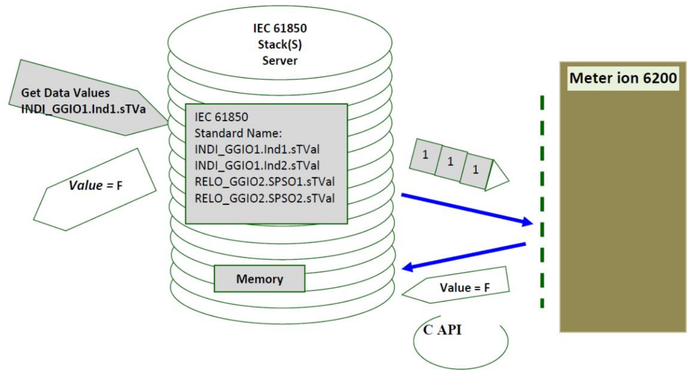


Fig (5.14) Relation Between the physical I/Os

The communication stack may have its own memory that holds a copy of the real value from the application. The copy could be updated cyclically, on request by the communication stack, or by an event from the application. The service <GetDataValues for **INDI_GGIO1.Ind1**> doesn't care about the implementation of the API. The difference between the three update mechanisms is not visible in the standard communication except that the delay is longer or shorter depending on the mechanism used. The timestamp and the quality information associated with the data value are provided the same way as the process value. In this research the server updated the memory tables when the values changed (new values ± old values).

5.8 Configuration of IEDScout and Wireshark

By configure IEDscout as a client we can communicate with the server that we build it before .Also we need to use Wireshark to capture MMS and goose frames.

5.8.1IEDScout

IEDScout is software produced by the company Omicron .It is a universal client to IEC 61850 servers (such as substation IEDs) and a publisher/subscriber for GOOSE messages.In this Thesis, IEDScout v3.0(demo version) has been used.

5.8.2 Wireshark

Wireshark is an open-source network protocol analyzer for UNIX and Windows systems. The latest version of Wireshark can be downloaded from their web site.

Chapter Six

Results and Discussion

Chapter Six

Results and Discussion

6.1 Introduction

IEC 61850 MMS has been briefly discussed in chapter 3. In this section, the protocol will be described in more details as it is necessary for the correct implementation of the IEC 61850 MMS model in the simulator.

6.2 MMS architecture

The MMS architecture is based on a common client-server model. Real devices used in this research contain an MMS server allowing the device to be monitored and managed from an MMS client. As MMS does not specify how to address clients and servers, an entity containing an MMS client or server must rely on the addressing scheme of underlying protocols in the process of establishing an application association to support the MMS environment. In practice, clients and servers are addressed by their IP address and the MMS server uses port number 102 [46].

6.3 Client –Server MMS Message

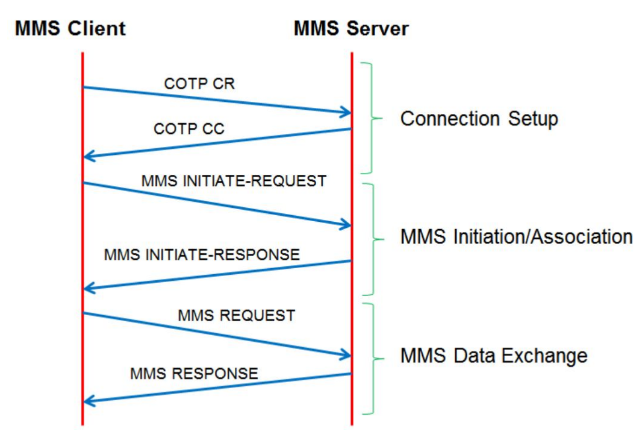


Figure (6.1) MMS messages flow between MMS client and MMS server [46]

The messages flow for the MMS data exchange is illustrated in Figure 6.1. After the normal TCP three-way handshake between the MMS client and MMS server, the COTP layer establishes a connection to transport ISO protocol over TCP by means of the Connection Request message from MMS client and Connection Confirm message from MMS server. Then the MMS Initiation/Association phase begins.

6.6 Research Results:-

The experiment results were collected by using the standard**:**

6.6.1 Build an IEC61850 server that read a grid voltage ,current and frequency and send them to a client (HMI or SCADA) also it receive command to switch on and off two output switch that can be used to control circuit breaker or any other switch see figure (6.2) bellow.

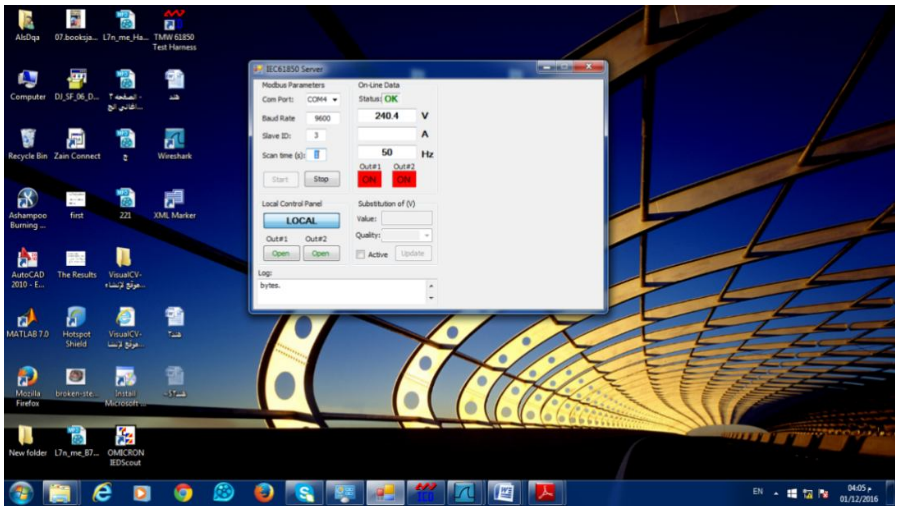


Fig (6.2) IEC61850 developed Server

6.6.2 Testing IEC 61850 client - server communication by using Test harness software as client that read measurement and indication status from server. Also remote control can be done. See figure 6.3

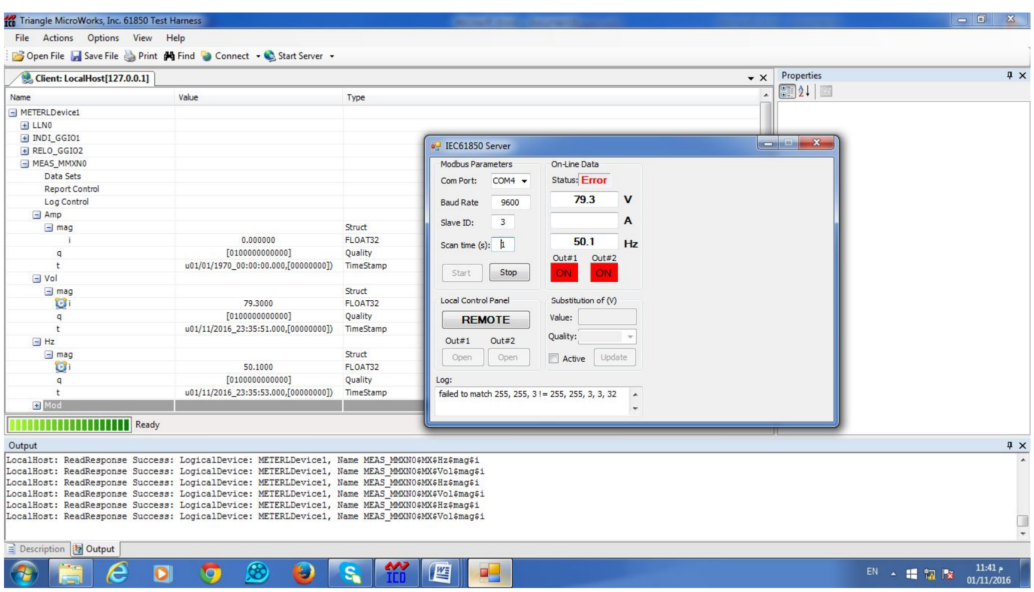


Fig (6.3) client –server communication

6.6.3 Successfully protocol conversion done by using power logic meter (ION 6200) which not support the protocol IEC 61850 , so protocol mapping between Modbus and iec61850 was achieved by wrote a C code to build Modbus master .See appendix (A)

6.6.4 Capture MMS and Goose frames using Wireshark ,see figure (6.4).

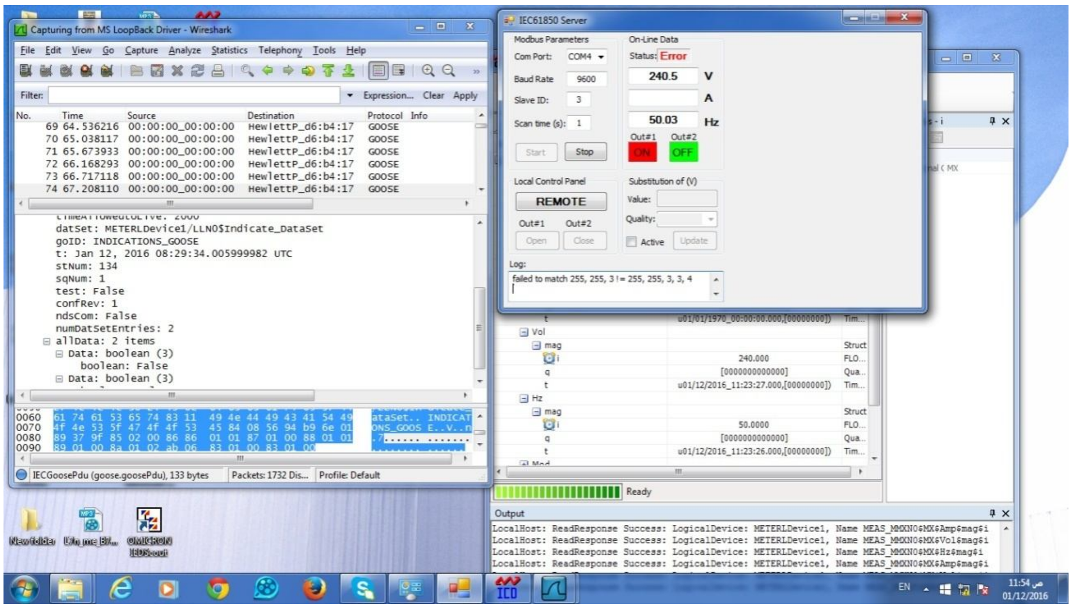


fig (6.4).Goose message send when status change

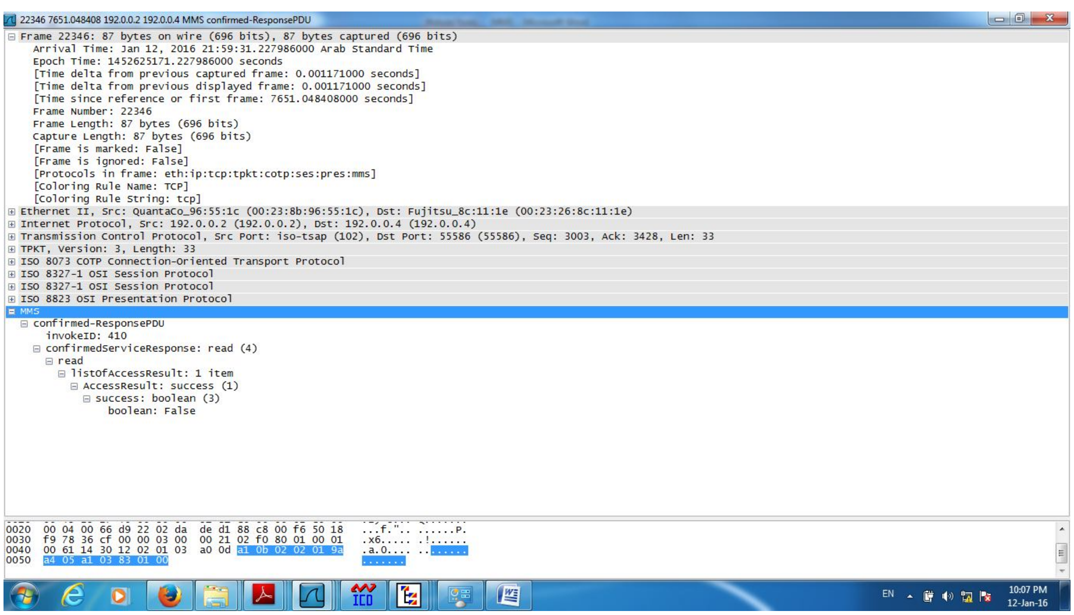


Fig (6.5) MMS message capture when client read the status of digital output (2) status off = false.

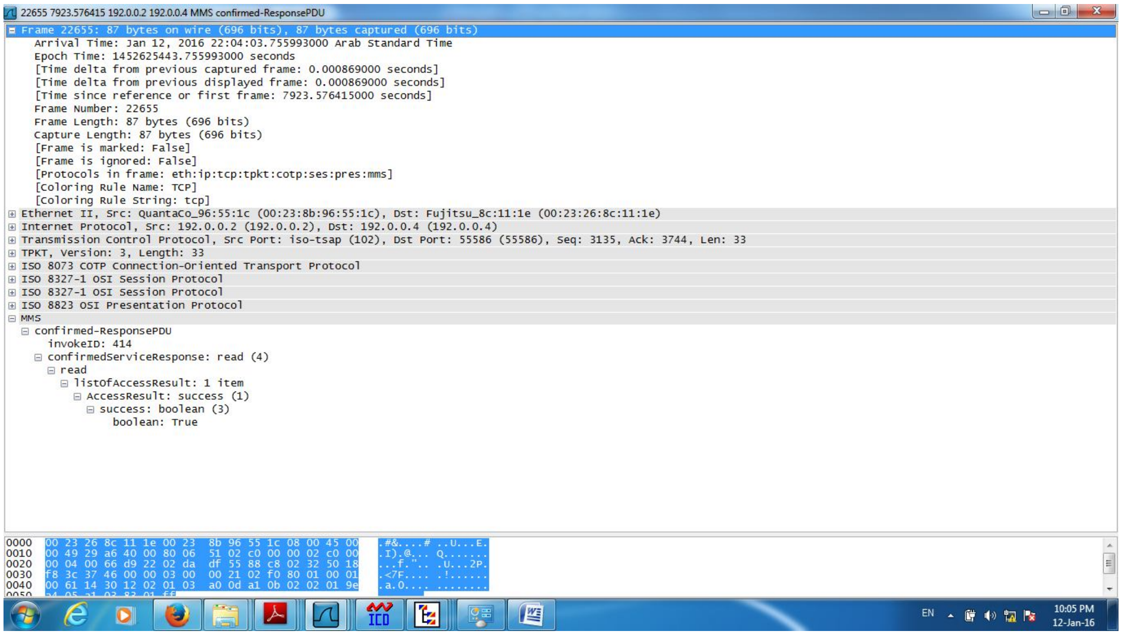


Fig (6.6) MMS message capture when client read the status of digital output (2) status off = ON

6.7 Result Discussion

From the recent result of testing, the two type of IEC 61850 messages can be analysis their frame using Wireshark software easily, and we can know if the data are in correct order or not if the Wireshark capture it without any modification – the new Wireshark versions can detects MMS & goose message – so no need to display the their frame format, or analysis will be as follow:

6.7.1GOOSE messages Analysis

GOOSE message is reflected from application layer to data link layer. It is not across the transmission layer and network layer. The messages contain information that allows the receiving device to know the changed status and the time when it changed. The changed time allows receiving devices to set the relating local time to the receiving events.

Even if there's no change of the status value, GOOSE messages keep sending message in a cycle time. This ensures the activated devices know the current status values of the peer devices. From the data in our exercises, there are many attributes included in GOOSE Control Block. These attributes are mentioned in the analysis block see appendix (G).

Every GOOSE message contains attributes they need. All of GOOSE messages use the same GOOSE structure.

GOOSE frame structure contain of header MAC comprises 12 bytes, first 6 bytes are destination address and the last 6 bytes are source address. Next 14 bytes are Priority tags which present Ethernet type, Application ID, message length and reserved number. Besides the 26 bytes enumerate the fundamental information of the GOOSE message, from byte 27 to the unlimited number of data are decoded with APDU (Application Protocol Data Unit),

6.7.2MMS messages Analysis

From captured frames we relies that MMS message used for control CB or any other switch from HMI(client) by using write command , also we notice MMS transaction corresponding to “Read” command issue from IED SCOUT and to show the acknowledgement message sends back by the Server object.

The mapping of IEC 61850 object and service models to MMS is based on a service mapping where a specific MMS service are chosen as the means to implement the various services of ACSI. For instance, the control model of ACSI is mapped to MMS read and write services. Then the various object models of IEC 61850 are mapped to specific MMS objects. For more information see appendix (C).

Chapter Seven

Conclusion and Future Work

Chapter Seven

Conclusion and Recommendations

7.1 Introduction

The IEC 61850 standard for "Communication networks and systems in substations" (Ed.1) has been progressively released between the years 2003 and 2005. Since then, vendors of substation equipment had implemented their own interpretation of the standard and this has sometimes led to interoperability issues. Currently there are only three companies that develop an IEC 61850 commercially available communication stack with the possibility for use in the public domain. These companies are: SISCO, Triangle Microworks and System Corp

SystemCorp released their first IEC 61850 communication stack (PIS-10) in 2010. New features have been added to the PIS-10 stack and ongoing revisions made based on the feedback provided by research centers such as the CSAEMS at the Cape Peninsula University of Technology and other organizations around the world.

7.2 Conclusions:-

From the problem statement expressed in the first chapter, the project aim to develop and investigated of the IEC 61850 conceptual modeling using SystemCorp communication stack (PIS-10) .The objectives which have been achieved are listed as follows:

- Literature review: summarized the related previous works and review methods design and theory for iec61850 hardware and software implementation.
- Trying different method for IEC 61850 modeling, development and implementation aspects, in addition to that protocol conversion method also presented.
- MMS and GOOSE frame format analysis done using data networking utilities (Wireshark, MMS-Ethereal, Test harness and IEDScout).

The IEC 61850 standard consists of many different parts and some are extremely difficult to understand and interpret even among domain experts.

Students also have to struggle with the challenges presented by the IEC 61850 standard, and this work was to develop a detailed understanding of the message structure communicated between the various devices within the substation network.

There have been significant challenges that have arisen during this research project. These include:

a) Very little detailed implementation acknowledge in the public domain – knowledge resides with vendors only. Almost no implementation work done in the academic sphere that is known of.

b) Very few platforms, lack of tools and forums for the development or adaptation of IEC 61850 solutions in the public domain. Deficiency in documentation, functionality not corresponding with what was marketed, versions of software not being mature and these facts not being communicated to the user community have compounded the problems when doing development, and have been progress at all stages of the project.

In this thesis I presented my understanding of the IEC 61850 standard as well as the design and implementation of a simulation tool. Also, I gave suggestions to protocol conversion method of this standard based on experience and lessons on Modbus , that can be used to convert other legacy protocol and develop this simulation .

7.3 Recommendations

Future directions for research work can be including the following suggestions:

- Investigation of the GOOSE message structure and its application within the IEC61850 standard using two IEC6I850 servers.
- Investigate the development method for “select-before-operate” for an IEC61850 Client Object for HMI (code to be written in C# language in order to be compatible with the PIS-10).
- Investigate the performance and system stability of a protection scheme on which both GOOSE and Sampled Value are used on the process bus.

References:-

[1] R.P.Gopta, "Substation Automation Using Iec61850 Standard", Fifteenth National Power Systems Conference (NPSC), IIT Bombay, December 2008.

[2] R. Mackiewicz, S.Heights," Technical Overview and Benefits of the IEC 61850 Standard for Substation Automation", IEEE PES Power Systems Conference, 2006.

[3] Muhammad Uzair, Communication Methods (Protocols, Format & Language)For The Substation Automation & Control **.**

[4] I. Mesmaeker, P. Rietmann, K. Brand, P.Reinhard," Practical Considerations In Applying IEC 61850 For Protection And Substation Automation Systems" , GCC Power 2005 Conference and Exhibition, Doha, Qatar from 27th – 29th November 2005 .

[5] D. Baigent, M. Adamiak, R. Mackiewicz ," IEC 61850 Communication Networks and Systems In Substations: An Overview for Users",2004.

[6] A Apostolov, Communications in IEC 61850 based substation automation systems, IEEE, Power Systems Conference,2006.

[7] Kostic, Frei,"Modelling and using IEC 61850-7-2 (ACSI) as an API", Power Tech, IEEE Lausanne, 2007, ieeexplore.ieee.org

[8] Y. Liang and R. H. Campbell, "Understanding and Simulating the IEC 61850 Stanard,"2008.[Online].Available:https://www.ideals.illinois.edu/handle/2142/11457. [Accessed: Aug. 23, 2010].

[9] P. M. Kanabar, M. G. Kanabar, W. El-Khattam, T. S. Sidhu andA. Shami, "Evaluation of Communication Technologies for IEC 61850 based Distribution Automation System with Distributed Energy Re-sources," in Power & Energy Society General Meeting, 2009. PES '09. IEEE,pp. 1-8, 26-30 July 2009.

[10] I. Ali and M. S. Thomas, "Substation Communication Networks Architecture," in Power System Technology and IEEE Power India Conference, 2008. POWERCON 2008. Joint International Conference on ,pp. 1-8,12-15 Oct. 2008.

[11] E. In, J .Park, S. Ahn, C. Jang," Design of one chip communication stack processor and MMS communication stack library based on IEC 61850", Networked Embedded Systems for Enterprise Applications (NESEA), 2011 IEEE 2nd International Conference on 8-9 Dec. 2011

[12] Zhang Shujian, Sun Ming, Ge Gaofei," The research based on IEC61850 intelligent terminal of substation ",Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on Date of Conference:27-29 May 2011

[13] J. Park ,IEC 61850 Standard Based MMS Communication Stack Design Using OOP, Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on 26-29 March 2012

[14] Yang. L., "Performance assessment of a IEC 61850-9-2 based protection scheme for a transmission substation", 2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies, December 2011

[15] Mini S. Thomas, Ikbal Ali, Nitin Gupta," IEC 61850 Source Code Implementation for Accessing IED Data", International Journal of Advanced Research in Electrical,Electronics and Instrumentation Engineering ,An ISO 3297: 2007 Certified Organization)Vol. 2, Special Issue 1, December 2013.

[16] T. S. Sidhu and Y. Yin, "IED Modelling for IEC61850 Based Substation Automation System Performance Simulation," in Power Engineering Society General Meeting, 2006. IEEE, pp. 1-7, 18-22 June 2006.

[17] T. S. Sidhu and Y. Yin, "Modelling and Simulation for Performance Evaluation of IEC61850-Based Substation Communication Systems," in IEEE Transactions on Power Delivery, vol. 22, pp. 1482-1489, July2007.

[18] Konka, W., Authur, C.M., and Atkins, R.C., "Traffic Generator of IEC 61850 sampled values"IEEE. in Smart Grid Modelingand Simulation (SGMS), 2011 IEEE First InternationalWorkshop on, 17-17 Oct. 2011, pp. 43–48.

[19] Fan Zhang, Yongli Zhu, Chunyu Yan, Jiangang Bi, Haijun Xiong, Shuai Yuan," A Realization Method of Protocol Conversion Between Modbus and IEC61850" Open Journal of Applied Sciences, 2013, 3, 18-23 doi:10.4236/ojapps.2013.32B004 Published Online June 2013 (http://www.scirp.org/journal/ojapps)

[20] RUS Bulletin, "Design Guide for Rural Substations", June 2001.

[21] Dongles prod foot , "Usa And 61850 For Dummies",2002

[22] K.P. BRAND, C. BRUNNER, W. WIMMER ,"DESIGN OF IEC 61850 BASED SUBSTATION AUTOMATION SYSTEMS", 21, rue d' Artois, F-75008 Paris ,http://www.cigre.org

[23] ACCORDING TO CUSTOMER REQUIREMENTS Introduction to MODBUS, Technical Tutorial, 2002.

[24] Ivan de Masmarker ,” How to use IEC61850 in protection and automation’, KP Brand, C Brunner ,October 2005

[25] "Overview and introduction to MMS " ,SISCO ,revision 2,1995

[26] Jan Tore ,Martin Gilie "a description of the manufacturing message specification (MMS)", 4th international conference 6/8/2007.

[27] Jan Tore ,Martin Gilie"An analysis of the manufacturing message specification protocol ", 5th international conference Oslo 23/5/2008.

[28] Klaus peter Brand, "The standard IEC 61850 as prerequisite for intelligent applications in substations," in Power Engineering Society General Meeting, 2004. IEEE, 2004

[29] John D.mcDONALD."Electric Power Substations Engineering". second edition .2006

[30] IEC61850, "Communication networks and systems for power utility automation - Part 7-1: Principles and models",2003

[31] IEC61850, "Communication networks and systems for power utility automation - Part 7-2 :Abstract communication service interface(ACSI)",2003

[32] IEC61850, "Communication networks and systems for power utility automation - Part 7-3: Common data classes",2003

[33] IEC61850, "Communication networks and systems for power utility automation - Part 7-4: Compatible logical node classes and data object classes," 2003.

[34] IEC61850, "Communication networks and systems for power utility automation - Part 8-1:Specific communication service mapping (SCSM) -Mappings to MMS (ISO 9506-1 and ISO9506-2) and to ISO/IEC 8802-3," 2003.

[35] IEC61850," Specific communication service mapping (SCSM) - sampled values Part 9-1:over serial and unidirectional multi point to point link ," 2003.

[36] IEC61850," Specific communication service mapping (SCSM) - sampled values Part 9-1: over ISO /IEC8802-3," 2003.

[37] R. Mackiewicz and Sterling Heights, Technical Overview and Benefits of the IEC 61850 Standard for Substation Automation,2004

[38] Jianqing Zhangand Carl A. Gunter ,”IEC 61850 Communication Networks and Systems in Substations:An Overview of Computer Science”, ILLINOIS SECURITY LAB

[39] Chul.Hwan Choi, Jong.Ho Park, Jin.Hee Han, Yong.Hark Shin, Yong.Hak Kim and Eung.Bo Shim, "The development of a client conformance tool based on IEC 61850," International Conference on,2011.

[40] Ralph Mackiewicz ”Overview Of Iec61850 And Benefits”, SISCO

[41] Jefrry a.vaughan ,"Demystify IEC61850",ABB.

[42] Balaji Sreenivasan / Blake Myers "IEC 61850 Protocol API User Manual (PIS stack) ",August 2010

[43] Power Logic™ ION6200 Power and Energy Meter Installation & Operation Guide2010

[44] Rnee lin ,NMODBUS,version1.2,2013

[45] powerLogic®ION6200 Serial Communications Protocol and ION / Modbus Register Map

[46] Hammer, E. and Sivertsen, E., 2008. Analysis and implementation of the IEC 61850 standard. Thesis. Technical University of Denmark.

[47] SystemCORP , Getting Started using SystemCORP Embedded TechnologyIEC61850 DLL, 2010

Appendix

Appendix (A)

Code of IEC 61850 server using visual studio version 2013

START S/W

1-Create the server using PIS 10 library

```
Try

    myServer = PIS10.Create(parameters, errorCode)

    in case of error send message

     If myServer = System.IntPtr.Zero Then

        MessageBox.Show(String.Format("Unable to create IEC 61850 Server! Error Code: {0:D}. See contact information to get help.", errorCode)Application.Exit()

            End If
```

1- **Load SCL file**

```
    errorCode = PIS10.LoadSCLFile(myServer, "MODBUS61850.ICD")

  in case of error send message

        If errorCode <> PIS10.ErrorCodes.ERROR_NONE Then
           MessageBox.Show(String.Format("Unable to parse SCL file! Error Code: {0:D}. See contact information to get help.", errorCode))
           Application.Exit()
        End If
```

2- **START THE SERVER**

```
  errorCode = PIS10.Start(myServer)

   in case of error send message

If errorCode <> PIS10.ErrorCodes.ERROR_NONE The MessageBox.Show(String.Format("Unable to start IEC 61850 Server! Error Code: {0:D}. See contact information to get help.", errorCode))

      Application.Exit()

   End If

 Catch

        End Try
```

3- **IEC61850 data set prepare and GUI**

```
Loc = New Dtyps.t61850BooleanObject    // local/remote indication

Health = New Dtyps.t61850BooleanObject   //health indication

Outputs(SPCSO - 1) = New Dtyps.t61850BooleanObject  //2 D O/P

Measurements(MMX_size - 1) = New Dtyps.t61850FLOAT32Object // MEASURMENT
```

```
''GUI prepare
cmbPort.SelectedIndex = 0
listDO.Add(DO1)
listDO.Add(DO2)
listAI.Add(txtAI1)
listAI.Add(txtAI2)
listAI.Add(txtAI3)
listCMD.Add(btnOutp1)
listCMD.Add(btnOutp2)
```

5.Reading from meter using modbus:

```
Public Sub DoWork()
    While blRunning
      Dim OutputStatus(SPCSO - 1) As UShort     // Define The Data Set To Modbus
      Dim fMMxValues(MMX_size - 1) As Single    // Define The Data Set To Modbus
      Try
        'Modbus read the holding register that related to 2 o/ps;
        If serialPort.IsOpen Then
          OutputStatus = master.ReadHoldingRegisters(SlaveID, 2003, 2)
       Modbus read the holding register that related to measurements (15 reading);
        Dim tempMMxValues(15) As UShort // Define An Array To Store The Measurement
          tempMMxValues = master.ReadHoldingRegisters(SlaveID, 99, 16)
          fMMxValues(0) = CSng(tempMMxValues(0) / 10) 'Va
          fMMxValues(1) = CSng(tempMMxValues(8) / 10) 'Ia
          fMMxValues(2) = CSng(tempMMxValues(15) / 100) 'freq
        End If
```

6. Updater the value & time in server:

```
'@ normal operation: Update Values to IEC61850 server if values or qualities changes
If DevHealthy Then
  'update SPCSOs
  For i As Integer = 0 To 1
    If CBool(OutputStatus(i)) <> Outputs(i).iObjectValue Then 'status changes
     Outputs(i).iObjectValue = CBool(OutputStatus(i))
```

```
'update server values and time
UpdateServer(SPCSO, i + 1, 1, PIS10.DataTypes.DATATYPE_BOOLEAN, Outputs(i).iObjectValue)
ConvertTo61850Time(Outputs(i).tObjectTime) 'event time
UpdateServer(SPCSO, i + 1, 3, PIS10.DataTypes.DATATYPE_TIMESTAMP, Outputs(i).tObjectTime)
End If
'GUI update
Me.Invoke(New UpdateGUIDelegate(AddressOf UpdateGUI), New Object() {SPCSO, i})
End If
Next
'update measurements
For i As Integer = 0 To MMX_size – 1
If Math.Abs(Measurements(i).fObjectValue - fMMxValues(i)) >= sgThreshold Then 'status changes
'skip server update if 'V' is substituted
If chkSubstitute.Checked AndAlso i = 0 Then Continue For
'update process value
Measurements(i).fObjectValue = fMMxValues(i)
'UI update
Me.Invoke(New UpdateGUIDelegate(AddressOf UpdateGUI), New Object() {MMX, i})
'update server values and time
UpdateServer(MMX, i + 1, 1, PIS10.DataTypes.DATATYPE_FLOAT32,Measurements(i).fObjectValue)
ConvertTo61850Time(Measurements(i).tObjectTime) 'event time
UpdateServer(MMX, i + 1, 3, PIS10.DataTypes.DATATYPE_TIMESTAMP, Measure-
ments(i).tObjectTime)
End If
Next
Else
End If
DevHealthy = True 'device healthy flag
Catch ex As Modbus.SlaveException
'In case of error The server return error code. 'You can get the function code and exception code.
Dim str As String = ex.Message
str = str.Remove(str.IndexOf("-"))strLastError = DateTime.Now.ToString + " :Modbus Exception
Function code" +str.Substring(str.IndexOf(":"))
DevHealthy = False 'set device healthy flag to error
```

```
'log the last error call update GUI
        Me.Invoke(New UpdateGUIDelegate(AddressOf UpdateGUI), New Object() {LOG, 0})
        Catch ex As Exception
          'Connection exception
          'No response from server.
          'The server maybe close the connection, or response timeout.
          strLastError = DateTime.Now.ToString + " :I/O Error- " + ex.Message
          DevHealthy = False 'set device healthy flag
          'log the last error call update GUI
         Me.Invoke(New UpdateGUIDelegate(AddressOf UpdateGUI), New Object() {LOG, 0})
        End Try
```

7. Send the last data in case of start or stop the meter

```
'@ device startup or stop
        If Health.iObjectValue <> DevHealthy Then
          Init_Reset_Data()
        End If
```

Appendix (B)

ION 6200 meter register addresses

Modbus Addr	Measurement (ION6200 Megawatt meters)	Measurement (all other ION6200 meters)	Format	Scale	Default Scale	Description
40001	serial number	serial number	UINT32	x1		See note 1.
40003	firmware revision	firmware revision	UINT16	x1		
40004	oem identification	oem identification	UINT16	x1		See note 2.
40005	meter options	meter options	UINT32	x1		See note 3.
40007	# meter power ups	# meter power ups	UINT16	x1		
40008	# peak demand resets	# peak demand resets	UINT16	x1		
40009	meter on-time	meter on-time	UINT32	x1		See note 4.
40011	# flash erase cycles	# flash erase cycles	UINT32	x1		
40013	device type	device type	UINT16	x1		See note 5.
40014	Reserved	Reserved				
40015	demand interval down counter	demand interval down counter	UINT16	x1		
40016 - 40099	Reserved	Reserved				
40100	kVln a	Vln a	UINT16	PVS	x10	See notes 6 & 7.
40101	kVln b	Vln b	UINT16	PVS	x10	See note 6.
40102	kVln c	Vln c	UINT16	PVS	x10	See note 6.
40103	kVln avg	Vln avg	UINT16	PVS	x10	See note 6.
40104	kVll ab	Vll ab	UINT16	PVS	x10	
40105	kVll bc	Vll bc	UINT16	PVS	x10	
40106	kVll ca	Vll ca	UINT16	PVS	x10	
40107	kVll avg	Vll avg	UINT16	PVS	x10	
40108	I a	I a	UINT16	PCS	x10	See note 8.
40109	I b	I b	UINT16	PCS	x10	
40110	I c	I c	UINT16	PCS	x10	
40111	I avg	I avg	UINT16	PCS	x10	

Read-Write Control Map

Modbus Addr[18]	Control Parameter (all ION6200 meters)	Format	Scale	Description
42001	Energy Reset	UINT16	x1	Reset MWh, MVAh, and MVARh to 0 (Megawatt meters) Reset kWh, kVAh, and kVARh to 0 (all other models)
42002	Peak Power Demand Reset	UINT16	x1	Reset MW, MVA, and MVAR peak demand to 0 (Megawatt meters) Reset kW, kVA, and kVAR peak demand to (all other models)
42003	Peak Current Demand Reset	UINT16	x1	Reset I peak demand.
42004	Digital Output #1	UINT16	x1	Refer to "Digital Output Control" on page 10.
42005	Digital Output #2	UINT16	x1	

Modbus Addr	Measurement (ION6200 Megawatt meters)	Measurement (all other ION6200 meters)	Format	Scale	Default Scale	Description
40112	I demand	I demand	UINT16	PCS	x10	
40113	I peak demand	I peak demand	UINT16	PCS	x10	
40114	I4	I4	UINT16	PnS	x10	See note 9.
40115	Frequency	Frequency	INT16	x100		
40116	PF sign total	PF sign total	INT16	x100		
40117	PF sign a	PF sign a	INT16	x100		
40118	PF sign b	PF sign b	INT16	x100		
40119	PF sign c	PF sign c	INT16	x100		
40120	MW total	kW total	INT16	PPS	x1	See note 10.
40121	MVAR total	kVAR total	INT16	PPS	x1	
40122	MVA total	kVA total	INT16	PPS	x1	
40123	MW a	kW a	INT16	PPS	x1	
40124	MW b	kW b	INT16	PPS	x1	
40125	MW c	kW c	INT16	PPS	x1	
40126	MVAR a	kVAR a	INT16	PPS	x1	
40127	MVAR b	kVAR b	INT16	PPS	x1	
40128	MVAR c	kVAR c	INT16	PPS	x1	
40129	MVA a	kVA a	INT16	PPS	x1	
40130	MVA b	kVA b	INT16	PPS	x1	
40131	MVA c	kVA c	INT16	PPS	x1	
40132	MW demand	kW demand	INT16	PPS	x1	
40133	MW peak demand	kW peak demand	INT16	PPS	x1	
40134	MVAR demand	kVAR demand	INT16	PPS	x1	
40135	MVA demand	kVA demand	INT16	PPS	x1	
40136	MVAR peak demand	kVAR peak demand	INT16	PPS	x1	
40137	MVA peak demand	kVA peak demand	INT16	PPS	x1	
40138	MWh del	kWh del	UINT32	x1		See note 11.
40140	MWh rec	kWh rec	UINT32	x1		
40142	MVARh del	kVARh del	UINT32	x1		
40144	MVARh rec	kVARh rec	UINT32	x1		
40146	MVAh del+rec	kVAh del+rec	UINT32	x1		
40148	V1 THD	V1 THD	UINT16	x10		
40149	V2 THD	V2 THD	UINT16	x10		
40150	V3 THD	V3 THD	UINT16	x10		
40151	I1 THD	I1 THD	UINT16	x10		
40152	I2 THD	I2 THD	UINT16	x10		
40153	I3 THD	I3 THD	UINT16	x10		
40154	I a demand	I a demand	UINT16	PCS	x10	

Appendix (C)

The IEC 61850 logical device object is mapped to an MMS domain. Table 2 summarizes the mapping of IEC 61850 objects and Table 3

IEC61850 Objects	MMS Object
SERVER class	Virtual Manufacturing Device (VMD)
LOGICAL DEVICE class	Domain
LOGICAL NODE class	Named Variable
DATA class	Named Variable
DATA-SET class	Named Variable List
SETTING-GROUP-CONTROL-BLOCK class	Named Variable
REPORT-CONTROL-BLOCK class	Named Variable
LOG class	Journal
LOG-CONTROL-BLOCK class	Named Variable
GOOSE-CONTROL-BLOCK class	Named Variable
GSSE-CONTROL-BLOCK class	Named Variable
CONTROL class	Named Variable
Files	Files

Table 2.
IEC 61850 to MMS object mapping

IEC 61850 Services	MMS Services
LogicalDeviceDirectory	GetNameList
GetAllDataValues	Read
GetDataValues	Read
SetDataValues	Write
GetDataDirectory	GetNameList
GetDataDefinition	GetVariableAccessAttributes
GetDataSetValues	Read
SetDataSetValues	Write
CreateDataSet	CreateNamedVariableList
DeleteDataSet	DeleteNamedVariableList
GetDataSetDirectory	GetNameList
Report (Buffered and Unbuffered)	InformationReport
GetBRCBValues/GetURCBValues	Read
SetBRCBValues/SetURCBValues	Write
GetLCBValues	Read
SetLCBValues	Write
QueryLogByTime	ReadJournal
QueryLogAfter	ReadJournal
GetLogStatusValues	GetJournalStatus
Select	Read/Write
SelectWithValue	Read/Write
Cancel	Write
Operate	Write
Command-Termination	Write
TimeActivated-Operate	Write
GetFile	FileOpen/FileRead/FileClose
SetFile	ObtainFile
DeleteFile	FileDelete
GetFileAttributeValues	FileDirectory

Table 3.
IEC 61850 services mapping (partial)

Appendix (D)

A.1 IEC 61850 ACSI

The IEC 61850 abstract services are summarized in Table A.1.

Table A.1: ACSI classes, copied from [59]

GenServer model	**LOG-CONTROL-BLOCK model:**
GetServerDirectory	GetLCBValues
	SetLCBValues
Association model	QueryLogByTime
Associate	QueryLogAfter
Abort	GetLogStatusValues
Release	
	Generic substation event model –
GenLogicalDeviceClass model	**GSE**
GetLogicalDeviceDirectory	**GOOSE**
	SendGOOSEMessage
GenLogicalNodeClass model	GetGoReference
GetLogicalNodeDirectory	GetGOOSEElementNumber
GetAllDataValues	GetGoCBValues
	SetGoCBValues
GenDataObjectClass model	**Transmission of sampled values model**
GetDataValues	**MULTICAST-SAMPLE-VALUE-**
SetDataValues	**CONTROL-BLOCK:**
GetDataDirectory	SendMSVMessage
GetDataDefinition	GetMSVCBValues
	SetMSVCBValues
DATA-SET model	**UNICAST-SAMPLE-VALUE-**
GetDataSetValues	**CONTROL-BLOCK:**

Appendix (E)

ICD File


```
<?xml version="1.0" encoding="UTF-8"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Header id="1" revision="1" version="3"/>
  <Communication>
    <SubNetwork name="SubNetworkName">
      <ConnectedAP apName="SubstationRing1" iedName="METER">
        <Address>
          <P type="OSI-AP-Title">1,1,9999,1</P>
          <P type="OSI-AE-Qualifier">12</P>
          <P type="OSI-PSEL">00000001</P>
          <P type="OSI-SSEL">0001</P>
          <P type="OSI-TSEL">0001</P>
          <P type="IP">127.0.0.1</P>
          <P type="IP-SUBNET">255.255.255.0</P>

        </Address>
        <GSE cbName="GSE_CB_GOOSE" ldInst="LDevice1">
          <Address>
            <P type="MAC-Address">02-00-4C-4F--4F-50</P>
            <P type="VLAN-PRIORITY">4</P>
            <P type="VLAN-ID">000</P>
            <P type="APPID">0001</P>
          </Address>
        </GSE>
      </ConnectedAP>
    </SubNetwork>
  </Communication>
  <IED configVersion="1.0" manufacturer="I&amp;C DEPARTMENT" name="METER" type="RTUType">
    <Services/>
    <AccessPoint name="SubstationRing1">
      <Server timeout="30">
        <Authentication/>
        <LDevice inst="LDevice1">
          <LN0 inst="" lnClass="LLN0" lnType="LLN0_0">
            <DataSet name="Measure_DS">
              <FCDA doName="Vol" fc="MX" ldInst="LDevice1" lnClass="MMXN" lnInst="0" prefix="MEAS_"/>
            </DataSet>
            <DataSet name="Indicate_DataSet">
              <FCDA daName="stVal" doName="Ind1" fc="ST" ldInst="LDevice1" lnClass="GGIO" lnInst="1" prefix="INDI_"/>
              <FCDA daName="stVal" doName="Ind2" fc="ST" ldInst="LDevice1" lnClass="GGIO" lnInst="1" prefix="INDI_"/>
            </DataSet>
            <ReportControl confRev="0" datSet="Indicate_DataSet" desc="Unbuf RCB" intgPd="5000" name="UNB_INDICATIONS" rptID="IND_UNBR_ID">
              <TrgOps dchg="true" dupd="true" period="false" qchg="true"/>
              <OptFields configRef="false" dataSet="true" entryID="true" reasonCode="true" seqNum="true" timeStamp="true"/>
              <RptEnabled max="5"/>
            </ReportControl>
            <ReportControl buffered="false" confRev="0" datSet="Measure_DS" desc="UNB_Meas" intgPd="10000" name="UNB_MeasurementsCB" rptID="MEAS_URCB_ID">
```

```xml
          <TrgOps dchg="true" dupd="true" period="false" qchg="true"/>
          <OptFields configRef="false" dataSet="true" entryID="true" reasonCode="true" seqNum="true" timeStamp="true"/>
          <RptEnabled max="5"/>
        </ReportControl>
        <GSEControl appID="INDICATIONS_GOOSE" confRev="1" datSet="Indicate_DataSet" desc="For GOOSE" name="GSE_CB_GOOSE" type="GOOSE"/>
      </LN0>
      <LN inst="1" lnClass="GGIO" lnType="GGIO_1" prefix="INDI_">
        <DOI name="Ind1">
          <DAI name="stVal">
            <Private type="SystemCorp_Generic">
              <SystemCorp_Generic:GenericPrivateObject xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="1" Field2="1" Field3="1" Field4="0" Field5="0"/>
            </Private>
          </DAI>
          <DAI name="q">
            <Private type="SystemCorp_Generic">
              <SystemCorp_Generic:GenericPrivateObject xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="1" Field2="1" Field3="2" Field4="0" Field5="0"/>
            </Private>
          </DAI>
          <DAI name="t">
            <Private type="SystemCorp_Generic">
              <SystemCorp_Generic:GenericPrivateObject xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="1" Field2="1" Field3="3" Field4="0" Field5="0"/>
            </Private>
          </DAI>
        </DOI>
        <DOI name="Ind2">
          <DAI name="stVal">
            <Private type="SystemCorp_Generic">
              <SystemCorp_Generic:GenericPrivateObject xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="1" Field2="2" Field3="1" Field4="0" Field5="0"/>
            </Private>
          </DAI>
          <DAI name="q">
            <Private type="SystemCorp_Generic">
              <SystemCorp_Generic:GenericPrivateObject xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="1" Field2="2" Field3="2" Field4="0" Field5="0"/>
            </Private>
          </DAI>
          <DAI name="t">
            <Private type="SystemCorp_Generic">
              <SystemCorp_Generic:GenericPrivateObject xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="1" Field2="2" Field3="3" Field4="0" Field5="0"/>
            </Private>
          </DAI>
        </DOI>
      </LN>
      <LN inst="2" lnClass="GGIO" lnType="GGIO_0" prefix="RELO_">
        <DOI name="SPCSO1">
          <DAI name="q">
            <Private type="SystemCorp_Generic">
```

```
            <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="2"
Field2="1" Field3="2" Field4="0" Field5="0"/>
          </Private>
        </DAI>
        <DAI name="t">
          <Private type="SystemCorp_Generic">
            <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="2"
Field2="1" Field3="3" Field4="0" Field5="0"/>
          </Private>
        </DAI>
        <DAI name="ctlModel">
          <Val>direct-with-normal-security</Val>
        </DAI>
        <SDI name="Oper">
          <DAI name="ctlVal">
            <Private type="SystemCorp_Generic">
              <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="2"
Field2="1" Field3="1" Field4="0" Field5="0"/>
            </Private>
          </DAI>
        </SDI>
      </DOI>
      <DOI name="SPCSO2">
        <DAI name="q">
          <Private type="SystemCorp_Generic">
            <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="2"
Field2="2" Field3="2" Field4="0" Field5="0"/>
          </Private>
        </DAI>
        <DAI name="t">
          <Private type="SystemCorp_Generic">
            <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="2"
Field2="2" Field3="3" Field4="0" Field5="0"/>
          </Private>
        </DAI>
        <DAI name="ctlModel">
          <Val>direct-with-normal-security</Val>
        </DAI>
        <SDI name="Oper">
          <DAI name="ctlVal">
            <Private type="SystemCorp_Generic">
              <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic" Field1="2"
Field2="2" Field3="1" Field4="0" Field5="0"/>
            </Private>
          </DAI>
        </SDI>
      </DOI>
    </LN>
    <LN desc="Non phase related Measurement" inst="0" lnClass="MMXN"
lnType="MMXN_0" prefix="MEAS_">
      <DOI desc="Measured value" name="Amp">
        <DAI name="q">
          <Private type="SystemCorp_Generic">
```

```
          <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"   Field1="3"
Field2="2" Field3="2" Field4="0" Field5="0"/>
        </Private>
      </DAI>
      <DAI name="t">
        <Private type="SystemCorp_Generic">
          <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"   Field1="3"
Field2="2" Field3="3" Field4="0" Field5="0"/>
        </Private>
      </DAI>
      <SDI name="mag">
        <DAI name="i">
          <Private type="SystemCorp_Generic">
            <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"   Field1="3"
Field2="2" Field3="1" Field4="0" Field5="0"/>
          </Private>
        </DAI>
      </SDI>
    </DOI>
    <DOI desc="Measured value" name="Vol">
      <DAI name="q">
        <Private type="SystemCorp_Generic">
          <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"   Field1="3"
Field2="1" Field3="2" Field4="0" Field5="0"/>
        </Private>
      </DAI>
      <DAI name="t">
        <Private type="SystemCorp_Generic">
          <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"   Field1="3"
Field2="1" Field3="3" Field4="0" Field5="0"/>
        </Private>
      </DAI>
      <SDI name="mag">
        <DAI name="i">
          <Private type="SystemCorp_Generic">
            <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"   Field1="3"
Field2="1" Field3="1" Field4="0" Field5="0"/>
          </Private>
        </DAI>
      </SDI>
    </DOI>
    <DOI desc="Measured value" name="Hz">
      <DAI name="q">
        <Private type="SystemCorp_Generic">
          <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"   Field1="3"
Field2="3" Field3="2" Field4="0" Field5="0"/>
        </Private>
      </DAI>
      <DAI name="t">
        <Private type="SystemCorp_Generic">
          <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"   Field1="3"
Field2="3" Field3="3" Field4="0" Field5="0"/>
        </Private>
```

```
                </DAI>
                <SDI name="mag">
                  <DAI name="i">
                    <Private type="SystemCorp_Generic">
                      <SystemCorp_Generic:GenericPrivateObject
xmlns:SystemCorp_Generic="http://www.systemcorp.com.au/61850/SCL/Generic"    Field1="3"
Field2="3" Field3="1" Field4="0" Field5="0"/>
                    </Private>
                  </DAI>
                </SDI>
              </DOI>
            </LN>
          </LDevice>
        </Server>
      </AccessPoint>
    </IED>
    <DataTypeTemplates>
      <LNodeType id="MMXN_0" lnClass="MMXN">
        <DO desc="Controllable enumerated status" name="Mod" type="ENC_0"/>
        <DO desc="Enumerated status" name="Beh" type="ENS_0"/>
        <DO desc="Enumerated status" name="Health" type="ENS_0"/>
        <DO desc="Logical Node name plate" name="NamPlt" type="LPL_0"/>
        <DO desc="Measured value" name="Amp" type="MV_0"/>
        <DO desc="Measured value" name="Vol" type="MV_0"/>
        <DO desc="Measured value" name="Hz" type="MV_0"/>
      </LNodeType>
      <LNodeType id="GGIO_0" lnClass="GGIO">
        <DO name="Mod" type="INC_0"/>
        <DO name="Beh" type="INS_0"/>
        <DO name="Health" type="INS_0"/>
        <DO name="NamPlt" type="LPL_0"/>
        <DO name="SPCSO1" type="SPC_0"/>
        <DO name="SPCSO2" type="SPC_0"/>
      </LNodeType>
      <LNodeType id="GGIO_1" lnClass="GGIO">
        <DO name="Mod" type="INC_0"/>
        <DO name="Beh" type="INS_0"/>
        <DO name="Health" type="INS_0"/>
        <DO name="NamPlt" type="LPL_0"/>
        <DO name="Ind1" type="SPS_0"/>
        <DO name="Ind2" type="SPS_0"/>
      </LNodeType>
      <LNodeType id="LLN0_0" lnClass="LLN0">
        <DO name="Mod" type="INC_1"/>
        <DO name="Beh" type="INS_2"/>
        <DO name="Health" type="INS_1"/>
        <DO name="NamPlt" type="LPL_0"/>
      </LNodeType>
      <DOType cdc="MV" desc="Measured value" id="MV_0">
        <DA bType="Struct" dchg="true" fc="MX" name="mag" type="mag_0"/>
        <DA bType="Quality" fc="MX" name="q" qchg="true"/>
        <DA bType="Timestamp" fc="MX" name="t"/>
      </DOType>
      <DOType cdc="LPL" desc="Logical Node name plate" id="LPL_0">
        <DA bType="VisString255" fc="DC" name="vendor"/>
        <DA bType="VisString255" fc="DC" name="swRev"/>
        <DA bType="VisString255" fc="DC" name="d"/>
      </DOType>
      <DOType cdc="ENS" desc="Enumerated status" id="ENS_0">
        <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="Mod"/>
        <DA bType="Quality" fc="ST" name="q" qchg="true"/>
```

```xml
    <DA bType="Timestamp" fc="ST" name="t"/>
  </DOType>
  <DOType cdc="ENC" desc="Controllable enumerated status" id="ENC_0">
    <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="Mod"/>
    <DA bType="Quality" fc="ST" name="q" qchg="true"/>
    <DA bType="Timestamp" fc="ST" name="t"/>
    <DA bType="Enum" dchg="true" fc="CF" name="ctlModel" type="ctlModel"/>
  </DOType>
  <DOType cdc="SPC" id="SPC_0">
    <DA bType="Struct" fc="CO" name="Oper" type="Oper_0"/>
    <DA bType="Quality" fc="ST" name="q" qchg="true"/>
    <DA bType="Timestamp" fc="ST" name="t"/>
    <DA bType="Enum" fc="CF" name="ctlModel" type="CtlModels"/>
  </DOType>
  <DOType cdc="INS" id="INS_0">
    <DA bType="Enum" dchg="true" fc="ST" name="stVal"/>
    <DA bType="Quality" fc="ST" name="q" qchg="true"/>
    <DA bType="Timestamp" fc="ST" name="t"/>
  </DOType>
  <DOType cdc="INC" id="INC_0">
    <DA bType="Enum" dchg="true" fc="ST" name="stVal"/>
    <DA bType="Quality" fc="ST" name="q" qchg="true"/>
    <DA bType="Timestamp" fc="ST" name="t"/>
    <DA bType="Enum" fc="CF" name="ctlModel" type="CtlModels"/>
  </DOType>
  <DOType cdc="SPS" id="SPS_0">
    <DA bType="BOOLEAN" dchg="true" fc="ST" name="stVal"/>
    <DA bType="Quality" fc="ST" name="q" qchg="true"/>
    <DA bType="Timestamp" fc="ST" name="t"/>
  </DOType>
  <DOType cdc="INS" id="INS_1">
    <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="Health"/>
    <DA bType="Quality" fc="ST" name="q" qchg="true"/>
    <DA bType="Timestamp" fc="ST" name="t"/>
  </DOType>
  <DOType cdc="INS" id="INS_2">
    <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="Mod"/>
    <DA bType="Quality" fc="ST" name="q" qchg="true"/>
    <DA bType="Timestamp" fc="ST" name="t"/>
  </DOType>
  <DOType cdc="INC" id="INC_1">
    <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="Mod"/>
    <DA bType="Quality" fc="ST" name="q" qchg="true"/>
    <DA bType="Timestamp" fc="ST" name="t"/>
    <DA bType="Enum" fc="CF" name="ctlModel" type="ctlModel"/>
  </DOType>
  <DAType id="origin_0">
    <BDA bType="Enum" name="orCat" type="OrCat"/>
    <BDA bType="Octet64" name="orIdent"/>
  </DAType>
  <DAType id="mag_0">
    <BDA bType="FLOAT32" name="i"/>
  </DAType>
  <DAType id="Oper_0">
    <BDA bType="BOOLEAN" name="ctlVal"/>
    <BDA bType="Struct" name="origin" type="origin_0"/>
    <BDA bType="INT8U" name="ctlNum"/>
    <BDA bType="Timestamp" name="T"/>
    <BDA bType="BOOLEAN" name="Test"/>
    <BDA bType="Check" name="Check"/>
  </DAType>
```

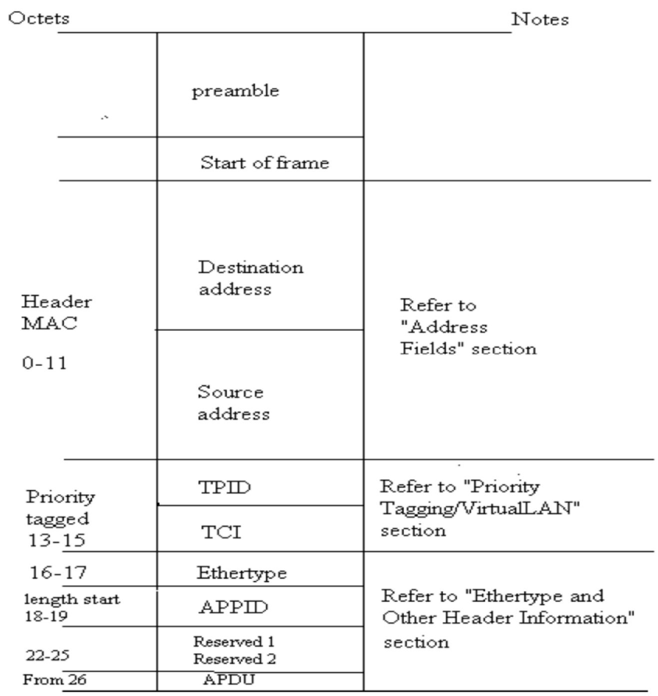
```xml
      <EnumType id="Mod">
        <EnumVal ord="1">on</EnumVal>
        <EnumVal ord="2">blocked</EnumVal>
        <EnumVal ord="3">test</EnumVal>
        <EnumVal ord="4">test/blocked</EnumVal>
        <EnumVal ord="5">off</EnumVal>
      </EnumType>
      <EnumType id="ctlModel">
        <EnumVal ord="0">status-only</EnumVal>
        <EnumVal ord="1">direct-with-normal-security</EnumVal>
        <EnumVal ord="2">sbo-with-normal-security</EnumVal>
        <EnumVal ord="3">direct-with-enhanced-security</EnumVal>
        <EnumVal ord="4">sbo-with-enhanced-security</EnumVal>
      </EnumType>
      <EnumType id="CtlModels">
        <EnumVal ord="0">status-only</EnumVal>
        <EnumVal ord="1">direct-with-normal-security</EnumVal>
        <EnumVal ord="2">sbo-with-normal-security</EnumVal>
        <EnumVal ord="3">direct-with-enhanced-security</EnumVal>
        <EnumVal ord="4">sbo-with-enhanced-security</EnumVal>
      </EnumType>
      <EnumType id="OrCat">
        <EnumVal ord="0">not-supported</EnumVal>
        <EnumVal ord="1">bay-control</EnumVal>
        <EnumVal ord="2">station-control</EnumVal>
        <EnumVal ord="3">remote-control</EnumVal>
        <EnumVal ord="4">automatic-bay</EnumVal>
        <EnumVal ord="5">automatic-station</EnumVal>
        <EnumVal ord="6">automatic-remote</EnumVal>
        <EnumVal ord="7">maintenance</EnumVal>
        <EnumVal ord="8">process</EnumVal>
      </EnumType>
      <EnumType id="Health">
        <EnumVal ord="1">Ok</EnumVal>
        <EnumVal ord="2">Warning</EnumVal>
        <EnumVal ord="3">Alarm</EnumVal>
      </EnumType>
    </DataTypeTemplates>
</SCL>
```

APPENDIX (G)

Goose frame format

ISO/IEC 8802-3 frame format

(Reference: International Standard IEC61850-5)



GOOSE message detected from exercise using vamp relay

No.	GOOSE message	Description
1.	01 0c cd 01 00 00	Destination Address for GOOSE
2.	00 1a d3 00 19 97	Source Address for Relay
3.	88 b8	Ether type Values (GOOSE)
4.	00 04	Application ID 4
5.	00 59	Length 89
6.	00 00	Reserved 1
7.	00 00	Reserved 2
8.	61 53	Implicit Sequence, APDU length 83 bytes
9.	80 09	GOOSE Control Block Reference, 9 bytes
10.	4c 4c 4e 30 24 67 63 62 31	LLNO$gcbl (Logic node0 GOOSE control block)
11.	81 01	Time to Live, 1 byte
12.	14	20 ms
13.	82 15	Data Set, 21 bytes
14.	56 61 6d 70 5f 32 52 65 6c 61 79	Vamp_Relay
15.	2f 4c 4c 4e 30 24 44 53 47 31	/LLNO$DSG1
16.	83 04	GOOSE ID, 4 bytes
17.	56 41 4d 50	VAMP
18.	84 08	Time and Quality, 8 bytes
19.	4b bb 5d 76 b9 37 48 00	date: 14, 03, 2010, time: 05: 28: 04.41099977493 Quality: 00
20.	85 01	Status Number, 1 byte
21.	34	52
22.	86 01	Sequence Number, 1 byte
23.	0d	13
24.	87 01	test, 1byte
25.	00	0
26.	88 01	Configuration Revision, 1 byte
27.	01	1
28.	89 01	needs Commission, 1 byte
29.	00	0
30.	8a 01	Number of Data Set Entries, 1 byte
31.	03	3
32.	ab 09	Boolean Values, 9 bytes
33.	83 01	
34.	00	Boolean Value=0
35.	83 01	
36.	00	Boolean Value=0
37.	83 01	
38.	00	Boolean Value=0