

Simulation of IEC-61850 Substation Using Virtualization and Open Source Software

A Project Report

Submitted by

A. Aravinda Prasad

(EE16B131)

*in partial fulfillment of the requirements
for the award of the degree of*

**BACHELOR OF TECHNOLOGY IN ELECTRICAL ENGINEERING
AND
MASTER OF TECHNOLOGY IN ELECTRICAL ENGINEERING
(DUAL DEGREE)**



**DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS**

JUNE 2021

THESIS CERTIFICATE

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my guide Dr.K.S. Swarup for providing me the opportunity to work under his guidance. I am grateful to him for providing his highly valuable inputs, insights and feedback on my work and for his time and support throughout the project.

I am indebted to the Electrical Engineering Department, IIT Madras for providing me the resources. I would also like to thank my family and friends for their continuous support

A. Aravinda Prasad

ABSTRACT

Key Words: *IEC-61850; Substation automation; Raspberry-Pi; Beaglebone Black; Embedded Systems; libiec61850; IED; Linux; OpenSource.*

With the rise in need for electricity and the associated problems with it, a new and innovative grid system has been developed called the smart grid. Addition of sensors for real time monitoring of the grid, distributed generation and storage system and permitting two-way power flow.

All the information collected is transmitted via network and is reached the grid control center. This whole process is facilitated by devices spread across the grid called Intelligent Electronic Devices. They collect voltage, current and power data from the sensors and store it in their local storage and act as server when the client tries to communicate with it.

This project aims to create an IED completely based on open-source hardware and software for laboratory micro-grids. Availability of low cost controllers and ease of software integration will help in quick and efficient deployment of a working IED which can used for laboratory purposes.

Contents

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
CONTENTS	iii
LIST OF TABLES AND FIGURES	0

1. Introduction

1.1 Advent of Smart Grids	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Scope of Work	3
1.5 Organization of Thesis	4

2. Literature Survey of IEC-61850 standard for Substation Automation

2.1 Introduction	5
2.2 Objectives and Advantages of IEC-61850 standard	5
2.3 Substation Architecture	6
2.4 Data and Function Modeling	8
2.5 Data Objects and Attributes	9
2.6 Communication	12
2.7 Substation Configuration Language	13
2.8 Conclusion	16

3. Review of Security Vulnerabilities in IEC-61850	
3.1 Introduction	17
3.2 Security Vulnerabilities	17
3.3 Different Types of Security Attacks	18
3.3.1 Denial of Service (DOS)	18
3.3.2 Data Interception and Tinkering	18
3.3.3 Viral Firmware	19
4. Overview of the Test Platform	
4.1 Design of the Experiment Platform	21
4.2 Creation of Server Instance	22
4.3 Structure of the IED	23
4.4 Summary of Software Tools	24
5. Simulation of IED and Results	25
6. Conclusion	32
References	33

LIST OF TABLES AND FIGURES

FIGURES

Fig 2.1	3 level hierarchical architecture	8
Fig 2.2	Visualization of IED	9
Fig 2.3	Example of naming convention of Dos	11
Fig 2.4	Communication Stack of IEC-61850	12
Fig 2.5	SCL file exchange inside a substations	14
Fig 3.1	Packet Sniffing	19
Fig 3.2	An exposed serial port	20
Fig 3.3	Kali Linux OS showcasing WiFi hacking tools	20
Fig 4.1	Raspberry Pi	22
Fig 4.2	Beaglebone Black	22
Fig 4.3	Server Flowchart	23
Fig 4.4	Structure of the IED device used	24
Fig 5.1	The main routine of the server instance	25
Fig 5.2	MMS server started at IP 192.168.43.152	26
Fig 5.3	The client side receiving real time data from the simulated server	27
Fig 5.4	LPHD1 LN transferring vital information about IED	27
Fig 5.5	Wireshark capture of the MMS packets	28
Fig 5.6	Packet dissection using the Wireshark	29
Fig 5.7	GOOSE message capture using Wireshark	30
Fig 5.8	GOOSE packet dissection	31

TABLES

Table 2.1	Structure of DA of a “Pos” DO	11
Table 2.2	Different parts of the IEC-61850 Standard	16

Chapter 1

INTRODUCTION

1.1 The Advent of Smart Grids

The new age Smart Grid is a culmination of power transmission, faster communication and data processing. Traditional electric grids were often modeled as a center for one way power flow. The authority over the transfer and consumption of electricity was limited to the metering device fitted at the consumer end.

Today, after witnessing the exponential growth in technology the traditional power grid system has had a tremendous changeover or it has been made “*smart*”. What makes a grid smart is the duplex communication between the source and its consumers and the real-time monitoring of the transmission lines. The amount of processing power used for monitoring of a sub-station today is several folds higher than the computers used on the Voyager.

This means that complex communication protocols can be used in the system. IEC-61850 is an international communication standard developed especially for the intelligent electronic devices (IED) placed at the substation. It uses the TCP/IP networks and high speed LAN switches for the required response times. It also guarantees interoperability within devices manufactured by different vendors.

1.2 Motivation

Communication protocols used in IEC-61850 are built upon the Ethernet and TCP/IP protocol. This means that the standard will pack itself with vulnerabilities and is exposed to the same threats as any network would be.

Although research for defending mainline networks is going on, there is no sufficient work done in the area of protection of substation or the methods are too old to fight back new attacks. This is mainly due to the lack of laboratory scale substation test beds and DIY devices. All the commercial IED's are proprietary both in hardware and software. These devices are expensive when bought in small number, only large orders are subsidized by the vendor.

Hence the only way is to develop or “*make*” the devices in-house. Thankfully, with the power of open source software and the availability of low cost hardware we can make our own IED compliant to IEC-61850 standard.

1.3 Objectives

The objective of this thesis is to develop an intelligent electronic device used in substation for automation that is compliant to IEC-61850 standard using open-source software tools and low cost hardware available in the market.

1.4 Scope of Work

This thesis thoroughly reviews the fundamentals of the IEC standard through various research papers and manuals, the software stack required for the operation of the device and the hardware available on the market with good technical support online and proposes a method to successfully deploy a device capable of emulating an actual IED.

1.5 Organization of Thesis

Chapter 1 provides a brief introduction on the smart grid and the motivation behind the thesis.

Chapter 2 covers in depth about the fundamentals of the IEC-61850 standard, various modeling of data and the device used for transmission.

Chapter 3 explains various security vulnerabilities faced by the standard and the research work done so far.

Chapter 4 gives a preview of the hardware and software stack used for the development of the device.

Chapter 5 captures the test run done with PC and emulation of the device using virtual machine.

Chapter 6 concludes the work and provides insights into scope of future work.

Chapter 2

Literature Survey of IEC-61850 standard for Substation Automation

2.1 Introduction

IEC 61850 is an encapsulation of international standards defining how to describe the modern devices in automated electrical substation and how to exchange the information between these smart devices. Before the provision of the standard, the concept of inter-operability between various vendors was virtually impossible. Companies had developed their own proprietary standard for communication between devices, this limited the options for the customers. After the standard was developed, full fledged data communication between different manufacturers was made possible and software support was also quickly developed. The complete functionality of the standard will be introduced in this chapter.

2.2 Objectives and Advantages of IEC-61850 Standard

International Electrotechnical Commission(IEC) and Electric Power Research Institute collaborated to develop this standard merging all existing protocols. The standard implemented well known Transmission control protocol/Internet protocol(TCP/IP), Manufacturing Message Specification(MMS) and Extensible Markup

Language(XML). The objectives were:

- Support for integration between multi-vendor devices
- Support for high speed data transfers
- Support for scalability and flexibility of substation systems
- Support for accommodating new technologies.

Once after it's implementation, the construction costs have significantly reduced compared to conventional substations. Usage of network driven message transfer instead of hard-wired has resulted in savings during design, installation, commissioning and operation. The benefits of the standard include:

- High speed data exchange between devices
- P2P communication
- Multi-vendor inter-operability
- XML based Substation Configuration Language(SCL) which enables sharing of information between different software tools
- Object oriented and hierarchical data model helps in modeling data and functions

2.3 Substation Architecture

The IEC-61850 standard classifies the substation system into 3 different hierarchical levels, **Station**, **Bay** and **Process**.

Station level refers to the Human Machine Interface(HMI), Gateway

and the station machines. The main functions of this level is to provide interface between the HMI and SCADA network.

Bay level responsible for the protection, measurement and control IED's. The functions at this level are intra and inter level communications via various data buses.

Process level consists of basic equipment of control and monitor such as current and voltage transformers, circuit breakers. Primary function at this level is to convert the analog signals measured into digital values for transmission and activate the breakers whenever command issued from the above levels. Merging Units(MU) is another device that merges and synchronizes analog values and sends them to destination IEDs.

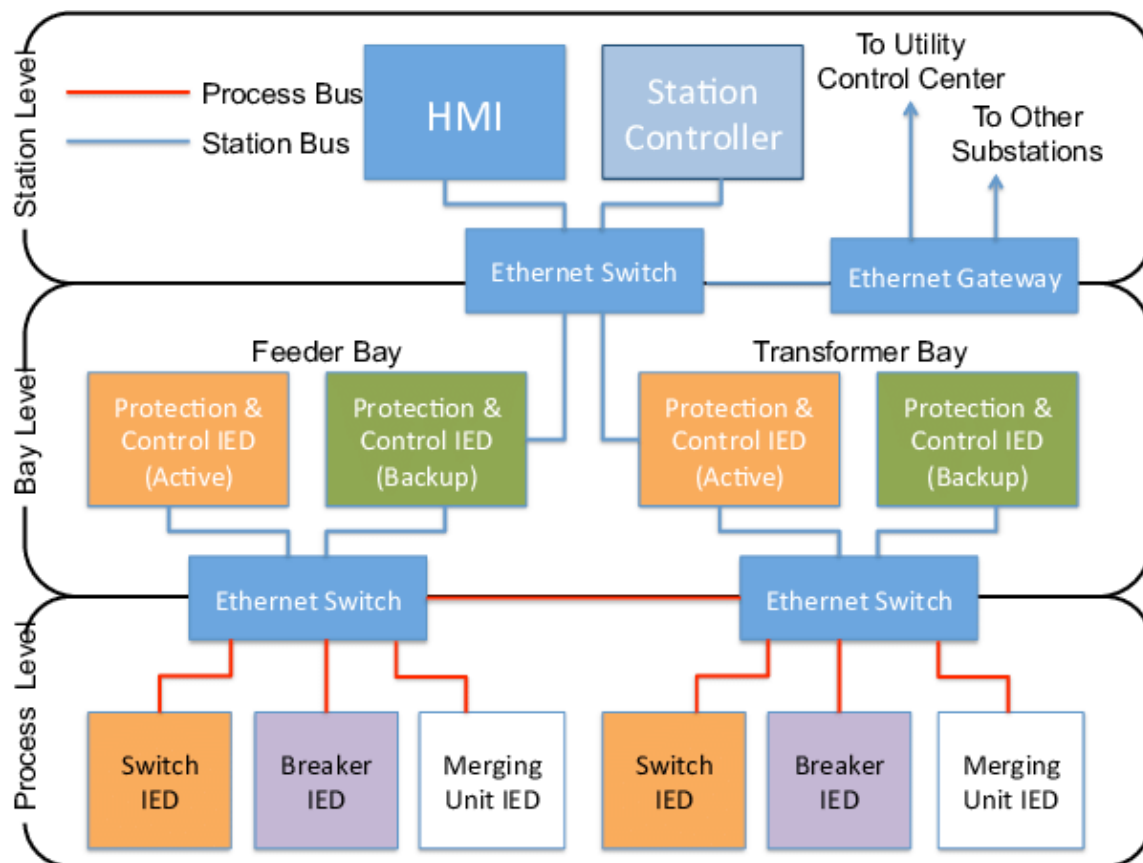


Fig.2.1: 3 level hierarchical architecture

2.4 Data and Function Modeling

The standard specifies a hierarchical way of data modeling in devices. The smallest entity to transfer data is called a **Logical Node(LN)** and the standard defines over 90LNs to enable all necessary function in the SA system.

LPHD(Logical Node for Physical Device) is a unique node whose function is to monitor and communicate critical information about the IED itself such as hardware health, name plate, status of power supply.

LN0 is a special node present in all logical device and contains data common for all nodes such as data sets, logs, GOOSE/GSSE control blocks and Sampled Value(SV) control blocks.

All these nodes are present inside a **Logical Device(LD)**. Logical devices in turn are encapsulated inside a physical device. There might be multiple LDs inside a physical device for several applications such as measurement, protection and control.

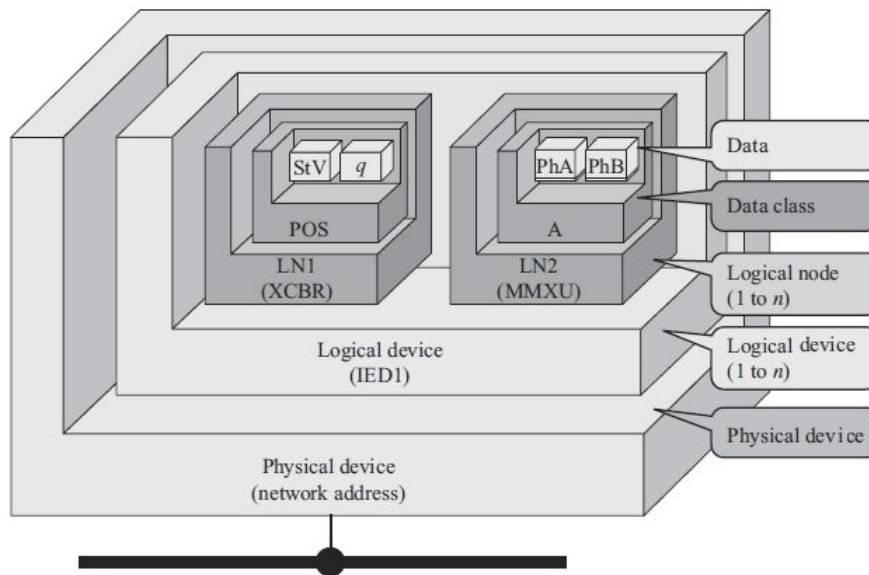


Fig.2.2: Visualization of IED

2.5 Data Objects and Attributes

The data that the nodes handle are classified into multiple **Data Objects(DO)** which further contains various **Data Attributes(DA)** associated with it. For example, the LN class CMMUX which is used to represent the current measurement unit inside a LD has the following DO's,

- Mode (Mod); Describes operation mode
- Behavior (Beh);
- Health (Health); Describes the health status if the device
- Name Plate (NamPlt); Shows technical details of the function
- Phase (A); Phase of the quantity

The standard defines over 30 different types of fundamental DOs. They are the building blocks of more complex DOs. Some examples for common DOs are, Double Point Control (DPC), Single Point Status (SPS), Single Point Setting (SPG), etc...Each of the common DOs contains DAs with a type (DAType) that belongs to a set of **Functional Constraint(FC)**. Below table shows the attributes with corresponding functional constraints of a DO to a Switch position (Pos). The standard also provides rules in forming standardized names for the devices used, the below figure explains how a DO and its corresponding DAs are named.

Table 2.1: Structure of DA of a "Pos" DO

DPC Class		
Data Object "Pos"		
Data Attribute Name	Attribute Type	Functional Constraint
stVal	BOOLEAN	ST (Status)
q	Quality	ST (Status)
t	TimeStamp	ST (Status)
subEna	BOOLEAN	SV (Substituted Value)
SubQ	Quality	SV (Substituted Value)
d	VISIBLE STRING255	DC (Description)
dU	UNICODE STRING255	DC (Description)
cdcName	VISIBLE STRING255	EX (Extension)

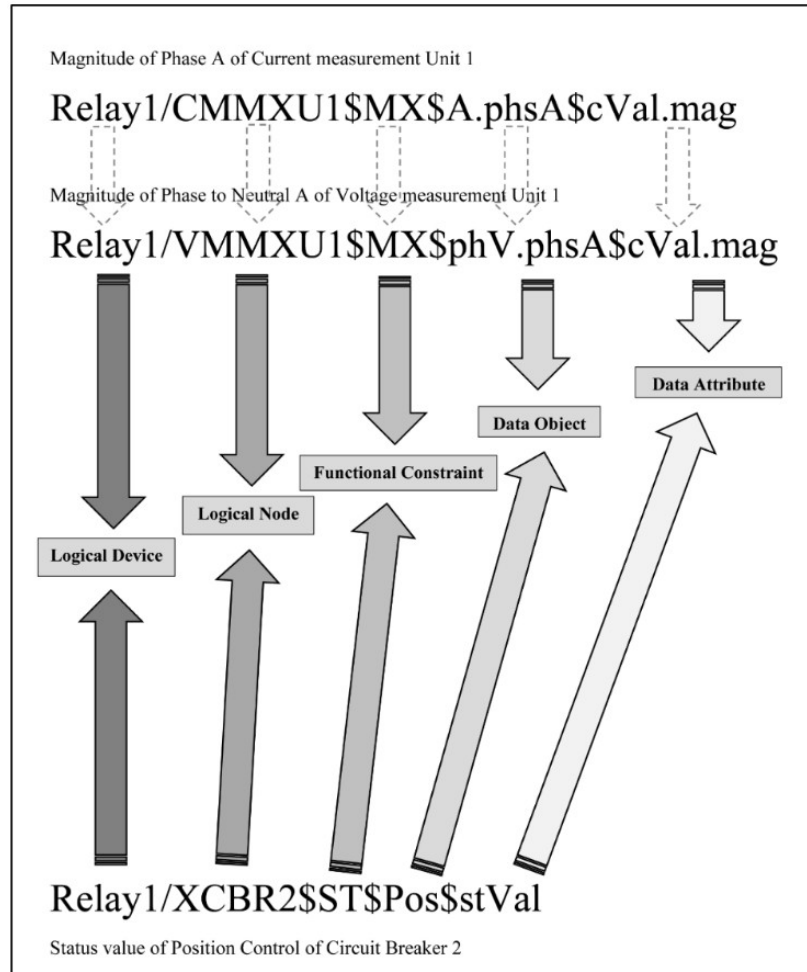


Fig.2.3: Example IEC-61850 naming convention of DOs

2.6 Communication

The highlighting feature of the IEC-61850 standard is the clearly defined protocols and its associated communication stack which enables the inter-operability of multi vendor devices. The mapping of abstract objects and service to real protocols is represented in the figure below.

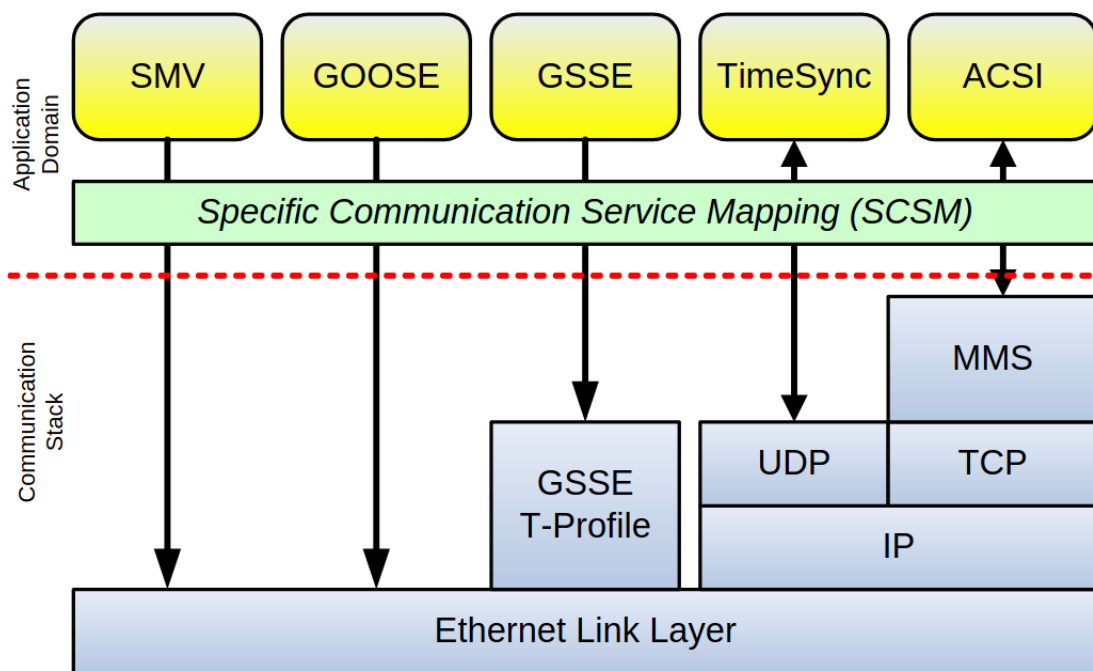


Fig.2.4: Communication Stack of IEC-61850

Transfer of large amount of payload consisting of measurement data and issuing of time-critical control signals are the two main types of messaging done in a SA system. The Client/Server communication is extremely flexible as the communication is driven by the Client side.

This communication is mapped to the MMS protocol through the TCP/IP and ISO CO protocol. The ACSI(Abstract Communications Standard Interface) defines the services that provide Client/Server type interaction. MMS is chosen for this communication because it supports complex naming conventions. But this affects the time taken for each transaction as it relies on all the seven layers in the OSI(Open System Interconnection) stack. On the other hand the time-critical communication such as the control signals uses the GSE(Generic Substation Event). This maps directly onto the Ethernet link layer without mapping to any other protocols and eventually helps us in reducing the number of physical wiring.

2.7 Substation Configuration Language

Most important entity in a SA system are the IEDs. They form the backbone for the complete automation of the grid system. The standard defines an XML(eXtensible Markup Language) based language to describe the IED used in the system. This allows the customers to use any software capable of reading the file to analyze the device or even develop their own software. These SCL files are classified into many types based on what they describe.

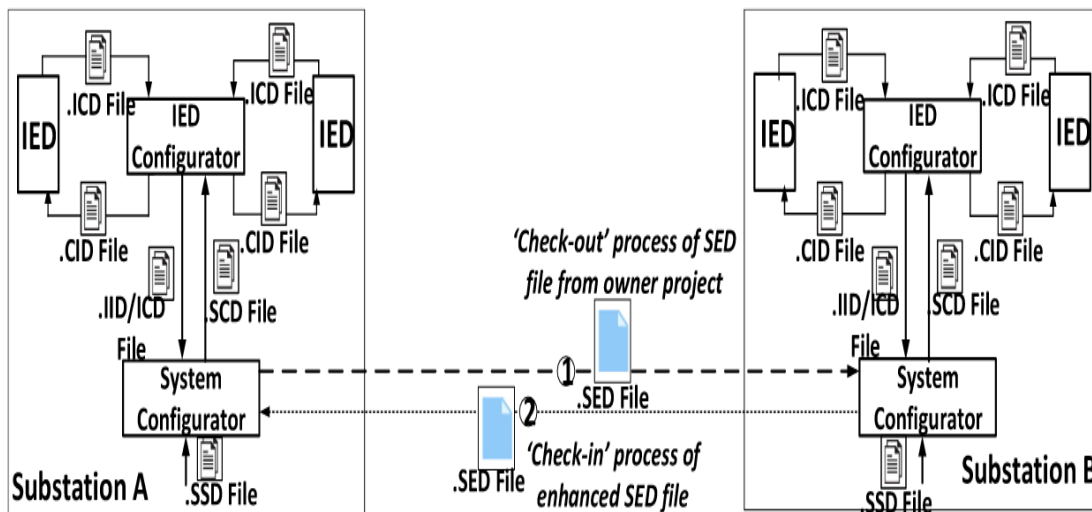


Fig.2.5: SCL file exchange inside a substation

IED Capability Description(ICD)

It defines the complete capability of an IED. The manufacturer of the device must supply this file to map complete system configuration. This file is sent from an IED configurator to system configurator during the setting up of the station.

System Specification Description(SSD)

This file contains the entire specification of the of the Substation Automation System including single line diagram for the substation and it functionalities(LNs)

Instantiated IED Description(IID)

This file contains all the instantiated objects obtained from the data template and is used when certain edits are carried out in the IED. This means that any modification needed in the capability of an IED in later stage of the design process can be done through this file.

Configured IED Description(CID)

This file contains complete information about an IED and is shared with every other IED in the system. It also contains information about the substation that is necessary for an IED.

System Exchange Description(SED)

This file is used whenever two different substation needs to communicate with each other. The system configurators needs to these files to understand the layout of the other system for seamless communication.

2.8 Conclusion

This chapter provides a comprehensive review of the IEC-61850 standard. Only the important topics pertaining to the project has been dealt in this chapter. The standard documentation of the IEC-61850 is very detailed whose contents are listed in the table below.

Table 2.2: Different parts of the IEC-61850 Standard

Different parts of the IEC-61850 Standard	
IEC 61850-5:2013	Communication requirements for functions and device models
IEC 61850-6:2009	Configuration language for communication in electrical substations related to IEDs
IEC 61850-7-1:2011	Basic communication structure - Principles and models
IEC 61850-7-2:2010	Basic communication structure - Abstract communication service interface (ACSI)
IEC 61850-7-3:2010	Basic communication structure - Common Data Classes
IEC 61850-7-4:2010	Basic communication structure - Compatible logical node classes and data classes

Chapter 3

Review of Security Vulnerabilities in IEC-61850

3.1 Introduction

The previous chapter provides us with the core features of the IEC standard, primarily it clarifies how inter-operability is achieved. However such sophistication does entails a series of security vulnerabilities. This chapter briefly covers the security flaws present in the standard and different kinds of attacks that can inflicted upon any power grid that adopts the it.

3.2 Security Vulnerabilities

The standard culminates a plethora of protocols stacked up over each other to provide a communication platform. But whenever multiple protocols work in various levels the one with weakest security measure would be the strength of the whole interface. So hence all the protocols must be tuned to work at same level of security. Improper authentication, lack of encryption of critical data and missing support for message integrity check are some of the vulnerability due to protocol security mismatch.

3.3 Different Types Of Security Attacks

The above mentioned vulnerabilities act as a gateway for security attacks. Since the standard uses Ethernet networking, substations are now a prime target to orchestrate lethal security attacks. There are many different types of attacks on large networks some of which are mentioned below.

3.3.1 Denial of Service (DOS)

Denial of service attack is a simulated by disrupting the connection between the server and client by keeping the service provider busy. A malicious node acting as a source of attack can be easily plugged into the local grid network and can send requests continuously to the server which keeps it busy serving that malicious node and not responding to the actual client which in this case is the control center of the grid. Several services include FTP, HTTP, TELNET etc...

3.3.2 Data Interception and Tinkering

The standard does not specify any encryption during the transmission of data. This makes it extremely easy for attackers to sniff out the data packets being transferred over the network and not even leave a trace of that. Sometimes attackers can intercept the critical messages being transferred sometimes even tinkering the payload and re-transmitting it which leads to improper functioning and even complete shutdown of the grid.

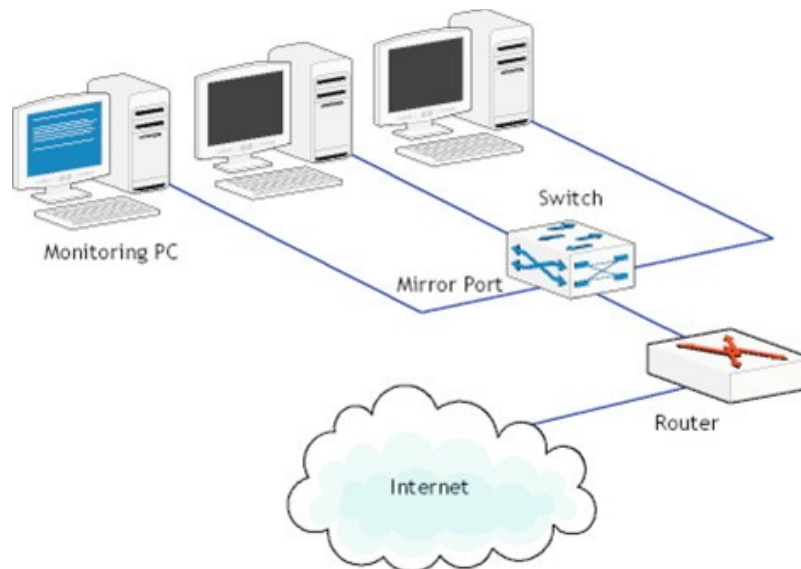


Fig.3.1: Packet Sniffing

3.3.3 Viral Firmware

This type of attack is much different than the previous ones because it is not abusing the network but the device itself. The embedded devices used as IEDs across the grids have very little power to withhold attackers from completely accessing its file system. An exposed serial port (UART) in a device can be used as a backdoor to the system. Kali Linux, an infamous Linux distribution is packed with all the necessary tools for hardware hacking. Once access is granted to the root file system, the firmware present can be easily reverse engineered using software tools and analyze their working. Then a malicious firmware could be updated onto it which can instructed to work in ways the attacker wants it to. Although this is a much harder way of hacking into the system, the ready to use tools and online support for hacking made it simple for attackers to implement it.

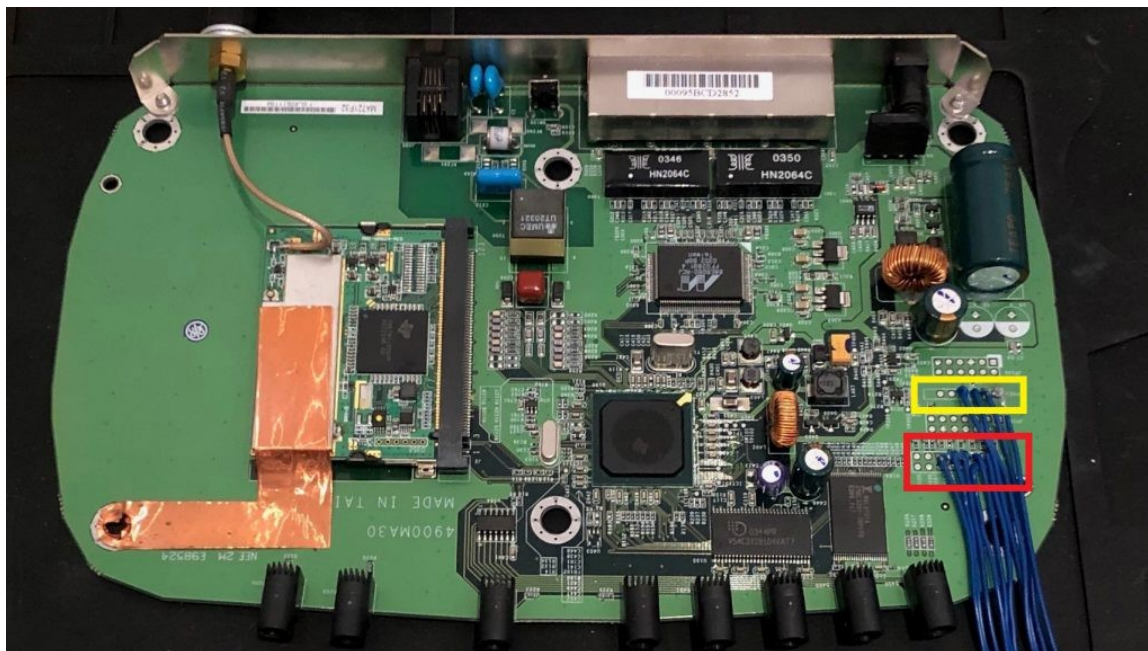


Fig.3.2: An exposed serial port

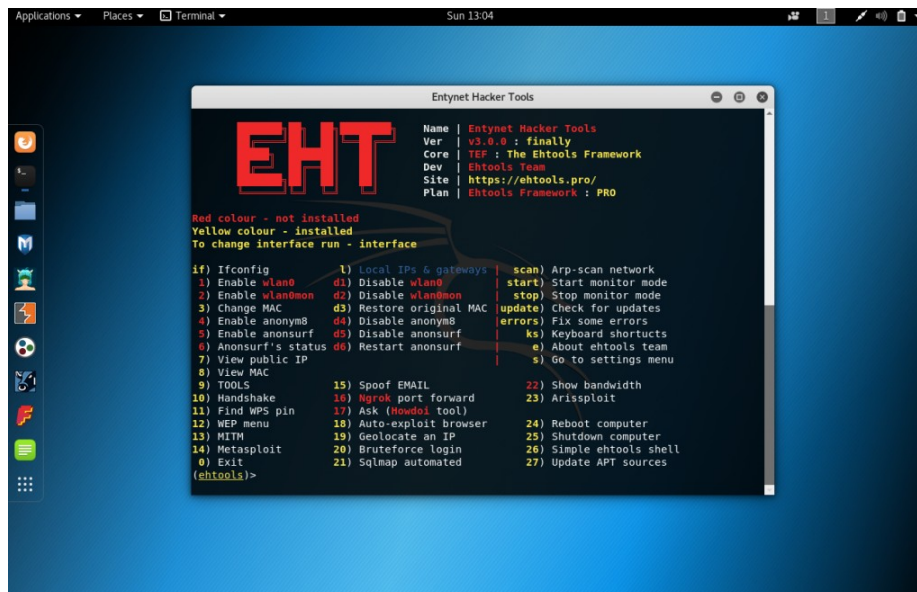


Fig.3.3: Kali Linux OS showcasing WiFi hacking tools

Chapter 4

Overview of the test platform

4.1 Design of the Experiment Platform

In this project, virtualization has been used to realize an IED device, as the IED firmware will be running on top of the Operating System. The virtual machine will be running the Linux operating system which is also used on the popular IOT hardware such as Raspberry Pi and Beagle Bone Black. Our server instance application built on the open-source libiec61850 framework will be running on that machine. At station level, the Human Machine Interface is simulated on the IEDScout software which will use Client/Server communication with the bay level IEDs. The communication message between the workstation and the IEDs are via MMS packets. The communication between process level and bay level devices consists of Merging Units which communicates with the IEDs using a SV Publisher/Subscriber model and GOOSE protocol. A SV subscriber instance is also run on the virtual machine. At last the whole setup is monitored by a Network Analyzer through Wireshark software and it helps us sniff the MMS and SV message packets going through the network. Since the IEDs at bay level have duplex communication, the device that runs it must have dual Network Interface Card present.



Fig.4.1: Raspberry Pi



Fig.4.2: Beaglebone Black

4.2 Creation of Server Instance

The algorithm to create a fully functional server instance carries over a 5-step process. The first step is to create a device model using the model generator tool provided with the libiec61850 framework and importing it into the source code. Then creation of a server object using the imported device model. Once server service is enabled it starts listening for client connections continuously until it finds one. Then the data transfer begins and runs as long as we want it to. There must also be a provision for stopping of services and destroying of server objects.

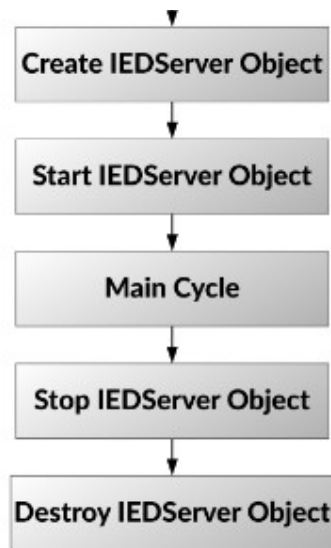


Fig.4.3: Server Flowchart

4.3 Structure of the IED

In this experiment, a generic template of an IED with single Logical Device (LD) and that encapsulates 3 Logical Nodes namely, LLN0, GGIO1, LPHD1. The ICD file is visualized using the OMICRON software.

simpleIO • Data Model • GenericIO		
LD	simpleIOGenericIO	
LN	LLN0	Logical node zero
LN	GGIO1	Generic process I/O
LN	LPHD1	Physical device information

simpleIO • Data Model • GenericIO • LLN0		
LN	LLN0 Logical node zero	
Name		Value
DO	Mod	
DA	stVal	[ST]
DA	q	[ST]
DA	t	[ST]
DA	ctlModel	[CF] status-only
DO	Beh	
DA	stVal	[ST]
DA	q	[ST]
DA	t	[ST]
DO	Health	
DA	stVal	[ST]
DA	q	[ST]
DA	t	[ST]
DO	NamPlt	MZ Automation
DA	vendor	[DC] MZ Automation
DA	swRev	[DC] 1.3.0
DA	d	[DC] libiec61850 server example
DA	configRev	[DC]
DA	ldNs	[EX]

simpleIO • Data Model • GenericIO • GGIO1		
LN GGIO1 Generic process I/O		
Name		Val
▶ DO AnIn1		
▶ DO AnIn2		
▶ DO AnIn3		
▶ DO AnIn4		
▶ DO Mod		
▶ DO Beh		
▶ DO Health		
▶ DO SPCSO1		
▶ DO SPCSO2		
▶ DO SPCSO3		
▶ DO SPCSO4		
▶ DO Ind1		
▶ DO Ind2		
▶ DO Ind3		
▶ DO Ind4		
▶ DO NamPit		

simpleIO • Data Model • GenericIO • LPHD1		
LN LPHD1 Physical device information		
Name		Val
◀ DO PhyHealth		
DA stVal	[ST]	
▶ DA q	[ST]	
▶ DA t	[ST]	
◀ DO Proxy		
DA stVal	[ST]	
▶ DA q	[ST]	
▶ DA t	[ST]	
◀ DO PhyNam		
DA vendor	[DC]	

Fig.4.4: Structure of the IED device used

4.4 Summary of Tools

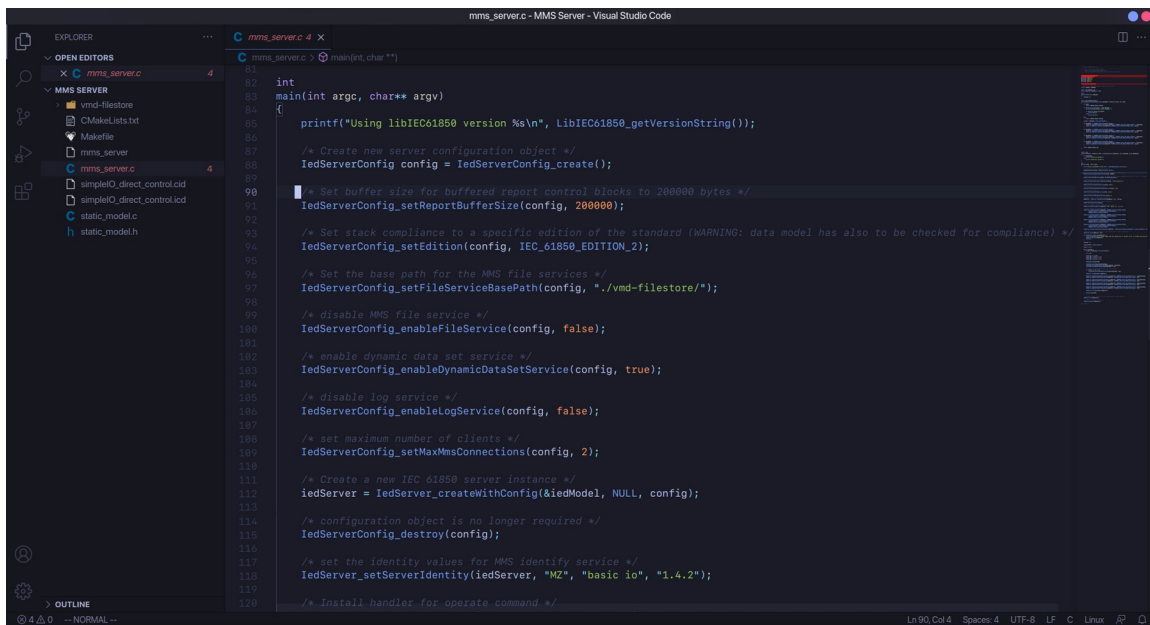
A brief summary of tools required for this experiment.

- Two PC's with virtualization enabled on one of them and running Linux operating system on it.
- Libiec61850 frame work with C build tools which can be installed through the command **sudo apt-get install build-essential**.
- OMicron software for simulation of client side workstation.
- Wireshark software for Analyzing the Network

Chapter 5

Simulation of the IED

The MMS server code is setup according to the flowchart in Fig.4.3. Next the server code is compiled using the build tools and run as sudo user to gain permission to open ports for listening. The server exposes itself with the same IP as of the system it is running as the client and server are connected on the same local network.

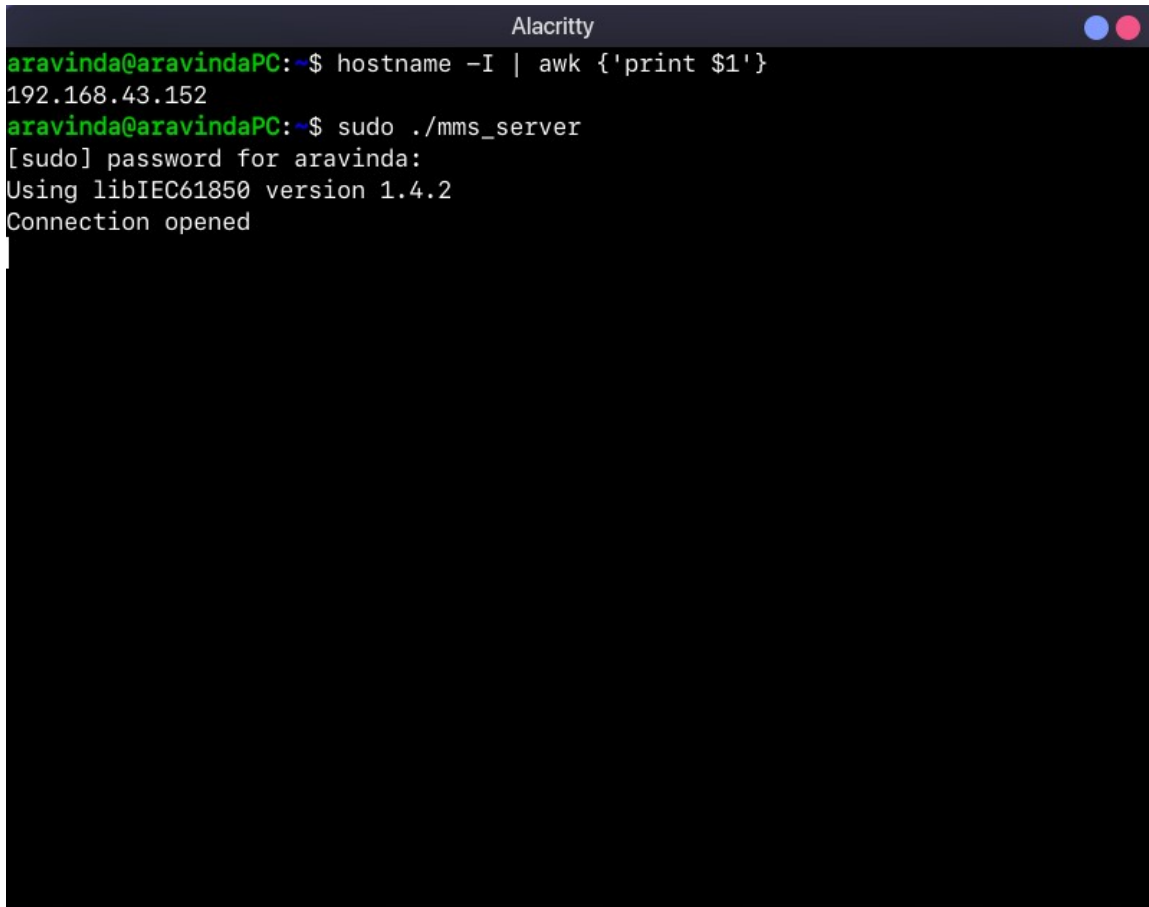


```

1  int
2  main(int argc, char** argv)
3  {
4      printf("Using libIEC61850 version %s\n", LibIEC61850_getVersionString());
5
6      /* Create new server configuration object */
7      IedServerConfig config = IedServerConfig_create();
8
9      /* Set buffer size for buffered report control blocks to 200000 bytes */
10     IedServerConfig_setReportBufferSize(config, 200000);
11
12     /* Set stack compliance to a specific edition of the standard (WARNING: data model has also to be checked for compliance) */
13     IedServerConfig_setEdition(config, IEC_61850_EDITION_2);
14
15     /* Set the base path for the MMS file services */
16     IedServerConfig_setFileServiceBasePath(config, "./vmd-filestore/");
17
18     /* disable MMS file service */
19     IedServerConfig_enableFileService(config, false);
20
21     /* enable dynamic data set service */
22     IedServerConfig_enableDynamicDataSetService(config, true);
23
24     /* disable log service */
25     IedServerConfig_enableLogService(config, false);
26
27     /* set maximum number of clients */
28     IedServerConfig_setMaxMmsConnections(config, 2);
29
30     /* Create a new IEC 61850 server instance */
31     IedServer = IedServer_createWithConfig(&IedModel, NULL, config);
32
33     /* configuration object is no longer required */
34     IedServerConfig_destroy(config);
35
36     /* set the identity values for MMS identify service */
37     IedServer_setServerIdentity(IedServer, "MZ", "basic io", "1.4.2");
38
39     /* Install handler for operate command */

```

Fig.5.1: The main routine of the server instance

A terminal window titled "Alacrity" with standard macOS window controls (red, yellow, green buttons). The terminal shows a user named "aravinda" at a machine named "aravindaPC". The user runs the command "hostname -I | awk {'print \$1'}", which outputs "192.168.43.152". Then, the user runs "sudo ./mms_server". The terminal shows the password prompt "[sudo] password for aravinda:", followed by the message "Using libIEC61850 version 1.4.2" and "Connection opened". The rest of the terminal area is black.

```
aravinda@aravindaPC:~$ hostname -I | awk {'print $1'}
192.168.43.152
aravinda@aravindaPC:~$ sudo ./mms_server
[sudo] password for aravinda:
Using libIEC61850 version 1.4.2
Connection opened
```

Fig.5.2: MMS Server started with IP 192.168.43.152

Once the server is running, it starts to listen to any client connections. OMicron software has a handy tool for discovering of IEDs by providing its IP address. This software perfectly replicates a Human Machine Interface found on various power substations. It can display us the Logical Devices and its hierarchy of Logical Nodes and the data it holds in real time. In our particular experiment we try to compute and send 4 oscillating sine values which replicates the AC measurements that is sampled on the grid and send it through the Logical Node GGIO1.

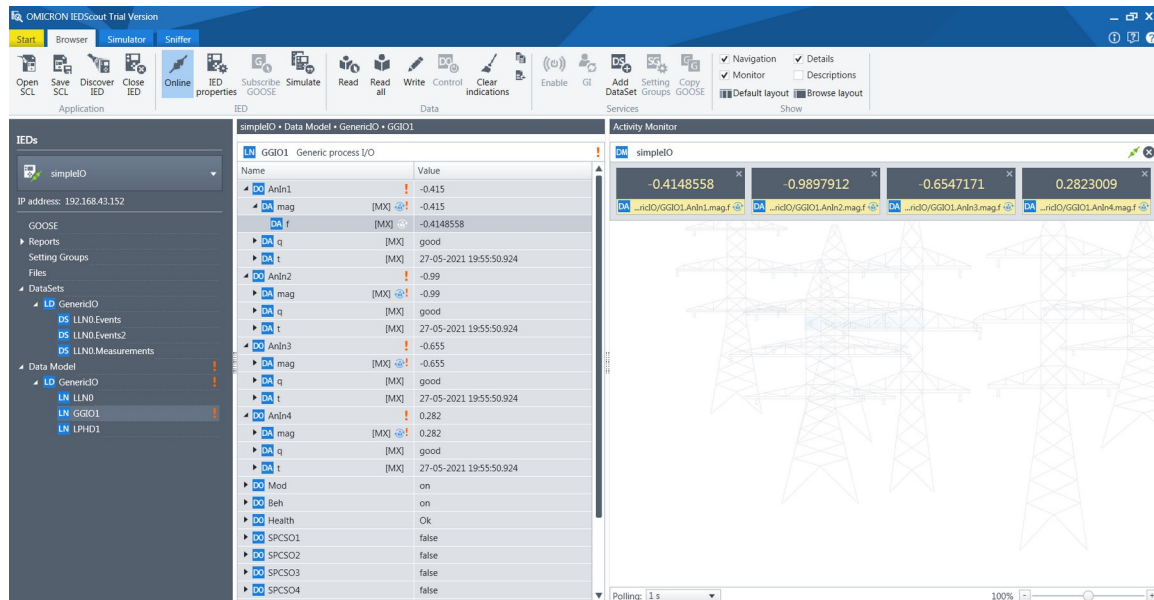


Fig.5.3: The Client side receiving real time data from the simulated server

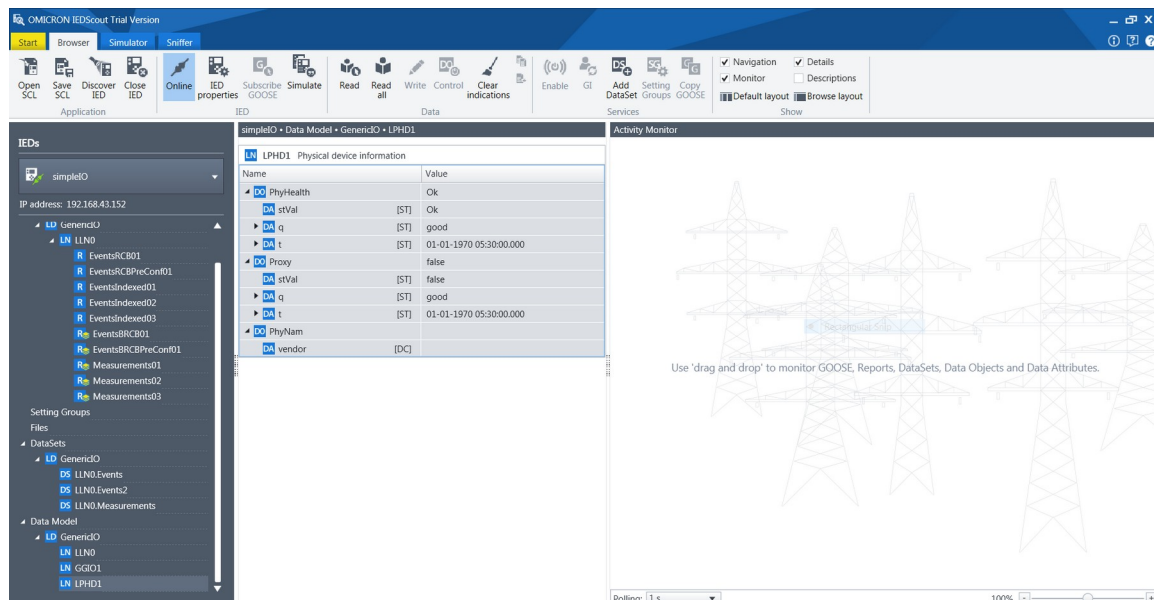


Fig.5.4: LPHD1 LN transferring vital information about the IED

As we can see the server created has successfully made contact with the client software and transfers data real-time at an interval of 100ms. Since the whole server is running off of a virtualization system, this can be ported across multiple hardware platforms. Also the API specification does not have an upper limit on the number connections that are made with the device. Hence this server software is scalable to any size required, only the appropriate ICD file is necessary without actually changing the server itself. The data is transferred as MMS packets over the network and this can be verified using the Wireshark Network Analyzer.



No.	Time	Source	Destination	Protocol	Length	Info
217	231.884100410	192.168.43.152	192.168.43.153	MMS	197	initiate-ResponsePDU
220	231.885159780	192.168.43.152	192.168.43.153	MMS	105	01 confirmed-ResponsePDU
230	231.887438538	192.168.43.152	192.168.43.153	MMS	124	02 confirmed-ResponsePDU
237	231.908545543	192.168.43.152	192.168.43.153	MMS	139	03 confirmed-ResponsePDU
242	231.932215327	192.168.43.152	192.168.43.153	MMS	784	04 confirmed-ResponsePDU
246	231.940528236	192.168.43.152	192.168.43.153	MMS	253	05 confirmed-ResponsePDU
248	231.945573855	192.168.43.152	192.168.43.153	MMS	86	06 confirmed-ResponsePDU
250	231.946263627	192.168.43.152	192.168.43.153	MMS	86	07 confirmed-ResponsePDU
251	231.946312174	192.168.43.152	192.168.43.153	MMS	132	08 confirmed-ResponsePDU
252	231.946389606	192.168.43.152	192.168.43.153	MMS	85	09 confirmed-ResponsePDU
255	231.947559668	192.168.43.152	192.168.43.153	MMS	284	10 confirmed-ResponsePDU simpleIO
257	231.948105789	192.168.43.152	192.168.43.153	MMS	260	11 confirmed-ResponsePDU simpleIO
258	231.948280747	192.168.43.152	192.168.43.153	MMS	458	12 confirmed-ResponsePDU simpleIO
261	231.956048779	192.168.43.152	192.168.43.153	MMS	963	13 confirmed-ResponsePDU
265	232.042683430	192.168.43.152	192.168.43.153	MMS	81	14 confirmed-ResponsePDU
268	232.061502153	192.168.43.152	192.168.43.153	MMS	610	15 confirmed-ResponsePDU 0.404849
271	232.105396751	192.168.43.152	192.168.43.153	MMS	131	16 confirmed-ResponsePDU

Fig.5.5: Wireshark capture of MMS packets

The destination (client) at IP 192.168.43.153 receives these packets. On dissection of the packets sent in the below picture, we can see that each packet has 4 items sent with a unique item ID in the format GGIO1\$ST\$SPCS01\$stVal which conforms to the naming convention specified by the IEC-61850 standard.

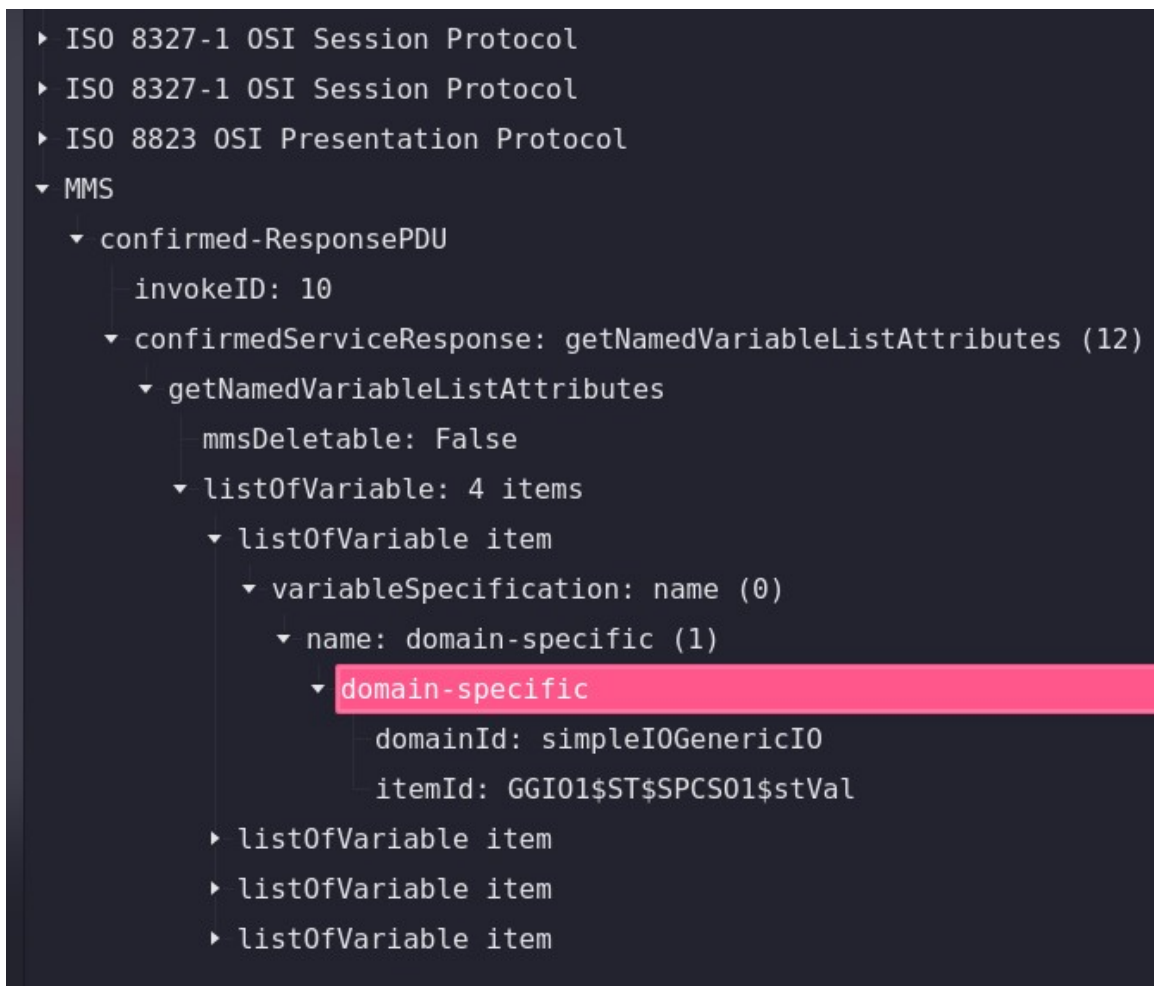


Fig.5.6: Packet dissection using the Wireshark

The communication between the process and bay level devices is done by running a SV Publisher/Subscriber service using the APIs provided by the libiec61850 framework on the virtual machine and simulating the IED using the OMicron software by loading the appropriate ICD file with appropriate Logical Node defined to receive GOOSE packets over the network. This simulates the communication done between the Merging Units and the IED on a power grid.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.43.153	234.5.6.7	UDP	82	4987 → 4988 Len=40
2	9.978175167	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
3	9.979298097	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	255	
4	10.980893212	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	255	
5	11.479618312	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
6	12.981136570	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
7	12.982806714	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	255	
8	13.183936121	HewlettP_18:a6:21	Iec-Tc57_01:00:01	GOOSE	202	
9	14.184208159	HewlettP_18:a6:21	Iec-Tc57_01:00:01	GOOSE	202	
10	14.482399477	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
11	15.184573012	HewlettP_18:a6:21	Iec-Tc57_01:00:01	GOOSE	202	
12	15.485096816	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	255	
13	15.983951756	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
14	16.184837302	HewlettP_18:a6:21	Iec-Tc57_01:00:01	GOOSE	205	
15	17.485862632	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
16	17.987199934	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	255	
17	18.987076773	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
18	20.488565440	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
19	20.489477618	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	255	
20	21.990147645	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	
21	22.991871661	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	255	
22	23.491584026	Dell_eb:1a:e6	Iec-Tc57_01:00:01	GOOSE	172	

Fig.5.7: GOOSE message capture using Wireshark

```

▶ 802.1Q Virtual LAN, PRI: 4, DEI: 0, ID: 1
▼ GOOSE
  APPID: 0x1000 (4096)
  Length: 154
  Reserved 1: 0x8000 (32768)
  Reserved 2: 0x0000 (0)
  ▼ goosePdu
    gocbRef: simpleIOGenericIO/LLN0$G0$gcbEvents
    timeAllowedtoLive: 3000
    dataSet: simpleIOGenericIO/LLN0$Events3
    goID: events
    t: May 29, 2021 09:17:18.118430316 UTC
    stNum: 1
    sqNum: 12
    test: True
    confRev: 2
    ndsCom: False
    numDataSetEntries: 8
    ▼ allData: 8 items
      ▶ Data: boolean (3)
      ▶ Data: bit-string (4)
      ▶ Data: boolean (3)
      ▶ Data: bit-string (4)
      ▶ Data: boolean (3)
      ▶ Data: bit-string (4)
      ▶ Data: boolean (3)
      ▶ Data: bit-string (4)

```

Fig.5.8: GOOSE Packet dissection

This dissection shows us the structure and the functional constraint of the values being transferred. In a power grid the various measurement units at the process level send their analog measurements to the Merging Units and there they are converted into equivalent digital values that are fit to be transmitted over the network. Now this whole communication has also been successfully simulated.

Chapter 6

Conclusion and References

This report provides us with comprehensive details and procedure of developing a fully functional Intelligent Electronic Device compliant to IEC-61850 substation automation standard using only open-source hardware and low-cost embedded hardware used for laboratory test beds. The methodology discussed can be scaled up to any level as the foundation of the software framework are designed to be scalable. With abundant security vulnerabilities created everyday aiming towards the network based systems, the power sector is no exception to it. India has faced multiple security attacks in the past from various infamous groups particularly at that power grids, which when disrupted can lead to catastrophic results. Frequent review and up-gradation of security must be done to avoid such outcomes. This project can greatly help in reducing the effort put into developing an IED device which can ease up the research process. The simulation results have successfully demonstrated the communication between the Station, Bay and process levels with appropriate communication protocols. The future scope of this project would be to add a level of encryption to the data transferred as any network analyzer could sniff the traffic. But such encryption must not hinder the response times as that would affect critical switching and breaking operations.

References

1. Falk, Hebert, "IEC 61850 Demystified", Norwood MA, Artech House, 2019.
2. R. E. Mackiewicz, "Overview of IEC 61850 and benefits," *2006 IEEE Power Engineering Society General Meeting*, 2006, pp. 8 pp.-, doi: 10.1109/PES.2006.1709546.
3. Campbell Roy H, "Understanding and Simulating the IEC 61850 Standard, Liang Yingyi" in IDEALS, 2008, UIUCDCS-R-2008-2967.
4. J. C. Tan, C. Zhang and Z. Q. Bo, "The importance of IEC 61850 interoperability testing," *2008 43rd International Universities Power Engineering Conference*, 2008, pp. 1-5, doi: 10.1109/UPEC.2008.4651594.
5. T. S. Ustun, A. Hadbah and A. Kalam, "Interoperability and interchangeability considerations in microgrids employing IEC61850 standard," *2013 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, 2013, pp. 1-5, doi: 10.1109/SEGE.2013.6707932.
6. S. M. S. Hussain, T. S. Ustun and A. Kalam, "A Review of IEC 62351 Security Mechanisms for IEC 61850 Message Exchanges," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5643-5654, Sept. 2020, doi: 10.1109/TII.2019.2956734.

7. A. Elgargouri and M. Elmusrati, "Analysis of Cyber-Attacks on IEC 61850 Networks," 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT), 2017, pp. 1-4, doi: 10.1109/ICAICT.2017.8686894.
8. R. A. G. Burbano, M. L. O. Gutierrez, J. A. Restrepo and F. G. Guerrero, "IED Design for a Small-Scale Microgrid Using IEC 61850," in *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 7113-7121, Nov.-Dec. 2019, doi: 10.1109/TIA.2019.2938734.
9. J. Ren and M. Kezunovic, "Modeling and simulation tools for teaching protective relaying design and application for the smart grid," 2010 Modern Electric Power Systems, 2010, pp. 1-6.
10. A. Hadbah, T. S. Ustun and A. Kalam, "Using IEDScout software for managing multivendor IEC61850 IEDs in substation automation systems," 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2014, pp. 67-72, doi: 10.1109/SmartGridComm.2014.7007624.