

UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

COMMUNICATION AND SYSTEMS ENGINEERING

Alexandros Giasis

**EVALUATION OF THE IEC 61850 COMMUNICATION
SOLUTIONS**

Master's thesis for the degree of Master of Science in Technology submitted for inspection, in Vaasa, 15th of December 2016.

Supervisor Mohammed Elmusrati

Instructor Mike Mekkanen

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincerest thanks to my supervisor, Professor Mohammed Elmusrati, for his support and guidance, in both my thesis and studies. His dedication and guidance during the studies are comparable to the parental dedication, always offering spiritual strength and encouragement.

My deepest thanks, respect and appreciation go to my instructor Mike Mekkanen, for his continuous help and guidance. His contribution via his knowledge, experience, patience and dedication was vital and gave me the spiritual strength and inspiration to continue and accomplish my thesis.

In addition, I would like to thank the Technobothnian's staff, Mr. Veli-Matti Eskonen, and Juha Miettinen for their laboratory help and supporting. Subsequently, I would like to thank my best friend Anas Shekhamis for his support as well as Ahmed Elgagouri for his initial help with my thesis.

Finally, I would like to thank my beloved parents for the lifelong affection, guidance, support, encouragement and education they offered me in order to be a responsible and useful person in society.

Alexandros Giasis

Vaasa, Finland, 13th of December 2016

TABLE of CONTENTS

ACKNOWLEDGEMENTS.....	2
LIST OF FIGURES	6
LIST OF TABLES.....	8
ABBREVIATIONS	9
ABSTRACT	12
1. INTRODUCTION	13
1.1. Background	13
1.2. Previous work	13
1.3. Thesis' motivation	14
1.4. Thesis' objective	14
1.5. Used methodology	15
2. BACKGROUND (Protocols Related to IEC 61850)	16
2.1. The OSI Model	16
2.2. Communication Protocols.....	18
2.2.1. TCP/IP and UDP Protocol.....	18
2.2.2. IP Protocol.....	19
2.2.3. Ethernet Protocol.....	20
2.2.4. IEEE 802.1Q supporting Virtual LANs	20
2.2.5. Tunneling and VPN.....	21
2.3. MMS (Manufacturer Message Specification) Standard	23
2.4. IEEE C37.118 Standard	24
2.5. Implementation of the IEC 61850 in Embedded Systems	24
3. INTRODUCTION to the IEC 61850 STANDARD	27
3.1. The IEC 61850 History	27
3.2. Objectives and Benefits	28
3.3. The Standard's Structure.....	29
3.4. Elaboration of the IEC 61850 Communication Architecture	33
3.4.1. The IEC 61850 modelling approach	33
3.4.2. Logical node and logical device concept	34

3.4.3.	Data object concept	35
3.4.4.	Logical allocation of functions and interfaces	36
3.4.5.	Mapping to real protocols	38
3.4.6.	The Substation Configuration Language (SCL).....	39
3.4.7.	The GOOSE message (Generic Object Oriented Substation Event). 41	
3.5.	Transfer Time & Round-Trip Time (RTT).....	41
3.6.	Ping Command and ICMP Protocol	44
3.7.	Message Types and Performance Classes.....	45
4.	INTRODUCTION to IEC/TR 61850-90-5: USE of IEC 61850 to TRANSMIT SYNCHROPHASOR INFORMATION ACCORDING to C37.118	48
4.1.	General	48
4.2.	Introduction to Synchrophasors	48
4.3.	Modelling Considerations	50
4.4.	Communication Requirements.....	51
4.5.	Security Model.....	53
4.5.1.	General	53
4.5.2.	Key management and cryptographic support.....	54
4.6.	Services	55
4.7.	Time Synchronization	57
4.8.	Synchrophasor Profile Mappings.....	57
4.8.1.	General	57
4.8.2.	A-Profile.....	58
4.8.3.	Session layer.....	58
4.8.4.	Tunneled payload	60
4.8.5.	KDC profile.....	62
4.9.	T-Profiles	64
4.10.	The Effects on the IEC 61850-5	66
5.	CONDUCTING THE PRACTICAL PART_1	67
5.1.	Introduction.....	67
5.2.	Description of the Practical Part_1	68
5.3.	Presentation and Configuration of the Devices in the Practical Part_1	69
5.3.1.	The Viola M2M Gateway	69

5.3.2.	The Viola Arctic 3G/LTE Gateway	72
5.3.3.	The D-Link Wireless N300 Multi-WAN Router	75
5.3.4.	The D-Link router configuration.....	76
5.3.5.	The Huawei E392 TDD-LTE USB Stick.....	78
5.4.	Analysis of the Technical Problems Faced in the Practical Part_1	79
6.	CONDUCTING THE PRACTICAL PART_2.....	81
6.1.	Introduction to Libiec61850-0.9.0.2 Library	81
6.1.1.	Building the library and the examples.....	83
6.1.2.	Analysis of the practical implementation.....	84
6.2.	Introduction to Hamachi	86
6.2.1.	Installation and configuration of Hamachi.....	88
6.2.2.	Analysis of the practical implementation.....	90
6.3.	Introduction to Beagleboard.org	91
6.3.1.	Configuration of the BeagleBone-Black	92
6.4.	Analysis of the Results.....	96
6.5.	Argumentation, Future Work & Optimization.....	103
7.	CONCLUSIONS	105

LIST OF FIGURES

Figure 1. The OSI and the TCP/IP architecture.....	17
Figure 2. The prehistory to IEC 61850.....	28
Figure 3. The modeling approach.....	33
Figure 4. The IEC61850 class model..	34
Figure 5. Logical device building block.....	35
Figure 6. Anatomy of a data class in IEC 61850-7-3.....	35
Figure 7. The anatomy of a data object name.....	36
Figure 8. Interface model of a substation automation system.....	37
Figure 9. Overview of functionality and profiles	39
Figure 10. Definition of the transfer time.....	42
Figure 11. Analysis of the round-trip time.....	43
Figure 12. Windows command prompt used for PING-ing.....	44
Figure 13. Representation of sinusoidal signal.....	48
Figure 14. Block diagram of an application including several PMUs and PDCs.....	49
Figure 15. Substation PDC model with legacy PMUs.....	50
Figure 16. Overview of the general service mappings..	57
Figure 17. IEC 61859-90-5 A-Profiles.....	58
Figure 18. The structure of IEC 61859-90-5 session protocol	59
Figure 19. IEEE 802.3 frame format for SV & GOOSE.....	61
Figure 20. Association of A-Profile to T-Profiles	64
Figure 21. The format of IP header.....	65
Figure 22. The DEMVE project, a section of the Technobothnia laboratory.....	67
Figure 23. The concept of the practical part_1.....	68
Figure 24. The M2M Gateway	69
Figure 25. The graphical user interface of the M2M gateway.....	70
Figure 26. Download of the client authentication certificate.....	71
Figure 27. The Arctic Gateway	72
Figure 28. The Arctic's Status screen.....	73
Figure 29. The Arctic's advanced network service/frequency.....	74
Figure 30. The Wireless D-Link Router.....	75

Figure 31. The setup of the DWR-116 internet connection.	76
Figure 32. The DWR-116 Internet configuration.	77
Figure 33. The DWR-116 advanced settings for port forwarding.	77
Figure 34. The HUAWEI E392 USB Stick.	78
Figure 35. Definition of the server's IP address in the client's .c file.	85
Figure 36. Terminal line, client_1 receiving reports from the server.	86
Figure 37. Pinging the address of the server from the client.	86
Figure 38. The LogMeIn administration web page.	88
Figure 39. The LogMeIn Hamachi client.	89
Figure 40. Pinging the Hamachi client.	90
Figure 41. Beaglebone-Black platform.	92
Figure 42. GOOSE round-trip times by the implementation of the libiec61850.	97
Figure 43. GOOSE round-trip times for PC-to-PC pinging via Hamachi.	98
Figure 44. GOOSE round-trip times for Beagle-to-Beagle pinging via 4G modem.	99
Figure 45. GOOSE round-trip times for Beagle-to-Beagle pinging via Hamachi.	101
Figure 46. Comparison plot of the three methods.	102

LIST OF TABLES

Table 1. The meaning of the interfaces.	38
Table 2. Synopsis of the performance requirements	47
Table 3. Summary of communication requirements.	52
Table 4. Performance classes to be added to part-5.	66

ABBREVIATIONS

ACSI	Abstract Communication Service Interface
API	Application Programming Interface
APDU	Application Protocol Data Unit
APN	Access Point Name
CID	Configured IED Description
GOOSE	Generic Object Oriented Substation Event
GPRS	General Packet Radio Service
GSSE	Generic Substation State Events
EDGE	Enhanced Data Rates for GSM Evolution
HMI	Human Machine Interface
HSPA	High Speed Packet Access
ICD	IED Capability Description
ICMP	Internet Control Message Protocol
IEC	International Electro-technical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISO	International Organization for Standardization
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
KDC	Key Distribution Center
L2TP	Layer 2 Tunneling Protocol

LAN	Local Area Network
LN	Logical Node
LD	Logical Device
LTE	Long Term Evolution
MMS	Manufacturing Message Specification
OSI	Open System Interconnection
PDC	Phasor Data Concentrator
ssPDC	Substation Phasor Data Concentrator
PDU	Protocol Data Unit
PMU	Phasor Measurement Unit
PMC	Phasor Data Concentrator
PPTP	Point-to-Point Tunneling Protocol
ROCOF	Rate of Change of Frequency
RSTP	Rapid Spanning Tree Protocol
RTT	Round Trip Time
SAS	Substation Automation System
SCL	Substation Configuration description Language
SCSM	Specific Communication Service Mapping
SNMP	Simple Network Management Protocol
SPDU	Session Protocol Data Unit
SV	Sample Values
TCI	TeleControl Interface
TCP	Transport Control Protocol

TR	Technical Report
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VPN	Virtual Private Network
VLAN	Virtual Local Network
XML	Extensible Markup Language
WAMPAC	Wide Area Monitoring, Protection, and Control
WAN	Wide Area Network
XML	Extensible Markup Language

UNIVERSITY OF VAASA**Faculty of Technology:**

Author:	Alexandros Giasis
Topic of the Thesis:	Evaluation of the IEC 61850 Communication Solutions
Supervisor:	Mohammed Elmusrati
Instructor:	Mike Mekkanen
Degree:	Master of Science in Technology
Department:	Department of Computer Science
Degree Programme:	Degree Programme in Telecommunications Engineering
Major of Subject:	Communications and Systems Engineering
Year of Entering the University:	2013
Year of Completing the Thesis:	2016

Pages: 109

ABSTRACT

Initially, when the IEC 61850 standard was prepared, it was intended to be used within the limits of a substation for information exchange between devices. In the course of time and due to the standard's advantages, its concepts are nowadays used as well in other application areas of the power utility system. The IEC 61850 is based to the maximum extent on other existing communication standards (IEC/IEEE/ISO/OSI), offering among others: visualization of the real applications through the ASCII interface, standardized messages to be exchanged (GOOSE, SV), one configuration language regardless of the device (IED) type/brand, and mapping to already implemented computing protocols (MMS, TCP/IP, Ethernet). The features mentioned above lead to cost reduction, reliability, and interoperability, making the IEC61850 the dominant standard for intra- and inter-substation communication.

The parts 90-1 and 90-5 of the IEC 61850 standard concern the application of the tunneling and routing method in order to extend the communication beyond the substation's limits. Although they establish the theoretical background, it can be mentioned a lack of information regarding real applications. So, the objective of this thesis was at first to establish the communication link which will allow the communication of devices belonging to different LANs and second, the acquiring of the round trip times from the exchanged messages. The experiments were conducted by a combination of software (Hamachi) and embedded platform (BeagleBone) pinging to each other first via tunneling and next via 4G mobile network. The acquired round-trip times were used to evaluate and compare the tunneling and the 4G routing method, estimating in parallel what are the perspectives of these methods to be used for inter-substation communication.

Keywords: IEC 61850, IEDs, Communication Protocols, GOOSE, LAN, Tunneling.

1. INTRODUCTION

1.1. Background

Initially, when the first edition of the IEC 61850 standard (*Communication Networks and Systems in Substations*) was prepared, it was intended to support the communications within the substation limits (Intra-substation), having as fundamental object the interoperable operation of devices/IEDs coming from different manufacturers.

In the meantime, due to the modern requirements of the power utility system, the scope of the IEC 61850 is no longer limited to substations but it has spread over a wide area of applications. Therefore, the IEC's Technical Committee 57 except for reviewing the old parts, is publishing also new in order to cover the demand for standardization of current or new areas of application; for example, the IEC/TR 61850-90-1 (suggesting tunneling for inter-substation communication), and the IEC/TR 61850-90-5 (providing routable profiles for the GOOSE and SV messages).

The term TR (*Technical Report*) implies that the 90-1 and 90-5 parts are not International Standards and their content is informative rather than normative. Although they define the theoretical background for the extension of the communication over wide area networks (WAN), in the real world the current IEDs do not have the inherited capability to support the routing or tunneling of Ethernet-based (GOOSE and SV) messages.

1.2. Previous work

Although a considerable amount of research has been conducted regarding IEC 61850 applications within the substation's LAN, there is a not extensive research regarding the performance of the tunneling and routing method for WAN applications extending the

SAS' limits. A few examples worth to be mentioned are: First, a publication in the PAC World Magazine (December 2010), *Performance Measurements for IEC 61850 IEDs and Systems*, (Steinhauser, Schossig, Klien, Geiger /Omicron Group). In the article is analyzed the round-trip time and several testing methods, without reference to inter-substation communication. Second, a publication in the Journal of Power and Energy Engineering (2015), *Conformance Test for IEDs Based on IEC 61850 Communication Protocol*, (Yeh, T.-H., Hsu, S.-C., Chung, C.-K., and Lin, M.-S); the article analyzes the Ping-Pong method for acquiring round-trip times, presenting in parallel a conformance test with real IEDs exchanging GOOSE messages within the LAN of the substation. Although the article provides a comprehensive analysis of the acquired results within the LAN, does not refer any extension of the method over wide areas networks!

1.3. Thesis' motivation

The motivation or challenge deriving from the above is that engineers or students pursuing to evaluate the performance of the methods introduced in the 90-1 and 90-5 parts, cannot solely be based on conventional devices/IEDs. On the contrary, the establishment of the communication link providing inter-substation connection has to be conducted via supplementary software and/or hardware simulating the communication aspects in a substation.

1.4. Thesis' objective

The fundamental objective of this thesis was to investigate the performance of the *tunneling* and *routing* methods by the exchange of GOOSE messages, extending communication from the substation limits to wide area network (WAN). Before reaching the final target, it was necessary to pass through some intermediate steps.

The challenge of this implementation except for establishing the communication link and exchange messages was also to find a method to measure the round-trip time of these messages since it is not defined within the IEC 61850 documentation. For sake of

simplicity, the round-trip time was assumed to be equal to the pinging-time which was acquired from the pinging of one device to another.

The comparison of the routing and tunneling methods was done, taking into consideration their drawbacks, the assumptions being made, and the acquired round-trip times, having as a reference point the time range provided by the Table_3. Although the acquired times were within the range suggested by Table_3, it was not possible to specify if this communication regards also the exchange of protection messages since they have the strictest time requirements.

1.5. Used methodology

The methodology being followed can be divided into the theoretical and practical part. The backbone of the theoretical part was the ten parts of the IEC 61850, plus the parts 90-1 and 90-5. It also included the DEMVE project tutorial, devices' manuals, network concepts, and IEEE/IEC publications and standards. The source of information regarding the measurement and analysis of the round-trip time was a publication of the Omicron Group in the PAC-World magazine.

The practical part initially required the familiarization with the IEDs, routers/ dongles, Linux language, and embedded platforms. Next, it followed the implementation of the tunneling method through the Hamachi software, and the routing method via the combination of the BeagleBone platform and the 4G-mobile modem.

2. BACKGROUND (Protocols Related to IEC 61850)

The communication protocols introduced in the current chapter consist part of the theoretical background of this thesis and are close-related to the communication aspects of the IEC 61850 standard. The exchange of IEC 61850 messages (GOOSE, SV) within/over the substation limits is based on already implemented communication protocols such as IEC, IEEE, ISO, & OSI, and therefore, a brief introduction of these protocols is considered vital for the readers to understand this thesis.

2.1. The OSI Model

According to *TECH-FAQ* (2016), the OSI model (*Open System Interconnection*) was created in 1980 by the International Organization for Standardization, and it was published in its current form in 1996. It consists a layered representation of the communication process, dividing the process into seven layers of functionality. The model specifies the functional requirements for each layer, without specifying or restricting the protocols to be used in order to achieve interoperability. The seven layers known also as a “*stack*” are:

- **Application Layer:** It is the top layer of the OSI model, consisting the interface that the user interacts with a particular application. For example, on a PC it provides the interface for the user to access the e-mail, firewall, browser, and so on.
- **Presentation Layer:** In consist part of the operating system, also referred as the “syntax” layer. Its main responsibility is to define the data syntax. In other words, it converts the data between application and network formats and vice versa. Some of the functions performed by the *Presentation* layer are data translation, data encryption/decryption, and protocol conversion.
- **Session Layer:** It is responsible for establishing, maintaining and terminating the communication “*session*” between two or more networked devices.

- **Transport Layer:** It guarantees data delivery between two or more networked devices. Some of the functions performed by this layer include reliability control of the communication link via flow control, fragmentation and reassembly, and error detection/recovery. The TCP and UDP protocols are used at this layer.
- **Network Layer:** The primary function of this layer is to perform routing, i.e. the establishment of the paths for data transmission among the nodes of a network. IP addressing is a function of this layer.
- **Data-Link Layer:** It is mainly responsible for setting up links over the physical network for data transmission, while it can correct errors created in the Physical Layer. It is divided into:
 - MAC (Media Access Control) layer: Its responsibility is to control the access of the communication medium from devices, preventing this way the data collision.
 - LLC (Logical Link Control) layer. It performs error checking, frame synchronization, and flow control
- **Physical Layer:** It is the 1st layer of the OSI model, dealing with the physical connection of devices. The electro-mechanical components (connectors, cables, etc.) belong to the physical layer and handle the transmission/receiving of raw data in the form of bits (0/1). Functions such as data encoding and multiplexing belong in this layer.

OSI Layer	TCP/IP Layer	TCP/IP Protocols	
7 Application	Application Layer		
6 Presentation		Telnet FTP SMTP DNS SNMP	
5 Session		NFS XDR RPC	
4 Transport	Transport Layer	TCP	UDP
3 Network	Internet Layer	RIP, OSPF, EGP IP, ICMP, ARP, RARP	
2 Data Link	Network Interface Layer	Ethernet, FDDI, Frame Relay, ATM, SLIP, PPP	
1 Physical			

Figure 1. The OSI and the TCP/IP architecture. (IT Wissen, 2016)

2.2. Communication Protocols

2.2.1. TCP/IP and UDP Protocol

The importance of the TCP and UDP protocols derives from the fact that they are used as the transport protocols for some types of messages within the substation (e.g. MMS, TimeSync). Additionally, the exchange of GOOSE and SV messages outside the substation will be held by routable-UDP.

According to *TechTarget* (2016), both TCP and UDP transport protocols are used in the Transport Layer of the OSI model.

- The TCP (*Transmission Control Protocol*) is a connection-orientation protocol, meaning that it establishes and maintains the connection until the applications of both ends have finished the data exchange. The TCP performs congestion control and error-free data transmission since it retransmits the lost packets. It offers better reliability regarding data transmission in comparison to the UDP, since the device/application sending the data, receives an acknowledgment of the successful receiving of data.

The TCP collaborates with the IP protocol and together rule the Internet. For example, the HTTP protocol is used to send files over the Internet, asking TCP to set up the connection. Next, the TCP divides the application file into packets and forwards them individually to the network/IP layer. Packets may be sent over different routes although they have similar source and destination IP addresses. The TCP protocol at the client's side performs assembling of packets into a file and asks the sender to retransmit any lost packets. The retransmission procedure may introduce latency into communication, so the TCP is not ideal for time-critical applications.

- According to the *TechTarget* (2016), the UDP (*User Datagram Protocol*) is a connectionless protocol that does not send acknowledgments regarding lost or successfully received data. So, it does not offer reliable data transmission. Same as

TCP, it runs on top of the Internet Protocol, and it is also referred as UDP/IP. Contrary to the IP, offers port numbering, helping to distinguish among different user requests.

Contrary to the TCP, the UDP sends the packets without performing congestion control and data retransmission; which leads to lower bandwidth requirements and latency. It consists ideal solution for network applications sensitive to latency such as gaming or VoIP.

Another interesting feature of the UDP is that it can be used in application vulnerable to data loss if the application is pre-configured to retransmit lost packets and re-assemble them to the correct order. In our case, this feature is crucial regarding the transmission of routable GOOSE since these messages are critical for the proper operation of SAS and are configured to be retransmitted in default time slots.

2.2.2. IP Protocol

The IP protocol enables the routing of packets and it consists an option for the transmission of data over arbitrarily long distances if the communication delays are within the limits set by the application.

According to the *TechTarget* (2016), the Internet Protocol (IP) belongs to the network layer of the OSI model. It is a two-layer program since it always collaborates with the TCP or UDP protocol. The IP receives the “*datagrams*” or “*packets*” from the TCP, providing them the IP address of the sender and receiver. IP forwards packets through intermediate nodes of the network till the final destination. The determination of the optimum path is known as “routing,” while packets of the same message may be routed over different paths. The communication model of the TCP/IP is based on the client/server model.

2.2.3. Ethernet Protocol

The importance of the Ethernet protocol derives from the fact that the exchanged GOOSE and SV messages use Ethernet frames for their transmission within the substation limits (*mapped onto the Ethernet data frames*).

According to the *TechTarget* (2016), the Ethernet belongs to the data-link layer of the OSI model and it is the most widely used local network technology, describing the format of data to be transmitted by network devices. It defines two units for the transmission of data, the packet, and the frame. Among others, the frame includes the “payload” of the transmitted data, addressing information about the physical-MAC address of sender and receiver, VLAN & priority tagging, information about the quality of service, and error-correction information. Each packet wraps a frame, affixing a number of bytes useful for the connection establishment and indication of the frame start.

The CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) is used to share the medium. Network devices detect if another device is transmitting at the moment, and if so (*collision detection*) will wait a short time before trying to retransmit.

The offered transmission speeds are the 100 BASE-T, providing speeds up to 100 Mbps, the Gigabit-Ethernet offering speeds of 1000 Mbps, and the 10-Gigabit Ethernet (GbE), offering up to 10 Gbps.

2.2.4. IEEE 802.1Q supporting Virtual LANs

According to *TechTarget* (2016), a local area network (LAN) consists of Ethernet switches, hubs, bridges and servers communicating with each other via the 2nd layer. Virtual-LANs (VLANs) are supported by the IEEE 802.1Q standard which defines the VLAN identifier by assigning a 4-byte ‘tag’ to the Ethernet packet. Three bytes are used to denote the VLAN-ID, while the other byte indicates the priority level of the packet.

A VLAN adopts or abstracts the concept of a LAN, and it consists of a number of ports assigned to a switch or a number of ports assigned to many switches, allowing the division of systems into logical subnets. Each subnet obeys to different rules regarding its communication. As a default, devices belonging to one Virtual-LAN cannot directly interact with devices belonging on other Virtual-LANs of the same network.

Some of the features provided by VLANs are extra security, network segmentation, service separation/isolation and simplified administration. The network administrator is allowed virtually divide his network to fulfill the functional and security requirements of his systems without running new cables or making significant changes to the current network.

Additionally, another interesting feature of VLANs is that they can be tunneled through Layer 3, allowing systems located in different physical locations to communicate as if they were located physically on the same LAN.

The IEC 61850-90-1 (2010: 52-54), highlights the issues to be taken into consideration when using the Ethernet protocol. VLAN (IEEE 802.1q) technology is suggested to restrict the access to the network or to allow only the authenticated partners. Ethernet switches supporting Virtual-LANs can be configured regarding which VLAN to accept on each port. For example, ports assigned for protection, will not allow another type of packets to pass through, decreasing this way the network traffic. Additionally, the priority tagging separates the critical protection messages (high priority), from other low priority traffic.

2.2.5. Tunneling and VPN

According to *Tech-FAQ* (2016), “*Tunneling*” or a “*tunneling protocol*” allows the transmission of data intended to be used within a private or local network through a public network. To achieve that, all the data to be transferred must be fragmented into smaller packets or frames and then forwarded through the tunnel. Each frame is

encrypted with an extra layer of tunneling encryption and encapsulated before routed to the right destination, where it is de-capsulated. After encapsulation, the nodes of the public network are unaware that substantially the transmission is part of the local network. In other words, Internet (public network) can be used to transfer data on behalf of a private network.

The encryption of the original frame prevents the interpretation of the content. Tunneling is also known as the encapsulation and transmission of VPN (*Virtual Private Network*) data, where the TCP/IP protocol provides the transport mechanism for VPN connectivity.

Some of the VPN tunnel types operating over the 2nd and 3rd layer of the OSI model are:

- The L2TP (*Layer 2 Tunneling Protocol*), and the PPTP (*Point-to-Point Tunneling Protocol*) are VPN protocols running in the data-link layer.
- The IPSec (IP Secure), can operate as a VPN protocol at the Network layer.
- The OpenVPN Technologies is a private company offering the open source OpenVPN software which operates over the 2nd or the 3rd OSI layer. It uses the industrial SSL protocol to support encryption and client authentication based on username/password credentials, certificates or smart cards. OpenVPN does not require a web browser for its operation and allows multiple clients to connect to an OpenVPN server via a single TCP or UDP port. The software is available for Windows, Mac, Android, and iOS systems. To use the service, a user has to download and install the OpenVPN program. *OpenVPN Technologies* (2016).

In our case, the “*tunneling*” is important since it is suggested as a method for substation-to-substation communication, (IEC 61850-90-1, 2010: 56-57). The 90-1 part considers two methods to achieve that, the tunneling, and the gateway approach.

The “*tunneling*” method allows the connection of multiple substation networks and the “direct access” to functions in remote substations. The tunnel does not care about the content of the transferred messages. Hence it does not need to be reconfigured if the information changes. Regarding the IEC 61850, the kinds of traffic are TCP/IP for C/S

communication and multicast Ethernet messages (GOOSE and SV). Tunneling method will be only applied if sufficient bandwidth is available. In the case of GOOSE transmission a higher bandwidth may be required, just to achieve low enough latency.

2.3. MMS (Manufacturer Message Specification) Standard

The interoperability of devices coming from different manufacturers was initially pursued through the development of the MMS standard (the 1980s) and later on through the of the IEC 61850 standard. The MMS is an international standard first published in 1988 and concerns automation systems in general. The IEC 61850 is highly influenced from the MMS standard since it implements MMS to execute some of its essential functions such as vertical communication, (DEMVE Training Material, 2014: 10).

According to the *TechTarget* (2016), the MMS was developed to optimize the automation of the industrial process. The standard was revised to include the communication among computers, intelligent devices, and systems of all kinds. The data is handling regard real-time process and supervisory control data among network devices. The main advantage of the MMS is that it is vendor-independent offering interoperability among devices of different vendors, and also it is independent of the application's function it has to execute.

MMS protocol and services are designed to operate over compliant to OSI and TCP communications profiles. The 8th part of IEC 61850 is dedicated to the mapping of ACSI to MMS, providing detailed instructions for the mechanism and rules required to implement the objects and services of the ACSI by the MMS concepts, objects, and services. The purpose of mapping is to offer interoperability across functions implemented by different vendors. The MMS is mostly used to transfer operational data of medium priority.

2.4. IEEE C37.118 Standard

The IEEE C37.118 was the ancestor of the IEC 61850-90-5 part, focusing on synchrophasors measurements without standardizing a communication profile for the exchange of packets. According to *IEEE C37.118 Standard for Synchrophasors Data Transfer for Power Systems* (2005), the standard was divided into two distinctive parts:

- IEEE C37.118-1: The first part emphasizes to measurements only, defining synchrophasors, frequency, and *ROCOF* (*Rate of Change of Frequency*). It specifies requirements regarding time-tag and synchronization, introducing in parallel evaluating methods regarding these measurements. Additionally, it defines a *PMU* (*Phasor Measurement Unit*), as a stand-alone unit or a functional unit collaborating with a physical device. The standard does not specify hardware, software, and computing methods for phasors, frequency, and ROCOF.
- IEEE C37.118-2: The 2nd part focuses on the communication part, defining a method for synchrophasor data exchange between power system equipment. It specifies the types, formats, use and contents of messages for real-time communication among PMUs, PDCs, and other applications.

Since the IEEE C37.118-2 was suffering from some problems and they were requests to enhance the standard, the IEC 61850 was chosen to provide the enhanced functionality. A migration strategy had to be applied in order to provide discrete and manageable steps for users and vendors desiring to use IEC 61850. The strategy has as starting, and end points the IEEE C37.118 and the IEC/TR 61850-90-5 respectively.

2.5. Implementation of the IEC 61850 in Embedded Systems

According to *Tech-FAQ* (2016), when we talk about embedded systems we refer to a combination of built-in computer hardware and software specially designed to execute a particular task. Some embedded systems are fixed while some others are programmable

and provide a programming interface. Thousands of modern devices such as industrial machines, ATMs, household appliances, vehicles, mobiles, medical instruments, etc. are hosts of embedded systems.

Some of the embedded systems' characteristics are:

- Theoretically, they are designed to handle a few simple tasks, although the procedure for accomplishing that task may be complicated as a computer program.
- Originally embedded systems had no user interface. The necessary data and programs were already incorporated, and no human interaction was required except for installing the device. Nowadays, many embedded systems provide a full-scale user interface, e.g. keyboards to enter numbers or names, etc.
- Originally they were simple, and they had limited functionality (switches, digital displays, LEDs), indicating the 'health' of the embedded system. Nowadays they have achieved a level of complexity (e.g. ATMs).
- CPU Platforms with microprocessors or microcontrollers are also considered embedded systems since in a sense the BIOS chip executes limited functions during the computer's boot up.
- Several operating systems or languages have been developed for embedded systems, such embedded Java or Linux.

In the case of the IEC 61850 standard, the embedded system applications provide remarkable advantages and make them a very useful tool for developing or testing an application. Platforms incorporating microprocessors or microcontrollers offer some key benefits explained below:

- An engineer or student can familiarize with the standard's concepts and applications without the need to buy a real IED, or to access a real laboratory/substation.
- Their cost is considerably lower in comparison to real devices!
- Their small size makes them portable and easy to be conveyed; they can be placed even in small offices without requiring new infrastructure, except for a PC and access to Internet.

- The combination of hardware/software allows the engineer or programmer to write, execute, or modify the application's code and test its performance before running it on real devices.
- These platforms also offer extensibility allowing the user to add extra devices or accessories according to the application's requirements.
- Except for the hardware, they exist libraries especially designed to run on embedded systems or even PCs, (in our case the libIEC61850-0.9.0.2). This library is provided for free and is developed to run IEC 61850 applications. Among others, it offers ready server/client IEC 61850 communication examples for familiarization, while users are allowed to develop their own applications.

An example of IEC 61850 embedded applications is the BeagleBone-Black being used in the practical part of this thesis (Figure 41).

Except for developing platforms, IEDs incorporate as well embedded modules. For example, an IED incorporating an embedded switch module is capable of performing actions of managed switches, if for example supports the RSTP protocol (for the fast re-healing of a failed network), or the SNMP protocol (regarding monitoring and management of network devices). Instead of being standalone modules, the embedded virtual switches within an IED have as result in the elimination of communication infrastructure, and finally the cost reduction, (Taikina-aho, 2011: 42-43).

3. INTRODUCTION to the IEC 61850 STANDARD

The IEC 61850 – *Communication Networks and Systems in Substations* is an international standard published by the IEC (International Electrotechnical Commission). Its initial objective was to support the communication within the substation, defining the interoperable communication among IEDs coming from different manufacturers. IEDs (*Intelligent Electronic Devices*) are capable of performing the functions of a SAS (*Substation Automation System*) regarding protection, control, and monitoring.

The standard supports interoperability mainly via four components: 1) Use of one **configuration language (SCL)** for the configuration of IEDs regardless of their type and brand. 2) Organization of data by a comprehensive standard **data-model**, allowing different types/brands of IEDs to exchange information since they use a common data structure. 3) Utilization of the **ACSI interface** which provides a number of standardized actions such as "write"/ "read". 4) **Mapping** of data model and commands over protocols already implemented in the power industry, TCP, UDP, Ethernet, etc.

3.1. The IEC 61850 History

According to Pinto Faria (2011: 5), in 1988 IEEE and EPRI initiated the UCA (*Utility Communications Architecture*) project in collaboration with private companies from the USA. The scope of the project was to pursue future interoperability among control systems being used for monitoring and control of the electric grid. The result was the emerging of UCA 1.0 (Standard for Communications Architecture).

Version 1.0 of UCA had some limitations, restricting the adoption of UCA architecture in the electric power utilities. Nevertheless, IEEE and EPRI continued their efforts to

improve the UCA architecture by running a number of research projects such as the MMS Forum Working Groups. The result of these efforts was the UCA version 2.0. In 1994, the working group “*Substation Control and Protection Interfaces*” of IEC Technical Committee 57 proposed a standard for communication in SAS (*Substation Automation Systems*). The standard for the informative interface of protection equipment has been published as IEC 60870-5-103. (Pinto Faria 2011: 5).



Figure 2. The prehistory to IEC 61850 (Gunter A., Zhangand 2009: 6).

Finally, in 1997 IEEE, EPRI and Working Group 10 (WG10) of the IEC Technical Committee 57 (TC57) collaborated to establish a common international standard for electrical utility communications. Their efforts were based on the UCA architecture and led to the emerging of the IEC 61850 standard. (Pinto Faria 2011: 5).

3.2. Objectives and Benefits

According to IEC 61850-1 (2003:12), the objectives of the IEC 61850 can be summarized as follows:

- To satisfy the functional and performance communication requirements of a SAS, while supporting the future evolution of substation automation.
- To make use of existing IEC/IEEE/ISO/OSI communication standards, to the maximum possible extent.
- To provide interoperability between IEDs supplied by different manufacturers, resulting in a simpler substation structure by enabling the integration of all control, protection, monitoring, and measurement functions by one common protocol.
- To support the functions of the substation according to the operational requirements. Nevertheless, the standard's purpose is neither to limit in any manner the functions involved in the operation of the SAS nor to restrict their free allocation within the substation.
- To offer a real object oriented approach for substations, supporting standardized device models and standardized configuration language (SCL).

According to *Siemens Efficient Energy Automation with the IEC 61850 Standard* (2010: 2), the main advantages of the standard are:

- **Simple substation structure:** The IEC61850 is developed in cooperation with manufacturers and users to provide a uniform interface, avoiding this way protocol diversity and integration problems.
- **Simplicity:** Simpler engineering, implementation, operation, and services. Save time and costs on design, configuration, commissioning, and maintenance.
- **Cost reduction:** The implementation of IEC 61850 standard means a lower cost regarding engineering, commissioning, operation, and maintenance.
- **More reliability:** Use of one communication channel for all data incorporating real-time synchronization via Ethernet.

3.3. The Standard's Structure

The standard mainly consists of 10 parts. Nevertheless, working Group 10 of Technical Committee 57 continues to review old parts or publishes new, to cover the demand for

standardization of current or new areas of application. For example, the IEC 61850-90-5 providing routable profiles for the GOOSE messages, IEC61850-70-410 related to hydroelectric power plant communication, and so on. The ten main parts introduced in IEC 61850-1 consist of:

IEC 61850-1

Introduction and Overview: The general overview of the standard is introduced in the 1st part, including the history, philosophy, and working approach to the standard. Part-1 provides as well a brief introduction of the other ten parts.

IEC 61850-2

Glossary: The 2nd part is dedicated to the introduction of specific terms and definitions used within a SAS and applied to all parts of the IEC 61850 series.

IEC 61850-3

General Requirements: The 3rd part specifies the general requirements of the communication network, emphasizing on the quality requirements. These requirements refer to reliability, maintainability, availability, security, data integrity, environmental conditions and auxiliary services.

IEC 61850-4

System and Project Management: Part-4 introduces the engineering requirements, consisting of the engineering process (parameter classification, engineering tools, and documentation), the overall system/IEDs life cycle and the quality assurance (equipment test, type tests, and system tests).

IEC 61850-5

Communication Requirements for Functions and Device Models: The scope of part 5 is to standardize the communication between IEDs and system requirements. The communication requirements are identified through the identification of all known functions and logical nodes. Additionally, are introduced the concepts of a logical node, function, PICOM, performance and dynamic scenarios.

IEC 61850-6

Configuration Description Language for Communication in Electrical Substations related to IEDs: In the part-6 is introduced the Substation Configuration Description Language (*SCL*), which is used to describe IED configuration and communication. SCL enables the interoperable exchange of communication system configuration data between system configuration tool and IED configuration tool from different manufacturers. Additionally, in part-6 is specified the file format to describe the communication related to IED configuration and IED capabilities.

IEC 61850-7-1

Basic Communication Structure for Substation and Feeder Equipment – Principles and models: In part 7_1 is introduced an overview of the interactions among SAS devices and the communication architecture. Additionally are introduced the modeling methods, information methods, and communication principles being used in IEC 61850-7-x. Last, there is a description of the relationships between the other parts of the standard.

IEC 61850-7-2

Basic Communication Structure for Substation and Feeder Equipment – Abstract Communication Service Interface (ACSI): Part 7_2 defines the ACSI interface for use in substations where there the real-time cooperation of IEDs is required. The ACSI is independent of the underlying communication systems and is defined with regards to hierarchical class models, services associated with these classes, and parameters related to each service.

IEC 61850-7-3

Basic Communication Structure for Substation and Feeder Equipment – Common Data Classes: This part is an amendment to the 1st edition of IEC 61850-7-3 standard, defining in parallel new common data classes that are used in power quality models and statistical/historical data representation.

IEC 61850-7-4

Basic Communication Structure for Substation and Feeder Equipment – Compatible Logical Node Classes and Data Classes: This part consists an addition to a set of specifications, regarding the SAS communication architecture. Additionally, it defines the models for power quality functions and standardizes logical nodes and data objects.

IEC 61850-8-1

Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO/IEC 9506-1 and ISO/IEC 9506-2) and to ISO/IEC 8802-3: The 8th part of the standard introduces a method for exchanging time-critical and non-time-critical data via LAN networks by mapping ACSI to MMS and ISO/IEC 8802-3 frames. MMS services and protocol are specified to operate over OSI and TCP communication profiles. Some of the protocol stacks used within the standard are routable; hence, the actual communication is not restricted solely to LANs. The content of exchanged data includes real-time control and monitoring, measured values and notification.

IEC 61850-9

Specific Communication Service Mapping (SCSM): The 9_1st part of the standard applies to electronic current (ECT) and electronic voltage transformers (EVT) or other bay devices such as protection relays. It specifies the mapping for communication services between the bay and process level as well as the mapping for the transmission of sampled values (as defined in IEC 61850-7-2) on a serial unidirectional multi-drop point-to-point link in accordance with IEC 60044-8.

IEC 61850-10

Conformance Testing: The last part of the standard introduces a set of specific techniques for testing the conformance of applications, as well as measurement techniques to be applied for the declaration of performance parameters in accordance to IEC 61850-5 requirements. The implementation of these techniques aims to facilitate the system integrator for the easy integration of IEDs, their correct operation and supporting of applications as intended.

3.4. Elaboration of the IEC 61850 Communication Architecture

3.4.1. The IEC 61850 modelling approach

According to the IEC61850-7_1 (2003: 15), the information models and modelling methods are the backbone of the IEC61850. The standard is modelling the common information found in real applications.

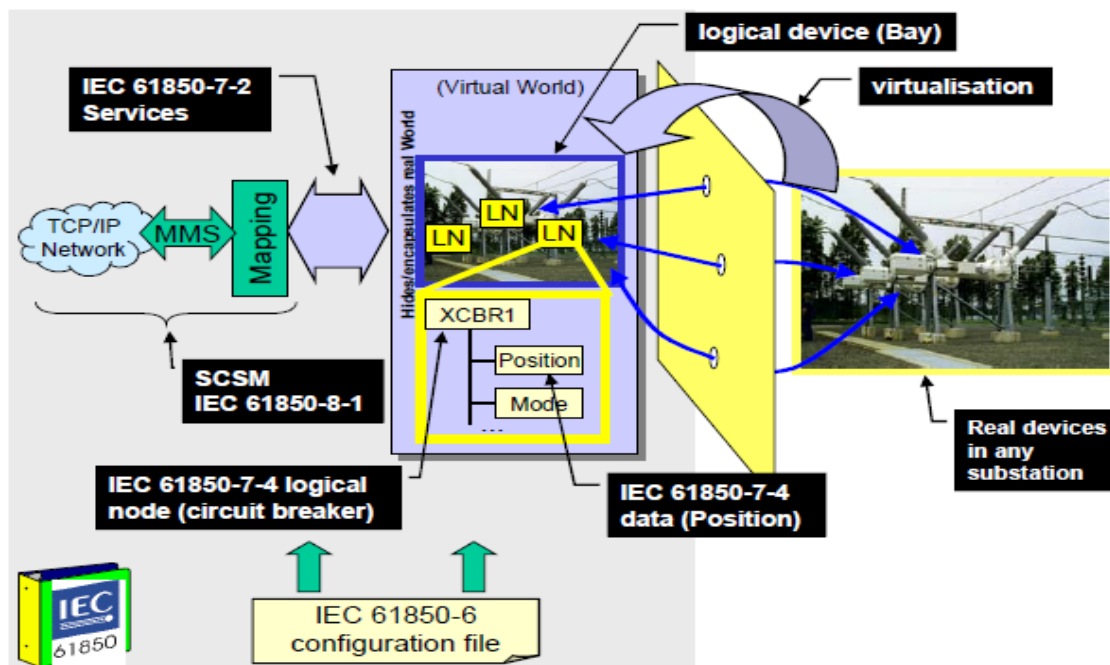


Figure 3. The modeling approach (IEC 61850-7_1 2003: 15).

The concept of virtualization is achieved through the ACSI (*Abstract Communication Service Interface*), which provides a virtual image of the real device and the data it contains. The virtual interface of an IED, provides information regarding its logical nodes (LN), logical devices (LD), data attributes, and communication services (e.g. *control*, *get data values*), independently from the concrete application and communication protocol in use. In figure 3, real devices on the right are modeled as virtual on the left and accessed through the ACSI services (IEC61850-7-1, 2003:15-16).

3.4.2. Logical node and logical device concept

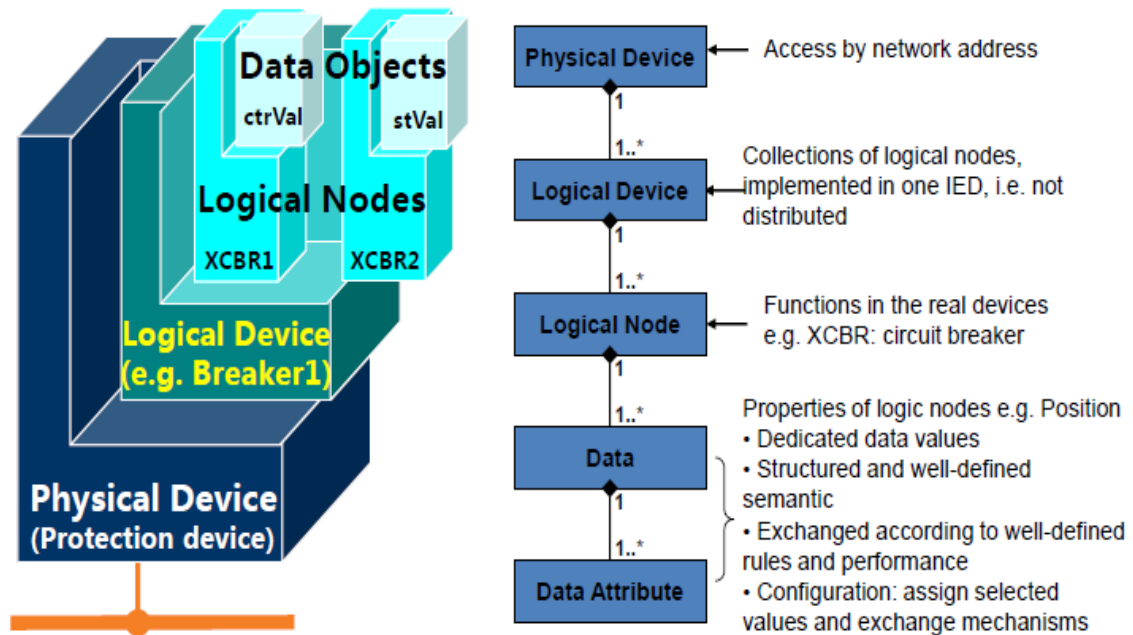


Figure 4. The IEC61850 class model. (Gunter A., Zhang and 2009: 15).

According to IEC61850-7-1 (2003:15-16), the standard decomposes the application functions into the smallest entities, capable of exchanging information (figure 4). These entities are called logical nodes (LNs), corresponding to well-known functions and are allocated to one or more physical devices. Additionally, logical nodes are categorized according to their role, for example concerning protection, measurements, and so on. Each logical node is represented with a standardized name, for instance, the *XCBR* represents a circuit breaker. The concept of all the logical nodes is defined and modelled in the IEC 61850-5. A group of logical nodes builds a logical device (LD), e.g., a bay unit. A logical device is always executed in one IED; hence, a LD is not distributed. Most of the functions consist of a minimum of three logical nodes, and about 90 logical nodes are defined. The logical nodes and data objects contained in a logical device are vital for the description and data exchange within a substation in order to achieve interoperability.

Additionally, the concept of the logical device has been introduced for communication purposes. A logical device except logical nodes is also composed of additional services

such as a GOOSE or SV exchange (figure 5). Logical nodes are grouped in logical devices, according to their common features. A logical device provides information as well about the physical device is hosting it, such the *nameplate* and *health*. (IEC61850-7-1, 2003:47).

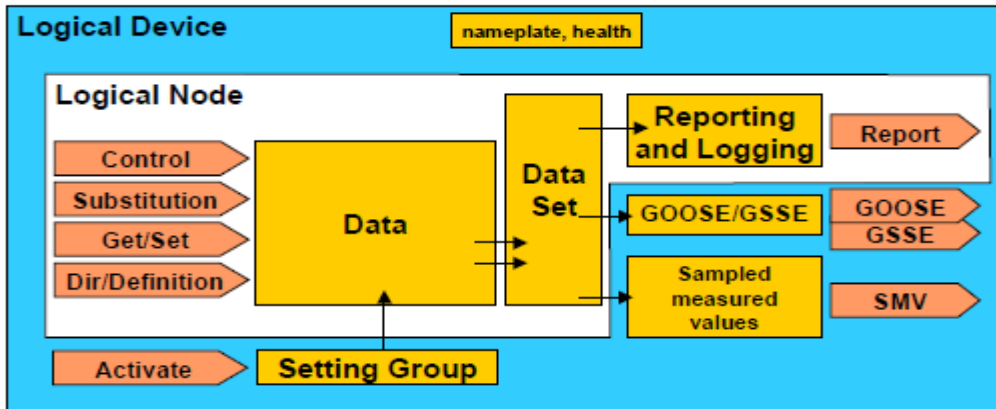


Figure 5. Logical device building block (IEC 61850-7-1 2003: 47).

3.4.3. Data object concept

According to Adamiak, Baignet, and Mackiewicz (2004: 63-64), every logical node contains data objects, representing specific information, for example, a measurement, a position or a status. A group of objects sharing the same services, attributes, relationships and semantics creates a data class. Each data class has a set of attributes with a defined name, type, and specific purpose (figures 6 & 7).

Class					
ATTRIBUTE NAME	ATTRIBUTE TYPE	FUNCTIONAL CONSTRAINT	TRGOP	VALUE / VALUE RANGE	MANDATORY/ OPTIONAL
DataName	Inherited from Data Class (see IEC 61850-7-2)				
DATA ATTRIBUTE					
Status					
stVal	BOOLEN	ST	dchg	TRUE FALSE	Mandatory
q	Quality	ST	qchg		Mandatory
t	TimeStamp	ST			Mandatory

Figure 6. Anatomy of a data class in IEC 61850-7-3. (Adamiak, Baignet, Mackiewicz, 2004: 63).

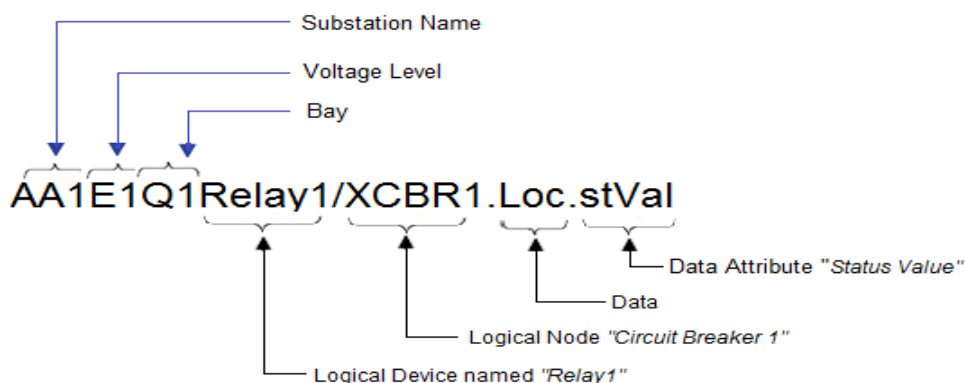


Figure 7. The anatomy of a data object name.

The above picture depicts a logical device named “relay1”, belonging to a particular substation, bay and voltage level. The logical device consists of a circuit breaker logical node (XCBR1). By reading the data object in Figure 7, we can determine if the breaker controlled by the AA1E1Q1Relay1 device is in local or remote operation.

3.4.4. Logical allocation of functions and interfaces

According to the IEC 61850-1 (2003:12), the functions within a SAS environment correspond to tasks that are executed in the substation. Functions exchange data with other functions, and they are performed by IEDs. All considered functions consist of LNs, regarding the protection, control, and monitoring of the substation equipment. Also, there are functions related to SAS maintenance, i.e. for communication management or software management.

The allocation of functions to devices (IEDs) is not fixed; hence, the standard has to support the free allocation of functions. To achieve that, the interoperability criterion has to be satisfied for functions resided to equipment provided by different suppliers. The allocation depends on performance and availability requirements, cost restrictions, the state of the art of technology, user’s philosophy, etc.

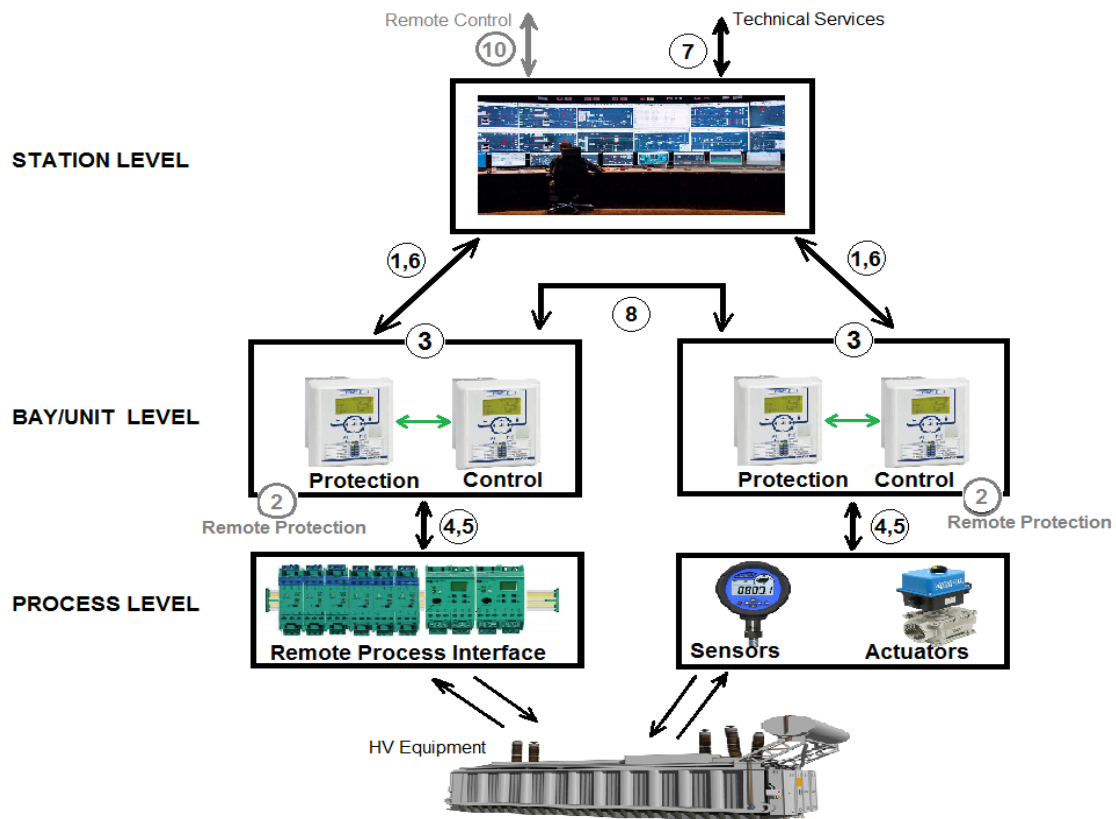


Figure 8. Interface model of a substation automation system.

As is shown in Figure 8, the functions of a SAS may be logically allocated on three different levels: station, bay, and process level. (IEC 61850-5 2003:14).

- a) *Process level functions*: They are functions related to the process, and they communicate to the bay level via the logical interfaces 4 and 5.
- b) *Bay level functions*: They are functions related mainly to the primary equipment of a bay and use mainly the data of one bay. They communicate via the logical interface-3 within the bay and via the logical interfaces 4 and 5 to the process level.
- c) *Station level functions*, divided into:
 - The *process related station-level functions* that use the data of more than one bays or the complete substation and act on the primary equipment, communicating mainly via the logical interface 8.
 - *Interface related station level functions*, representing the SAS interface to the local station operator HMI, to a remote control center *TCI*, or to the remote

engineering for maintenance and monitoring and they communicate with the bay level via the logical interfaces 1 and 6.

Table 1. The meaning of the interfaces. (IEC 61850-5 2003: 15).

IF1	Exchange of protection data between bay and station level
IF2	Exchange of protection data between bay level and the remote protection
IF3	Bay level data exchange
IF4	Exchange of process – bay level data (instantaneous current/voltage Transformer data/samples)
IF5	Exchange of control data between process and bay level
IF6	Exchange of control data between station and bay level
IF7	Data exchange between station and a remote workstation
IF8	Bay-to-bay data exchange (especially regarding fast functions such as interlocking)
IF9	Station level data exchange
IF10	Exchange of control data between SAS (devices) and a remote controlling facility

3.4.5. Mapping to real protocols

The IEC61850-8-1 maps the abstract objects and services to the MMS (*Manufacturing Message Specification*) protocol. The MMS was chosen because it is the only public (ISO standard) protocol having a proven application track record that can easily support the complicated naming and service models of the IEC 61850. Theoretically, the IEC 61850 can be mapped to any protocol, but this can be very complicated regarding the mapping of objects and services to a protocol providing only read/write/report services. So, the choice of MMS is ideal because it supports complex named objects and a wide set of services, supporting the mapping to IEC 61850 in a simple way. (Adamiak, Baignet, and Mackiewicz, 2004:64).

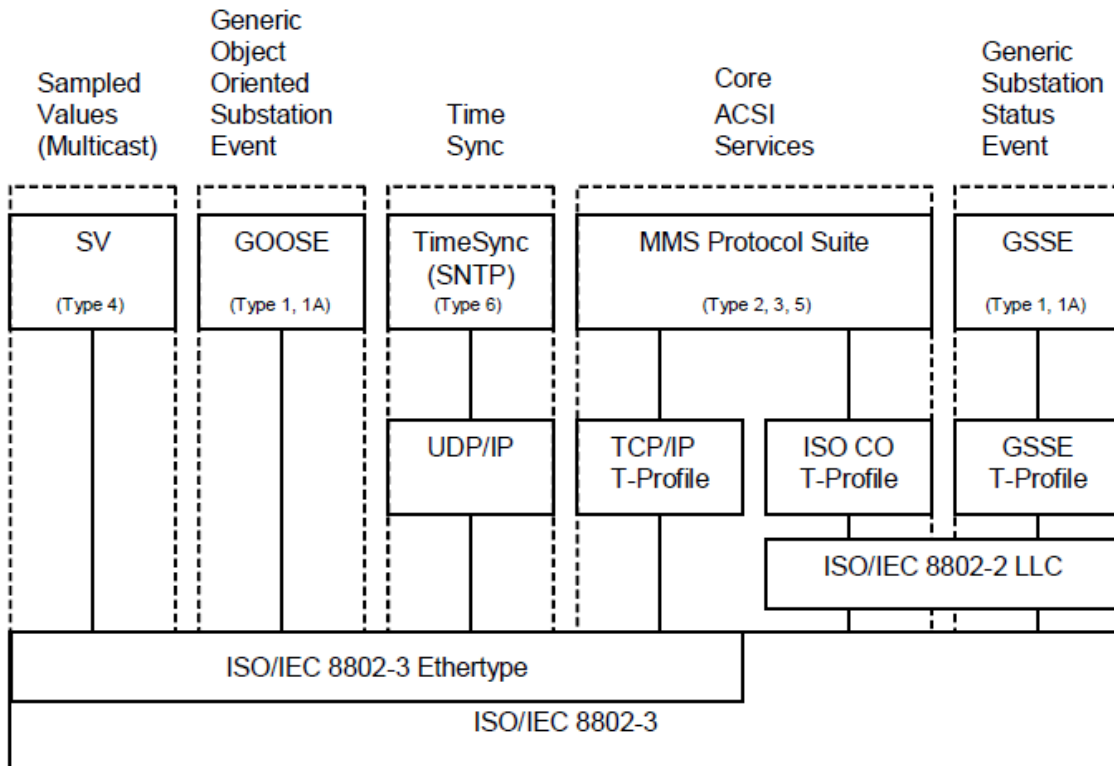


Figure 9. Overview of functionality and profiles (IEC 61850-8-1, 2004: 19).

Except for the mapping to the application layer, in the part 8-1 are defined profiles for the other layers of the communication stack, depending on the service provided (figure 9). The GOOSE and Sampled Values go through the application and presentation layer and then are mapped directly onto the Ethernet data frame, eliminating this way the processing delays of any middle layers. The MMS can operate over TCP/IP or ISO, while the GSSE (*Generic Substation Status Event*) a similar to the GOOSE application operates over connectionless ISO services. For mapping to an Ethernet data frame all data use either the data type “Ethertype” in the case of GOOSE, SV, TimeSync, and TCP/IP or the data type “802.3” in the case of GSSE messages.

3.4.6. The Substation Configuration Language (SCL)

According to the 6th part of the IEC 61850 (2004: 7-19), the SCL (*Substation Configuration Language*) is a description language for the configuration of substation

IEDs, and it is based on the Extensible Markup Language (*XML*). It is used to describe IED configuration and communication systems according to IEC 61850-5 and 7.

The SCL specifies a file format to describe the communication related to IED configuration, IED parameters, configuration of system communication, switchyard (function) structures, and the relations among them. It uses four different file types, each aiming to contain a different piece of the substation configuration. The six SCL files are introduced below:

- The **.ICD** file, (*IED Capability Description*): This file contains exactly one IED section describing the capabilities of an IED. It is used for data exchange between the IED and the system configuration tool.
- The **.IID** file, (*Instantiated IED Description*): It specifies the configuration parts of an IED which have been standardized by the IEC 61850. (DEMVE Training Material, 2014: 170).
- The **.SSD** file, (*System Specification Description*): In this file, is described the single line diagram of the SAS and the required LNs. It is used for data exchange between a system specification and system configuration tools.
- The **.SCD** file, (*Substation Configuration Description*): This file contains all the IEDs, a section of communication configuration and a section of substation description. It is used for data exchange between the system configuration and IED configuration tools.
- The **.CID** file (*Configured IED Description*), consisting of two sections; the communication section containing the current address of the IED, and the optional substation section referred to this IED. It is used for data exchange between the IED configuration tool and the IED.
- The **.SED** file (*System Exchange Description*): The **.SED** file is introduced in the 2nd edition of the IEC 61850-6, and it is used to exchange system interface descriptions among all the communicating systems (PAC World, 2015).

3.4.7. The GOOSE message (Generic Object Oriented Substation Event)

According to the IEC 61850-2 (2003: 11), when any change of state occurs, IED multicasts a high speed, binary object, GOOSE report, typically containing the double command state of each of its status inputs, outputs, and relays, actual and virtual. These reports are re-issued sequentially at intervals of 2, 4, 8...60000 ms. A GOOSE message is a high-speed trip signal that has a high probability of delivery.

The GOOSE is a layer_2 multicast message, originally intended to be used within the SAS limits (1st edition of the IEC 61850). It is used for horizontal communication within the SAS, meaning that IEDs on the same hierarchy (the same bay or voltage level) communicate with each other. The IEC 61850-8-1 part defines the GOOSE frame structure, including the packet sequence number, TTL (time to live for the packet), time tag, source identification, revision number of the current configuration, etc. (DEMVE Training Material, 2014: 34-44).

Data can be multicast using the GOOSE protocol so that they can be received by any number of receivers. The receiving IEDs are called subscribers. The GOOSE protocol usually is used to transfer values which change relatively rarely in contrast with the SV protocol which is used for real-time transfer of measured values such as current or voltage.

3.5. Transfer Time & Round-Trip Time (RTT)

Ideally the transmission of data/packets is done at the speed of light, but in reality, several factors are exceeding the time required for a packet to reach its destination, meaning that they introduce an unwanted delay (*latency*) in the transmission. Regarding networks these factors include: the rate of data transmission, the transmission medium (air, copper, optical fiber), the distance between the sender and receiver, the number of nodes the packet passes till to reach its destination, the traffic on the network, the

priority of the packet, the processing speed of sender and receiver, and possible interference of the medium (TechTarget, 2016).

The transfer time as specified in the IEC 61850-5 (2003: 45), refers to the complete transmission of a message, including the processing delay at both ends. The transfer time starts counting from the moment the sender puts the data on top of its transmission stack till the moment the data are extracted from the transmission stack of the receiver

Figure 10 indicates a function (f_1) in physical device PD1, sending data to a function (f_2), in physical device PD2. The transfer time regards the whole transmission procedure, being a sum of the individual times of the communication processors plus the transfer time required by the network, (including proceeding times by routers or other devices belonging to the network).

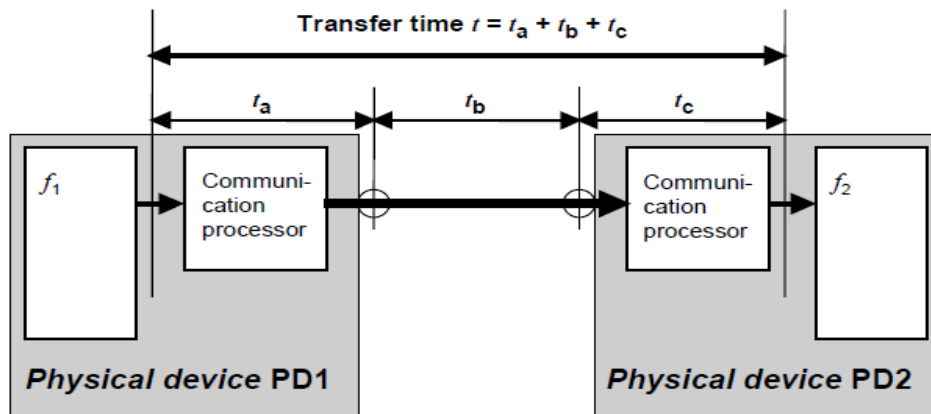


Figure 10. Definition of the transfer time (IEC 61850-5 2003: 45).

In our case, the exchange of GOOSE messages is based on a publisher/subscriber mechanism, where the sender writes the values in a local buffer (sender side) and the receiver accessed the values from a local buffer at the receiver's side (IEC 61850-7-2, 2003: 108). Table 2 on page 47 provides a summary of the message types and their transfer time requirements.

Although the IEC 61850 defines and analyzes the transfer time, it does not describe the round-trip time; hence, we will based on a publication of the *PAC World Magazine* to provide an analysis of the round-trip time.

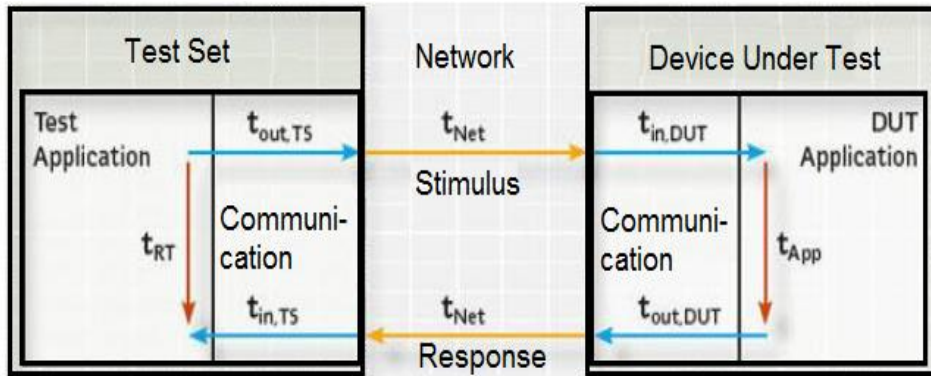


Figure 11. Analysis of the round-trip time, (PAC World, 2011).

According to PAC World (2011), in order to test the round-trip time, a stimulus is sent to a DUT (*Device under Test*), while the DUT tries to respond as fast as possible. So, the round-trip time (t_{RT}) refers to the time interval between sending the stimulating signal and receiving the response.

Figure 11 has derived from the figure 10, and it is adopted for round-trip tests scenarios.

Combining the two figures, the relation of times is:

- $t_a = t_{out,TS} + t_{out,DUT}$
- $t_c = t_{in,DUT} + t_{in,TS}$
- $t_b = t_{Net}$ (time required by the network to deliver the packet)

From the above derives that the round-trip time is the sum of seven intermediate times:

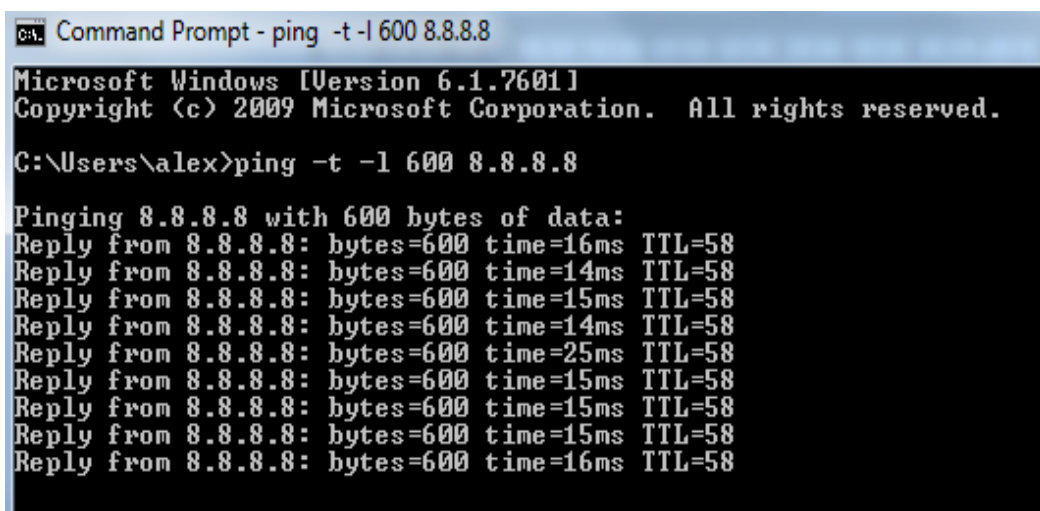
$$t_{RT} = t_{out,TS} + t_{Net} + t_{in,DUT} + t_{App} + t_{out,DUT} + t_{Net} + t_{in,TS}$$

3.6. Ping Command and ICMP Protocol

The ICMP (*Internet Control Message Protocol*) allows routers or other devices to send error or control messages to other routers or devices (Comer E. 2000: 130). The ICMP protocol provides communication between the IP software on one machine and the IP software on another.

The ICMP *echo request* and *echo reply* message, also known as PING is one of the most widely used debugging tools provided by TCP/IP protocol. A computer or router sends an ICMP echo request to a specific destination, and the machine receiving the echo generates an echo reply and forwards it back to the original sender (Comer E. 2000: 133-134).

In our experiment for the sake of simplicity, it was assumed that the PING time is equal to the round trip time, which is not exactly precise since the PING software utility does not perform packet processing. As denoted above the ICMP protocol is used for diagnostic/debugging purposes, and it is not an actual communication protocol. Additionally, the ping software utility was chosen because the transferred messages can “travel” through the nodes of the internet (WAN); that was impossible to be achieved by GOOSE messages since they can only be used within a LAN.



```
Command Prompt - ping -t -l 600 8.8.8.8
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\alex>ping -t -l 600 8.8.8.8

Pinging 8.8.8.8 with 600 bytes of data:
Reply from 8.8.8.8: bytes=600 time=16ms TTL=58
Reply from 8.8.8.8: bytes=600 time=14ms TTL=58
Reply from 8.8.8.8: bytes=600 time=15ms TTL=58
Reply from 8.8.8.8: bytes=600 time=14ms TTL=58
Reply from 8.8.8.8: bytes=600 time=25ms TTL=58
Reply from 8.8.8.8: bytes=600 time=15ms TTL=58
Reply from 8.8.8.8: bytes=600 time=15ms TTL=58
Reply from 8.8.8.8: bytes=600 time=15ms TTL=58
Reply from 8.8.8.8: bytes=600 time=16ms TTL=58
```

Figure 12. Windows command prompt used for PING-ing.

Figure 12 depicts the Windows' command prompt being used in our experiment for pinging a specific address and acquiring of the PING time. Among others, the interface allows the adjustment of the packet size, the number of ICMP echo to be sent and provides statistics about the successfully received and the lost packets.

3.7. Message Types and Performance Classes

According to IEC 61850-5 (2003: 45-49), the communication link within a SAS is required to transfer a variety of messages, differing in the content, length, security and the allowed worst case transfer time. The requirements for the message types are divided into two independent performance classes; one for control and protection, including three performance classes (P1, P2, P3) and another one for metering and power quality applications, including the classes (M1, M2, M3),

The IEC 61850-5 (2003: 45-49) divides the message types into seven categories:

- Type 1 – “*Fast messages*”: Typically containing a small binary code containing data or commands, e.g. “Trip,” “Open,” “Close,” “Start,” “Block,” “Unblock,” “Stop,” “Trigger,” “Release.” The receiving IED should act immediately by the related function on the reception of this message type.
 - Type 1A “*Trip*”: The trip is the most important (fast) message in the substation. Hence it has more demanding requirements in comparison to the other fast messages. For Performance Class P1, the total transfer time shall be 10 ms, while for P2 or P3, the total transfer time shall be 3 ms.
 - Type 1B “*Others*”: The other fast messages also have an important role, but they have less demanding requirements in comparison to the trip. For Performance Class P1, the total transfer time shall be \leq to 100 ms, while for the P2 and P3, the total transfer time shall be 20 ms.

- Type 2 – “*Medium speed messages*”: For this type of message, the time at which the message was originated is the most important, while the transmission time is less critical. IEDs should have their own clocks, and the sender should attach a time-tag to the message. The transmission time should be less than 100 ms.
- Type 3 – “*Low-speed messages*”: Type 3 includes complex messages that may or may not require time-tagging. This message type should be used for low-speed transmission of event records, auto-control functions or reading/changing set-point values. The total transfer time shall be less than 500 ms.
- Type 4 – “*Raw data messages*”: This type includes the output from digital transducers and digital instrument transformers. These messages consist of continuous streams of synchronized data from each IED.
- Type 5 – “*File transfer functions*”: This message type is used for the transmission of large data files related to recording, settings, etc. Data must be divided into blocks of limited length, allowing this way other network activities. The transfer time is not critical; hence, there are not specific limits. A typical time value could be equal or greater than 1000 ms.
- Type 6 – “*Time synchronization messages*”: This message type is used for synchronization of the internal clocks of the IEDs in the SAS.
- Type 7 – “*Command messages with access control*”: This message type is used to transfer control commands, generated from local or remote HMI functions, with a higher degree of security. Type 7 is based on Type 3 message, with the addition of password or other verification procedures.

Table 2 provides a summary of the above-analyzed message types and performance classes. Obviously, the highest demand regards fast-trip messages (GOOSE), as well as instruments data output (SV).

Table 2. Synopsis of the performance requirements

Type	Application	Performance Class	Requirements (Transmission Time)
1A	Fast messages (Trip)	P1	10 ms
		P2/P3	3ms
1B	Fast messages (Other)	P1	100 ms
		P2/P3	20 ms
2	Medium Speed		100 ms
3	Low Speed		500 ms
4	Raw Data	P1	10 ms
		P2/P3	3 ms
5	File Transfer		≥ 1000 ms
6	Time Synchronization		(Accuracy)
7	Control Commands		

4. INTRODUCTION to IEC/TR 61850-90-5: USE of IEC 61850 to TRANSMIT SYNCHROPHASOR INFORMATION ACCORDING to C37.118

4.1. General

The IEC/TR 61850-90-5 (2012) focuses on the exchange of synchrophasor data between PMUs (*Phasor Measurement Unit*), PDCs (*Phasor Data Concentrator*), and control center applications. Apart from the primary scope, the 90-5 part provides as well routable profiles for GOOSE and SV messages. The routable packets can be used to transmit synchrophasor or common IEC 61850 data. The use of the IP is an option for the transmission of data over arbitrarily large distances, which extends the limits of a typical local area network.

4.2. Introduction to Synchrophasors

According to the *Phasor-RTDMS* (2015), a phasor is a sinusoidal signal represented by the magnitude and phase (with respect to a reference). The signal's magnitude represents the amplitude of the signal, and the phase is the distance between the signal's sinusoidal peak and a chosen reference. For example a reference to a fixed point in time (e.g. $t = 0$).

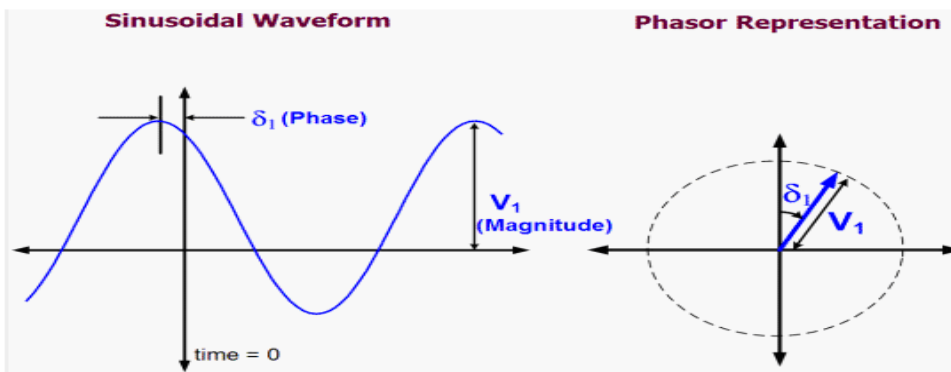


Figure 13. Representation of sinusoidal signal (Phasor-RTDMS).

Phasors (voltage and current) are used in many protection and data acquisition functions. Their use is increased further by matching them to a common time base, which can be achieved by synchronizing the phasor to a precise time. Phasors synchronized to a common time source and with reference to a common nominal frequency are defined as *synchrophasors*. Synchrophasors provide a good method for tracking the power systems, dynamic phenomena, improving this way the power system operation, monitoring, protection, and control. (IEEE C37.118.2, 2011: 6).

According to the IEC/TR 61859-90-5 (2012: 36-38), the smallest phasor network consists of two nodes; one PMU (*Phasor Measurement Unit*) and a PDC (*Phasor Data Concentrator*), communicating with each other. Usually, more than one PMU located at the same or different substations is connected to a PDC providing real-time data. Additionally, several PDCs located in various utilities can be connected to a ssPDC central PDC.

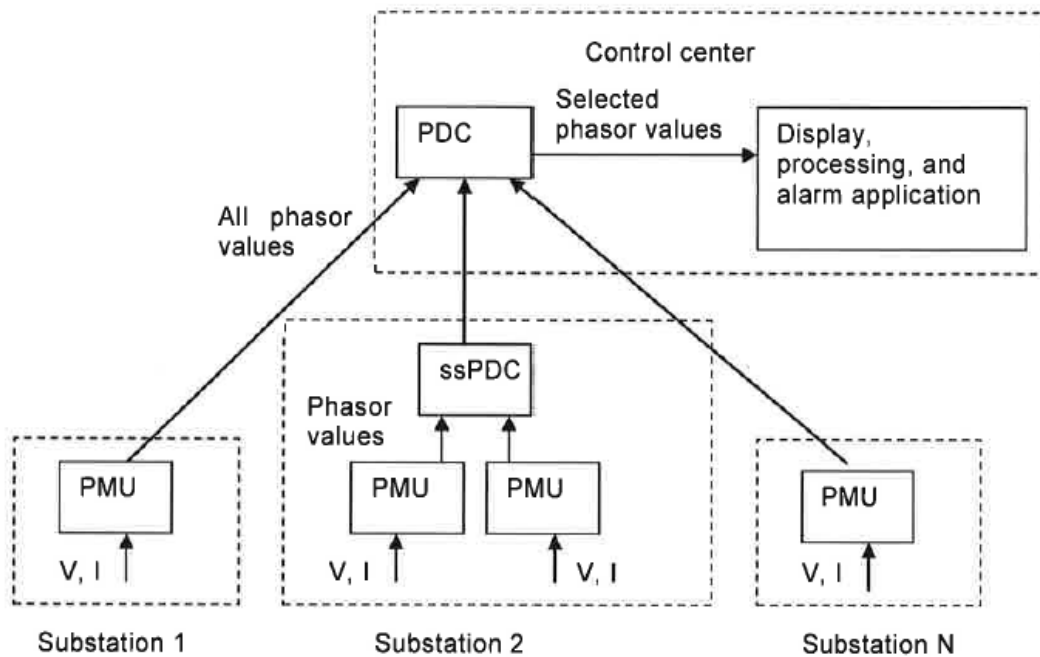


Figure 14. Block diagram of an application including several PMUs and PDCs. (IEC/TR 61859-90-5, 2012: 21).

The role of the PMU and PDC devices is to:

- *PMU*: Samples an analog voltage or current waveform in synchronization with a GPS-clock, calculating in parallel synchrophasor values, such as frequency and ROCOF (*Rate of Change of Frequency*). The samples from the different waveforms are assigned a time-tag to provide a common reference for the synchrophasor calculations obtained from various locations. The output of a PMU is a repetitive stream of data in IEC 61850-90-5 or IEEE C37.118.2 format.
- *PDC*: Receives phasor data streams from one or more PMUs or PDCs. Selects, validates, aggregates, decimates, and interpolates data from multiple PMUs or PDCs. Optionally converts data to and from IEC 61850-90-5, according to communications requirements, while in parallel secures data transmission, and calculates derived data values.

4.3. Modelling Considerations

According to the IEC/TR 61859-90-5 (2012: 41-44), for the description of a system in IEC 61850, the client and server need to be modeled as logical nodes on an IED. The use of some existing logical nodes is possible but more beneficial might be the introduction of application-specific logical nodes. For applications related to wide area control and frequency stability, the conventional methods might be too slow, so it is necessary the use of UDP-based GOOSE.

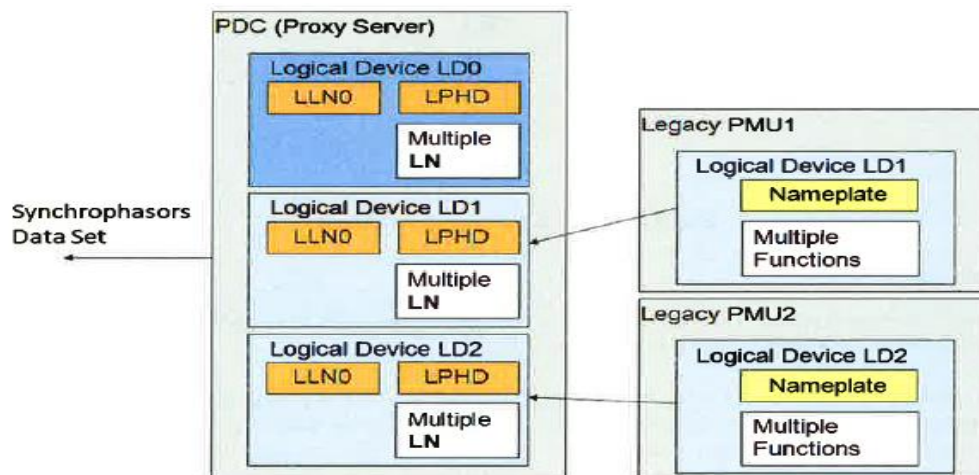


Figure 15. Substation PDC model with legacy PMUs. (IEC/TR 61859-90-5, 2012: 44).

Regarding the PMU model, the PMU function within an IED is responsible for the calculation and publication of synchrophasor measurements as specified in IEEE C37.118.1. These calculations are based on the sampling of an analog input within the IED containing the PMU function. The PDC function receives data from multiple PMUs within the substation, sorting them according to their time-tag.

4.4. Communication Requirements

According to IEC/TR 61850-90-5 (2012:46-50), the communication mechanism planned through the 90-5 edition aims to serve the needs for WAMPAC (*Wide Area Monitoring, Protection, and Control*) applications using synchrophasors measurements according to IEEE C.37.118. The communication within the substation will be based on SVs or GOOSE messages. Communication beyond the substation limits will be achieved either by *tunneling* (see 2.2.5 sub-chapter) the SV messages via a high-speed network like SDH or SONET, or via IP networks if their latency is within the acceptable limits. For IP communication the standard has to be enhanced by a mapping of SV and GOOSE onto an IP-based protocol. The UDP transport protocol has been chosen for the streaming of synchrophasor data, due to the periodic nature of these services and its successful operation in IEEE C.37.118 applications. The routable SV and GOOSE will be called R-SV and R-GOOSE respectively.

Synchrophasor communication requirements depend on the application they serve and need to be assessed individually for each case. These requirements consist of the rate of measurements' transmission, the delay, the variation in delay (jitter) and the reliability of delivery.

The following table summarizes the communication requirements outlined in the 3.7 subchapter. The three first columns derive from an individual case implementation, while qualitative requirements (jitter, lost packets) indicate the values to be expected from the operation of the application.

Table 3. Summary of communication requirements. (IEC/TR 61850-90-5 2012: 50).

Factor	Reporting rate range	End-to-end latency	Measurement timing error	Sensitivity to transmission jitter	Sensitivity to lost packets	Currently covered in 61850
Sync-check	$\geq 4/s$	100 ms	50 μs	Medium	High	SV service
Adaptive relaying	$\geq 10/s$	50 ms	50 μs	Low	Medium	SV service
Out-of-step protection	$\geq 10/s$	50-500 ms	50 μs	Medium	Medium	SV service
Situational awareness	1/s to 50/s	5 sec.	50 μs	Low to medium	Low to medium	Periodic reporting SV service
State-estimation & security assessment	1/300 s to 10/s	5 sec.	50 μs	Low	Medium	Periodic reporting SV service
Data archiving	Any	N/A	50 μs	Low	Medium	All as needed
Wide area controls	$\geq 10/s$	50-500 ms	50 μs	Medium	High	SV service
Predictive dynamic stability maintaining system	$\geq 25/s$ or 30/s	50 ms	50 μs	Medium	High	SV service
Under voltage load shedding	$\geq 25/s$ or 30/s	100 ms	50 μs	Low	High	SV service
Phenomenon assumption type WAMPAC (PMU- PDC)	1/s - 10/s	5 s	50 μs	Low to medium	Low to Medium	Periodic reporting SV service
Phenomenon assumption type WAMPAC (PMU- IED)	50/s or 60/s	20 ms	50 μs	Medium	High	SV service

As can be seen, the SV-service can be applied theoretically in all applications, except for wide area communication (Internet), since it is mapped directly onto Ethernet. In this case, the routable-SV can be used, keeping in mind that the R-SV does not cover the strict requirements regarding sampled values for classical protection.

The most important thing deriving from this table is the end-to-end latency requirement having a range of 50-500 ms. This requirement concerns wide area communications, and

roughly speaking it can be translated to a round-trip time around 100-1000 ms! Although this range concerns an individual case, it consists a good example to compare the acquired round-trip times from the practical part of this thesis.

4.5. Security Model

4.5.1. General

According to IEC/TR 61850-90-5 (2012: 51-52), the security model is based on the security threads and foresees issues regarding information authentication & integrity, and confidentiality (optional).

Theoretically, it should be provided an end to end method for data authentication and integrity, regardless of the data hierarchy. A typical way to provide the security function is through some message authentication code. The 90-5 part specifies symmetric and asymmetric key authentication, as well as symmetric key encryption for confidentiality.

The mapped SV are used to carry two types of information: CT/PT information per IEC 61850-7-2 and synchrophasor measurements per IEEE C37.118.1. These two types of data have different messaging rates. The traffic classes can be distinguished based on the type of information, messaging rates and security requirements.

- *Class A (Intra-substation)*: Intra-substation traffic is characterized by the exchange of high-resolution waveform information within the substation. It should support both synchrophasor and CT/PT values and for that is required a lightweight mechanism for authentication and tamper detection.
- *Class B (Inter-substation)*: Inter-substation traffic is characterized by the exchange of medium resolution waveform information between different substations. This type of traffic allows as well synchrophasor values to be transferred apart of the substation limits.

- For this type of traffic, except for authentication and tamper detection, optional encryption functionality is specified to provide confidentiality.

IEC 61850-9-2 SV and IEC 61850-8-1 publishers and subscribers do not provide a routing capability for the transmitted packets (*Class A traffic*). Nevertheless, this technical report specifies a mechanism for routing these packets and forwarding them as *Class B traffic*.

4.5.2. Key management and cryptographic support

According to IEC/TR 61850-90-5 (2012: 54-56), Key management and cryptographic support is designed to provide the following functionality:

- Synchronphasor data must be exchanged in a continuous fashion, even if the digital keys being used for encryption and signatures are changed.
- Synchronphasor data is delivered by connectionless protocols/services, operating typically in a multicast environment.
- Support of symmetric and asymmetric cryptography.

The Key Distribution Center (*KDC*) is used to support multicast transmission and symmetric key coordination between publisher-subscribers. KDC can be a standalone node, or it can be an IED function. For the second scenario, the benefit is that the IED device determines when to issue the next key, so a mechanism informs subscribers of an upcoming key change and allows them enough time to obtain the new key, without interruption information exchange. To achieve this, the 90-5 part needs to provide a warning mechanism for an upcoming key change. This mechanism is provided by the *TimetoNextKey* attribute.

The symmetric keys used for signature or encryption should be changed periodically by the publisher. Configuration should allow the setting of minimum and maximum *TimeToNextKey* values. The aspects regarding the key distribution center (*KDC*) are:

- KDC must be able to authenticate KDC clients on per data stream basis.
- There are several types of data streams within IEC 61850 domain. Such as R-GOOSE, R-SV, UDP GOOSE/SV, IEC 61850-90-5 Tunnel and Client-Server profile. Each data type should be further restricted to provide more granular key or authentication exchange. Extra restriction required for encryption, based on the data content.
- Except for publishing keys, KDC shall support key publishing upon client request.

4.6. Services

The IEEE C.37.108.2 standard describes four message types (frame types) without defining “*services*” explicitly. The four message types are *The Data Frame*, *The Configuration Frame*, *The Header Frame* and *The Command Frame*. These frames have now to be mapped to IEC 61850 services. Regarding command, services can be performed by similar IEC 61850 services. (IEC 61850-90-5 2012: 57).

- *Control Blocks*: According to IEC/TR 61850-90-5 (2012: 57-60), there are defined two new control blocks, whose functions allow sending of SV and GOOSE information. For compatibility reasons the GOOSE and SV control blocks will remain the same, but two new functional restrictions will be added to LN0.
 - *RS (Routable SV)*: It indicates a functional restriction for R-SV packets. The control blocks with this restriction are defined as R-MSVCB (*Routable Multicast Sample Value Control Block*).
 - *RG (Routable GOOSE)*: It indicates a functional restriction for R-GOOSE packets. The control blocks with this limitation are defined as R-GoCB (*Routable GOOSE Control Block*).

The new control blocks have an impact on several parts of the standard. Among others, IEC 61850-8-1 should be updated to specify the mapping of the GoCB to the R-GoCB and IEC 61850-9-1 should be updated to specify the mapping of the MSVCB to the R-MSVCB.

The *R-MSVCB* uses the *SV* profile to transmit the synchrophasor streams by utilizing an IPv4 or IPv6 transport profile. To accomplish that, the R-MSVCB structure will be same as MSVCB but with the addition of the *SecurityEnable* and the *UDPCOMADDR* attributes.

The *R-GoCB (Routable GOOSE Control Block)*: For the transmission of GOOSE over IP the R-GoCB should support IPv4 and IPv6 multicast. The R-GoCB structure will be same as the GoCB, enhanced by the *SecurityEnable* and the *UDPCOMADDR* attributes.

The detailed structure of the routable SV and GOOSE control blocks can be seen in the 90-5 part. The new UPPCOMADDR attribute provides information regarding the priority, VID (Virtual ID), APPID, (Application ID), IP-address, etc.

- *Data Transmission Service*: For the transmission of synchrophasor data frames, the IEEE C37.118.2 allows RS-232 serial, TCP or UDP protocol. TCP and UDP are routable protocols since they are IP-based, while the RS-232 is less popular since networks are dominating. The application of TCP is limited as well in IEEE C37.118.2, so for obvious reasons, the majority of applications use the UDP for the transmission of data frames. (IEC/TR 61850-90-5 2012: 62)
- *Common Data Fields*: For a successful synchrophasor protocol, there are some data which must be present in any case and are arranged in defined order at the beginning of the dataset. These data include, among others: the time stamp, the timing source, and synchronization status information. (IEC/TR 61850-90-5 2012: 62)

4.7. Time Synchronization

For synchrophasor measurements, the UTC-time is required and can be provided by any source can offer UTC at the required accuracy and reliability. The required accuracy is 1 μ s. The measurement is made continuously, and a typical system requires an accuracy of 5 μ s (IEC/TR 61850-90-5 2012: 63).

4.8. Synchrophasor Profile Mappings

4.8.1. General

According to IEC/TR 61850-90-5 (2012: 66), there is no extension requirement for control and configuration services since they are using IEC 61850 methods with MMS over TCP/IP. Regarding data transmission, a new UDP mapping is required, with the desire to utilize the GOOSE and SV protocols without changes. Therefore, it is identified the ability to “*tunnel*” the currently Ethernet-based GOOSE and SV packets over UDP/IP. Figure 16 provides an overview of the synchrophasor service mapping.

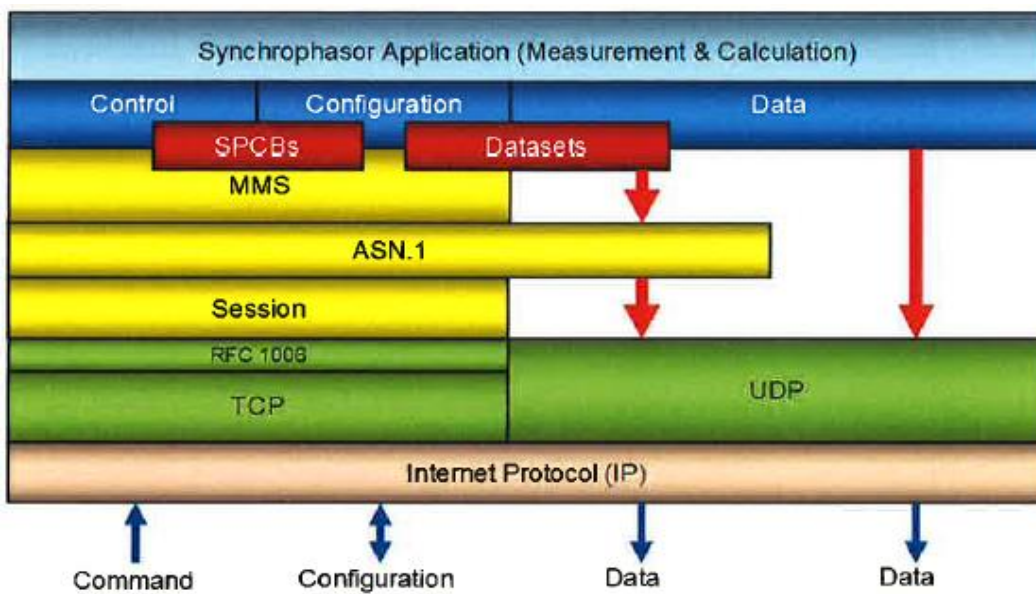


Figure 16. Overview of the general service mappings. (IEC/TR 61850-90-5 2012: 66).

4.8.2. A-Profile

The A-Profiles (*Application Profiles*) consisting of the GOOSE and SV APDUs (*Application Protocol Data Units*), will be tunneled using the session protocol defined in this document. Figure 17 shows the A-profile being used to transport synchrophasor information in a secure and routable way, intending to minimize the changes of the used APDU. The A-profile uses the IEC 61850-8-1 GOOSE and IEC 61850-9-2 SV as application and presentation layer. (IEC/TR 61850-90-5, 2012: 67).

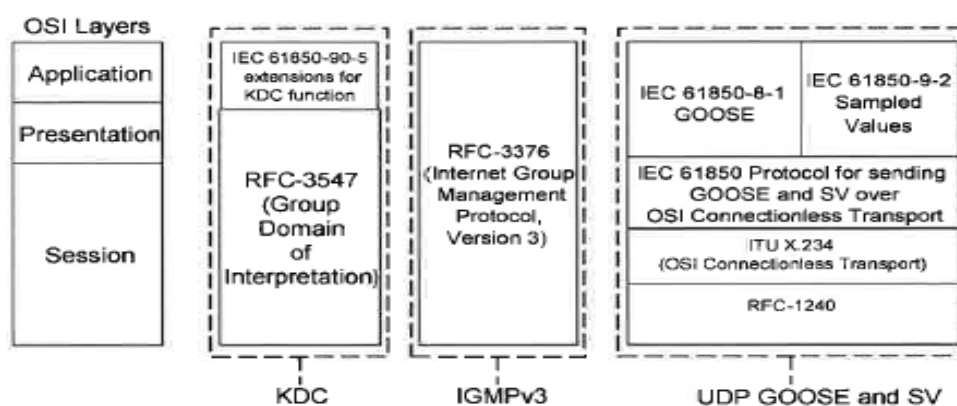


Figure 17. IEC 61850-90-5 A-Profiles. (IEC/TR 61850-90-5, 2012: 67).

The *Session Layer* consists of a session protocol capable of conveying required key parameters, the ITU X.234 (OSI Connectionless Transport) and the RFC-1240 (OSI Connectionless Transport over UDP).

4.8.3. Session layer

According to IEC/TR 61850-90-5 (2012: 72-84), the session header contains the following information (Figure 18):

- *Session Identifier (SI)*: There are four types of payload allowed by the Session Identifier:

- Tunneled GOOSE and SV packets: This type of SI allows both GOOSE and SV to be contained in the payload. The PDU can be only tunneled type.
- SPDUs containing non-tunneled GOOSE APDU. This SI restricts the payload to contain GOOSE APDU types.
- SPDUs containing non-tunneled SV APDU. This SI restricts the payload to contain PDU types of SV APDU.
- SPDUs containing non-tunneled management APDU. This SI limits the payload to contain PDU types of MNGT APDU.

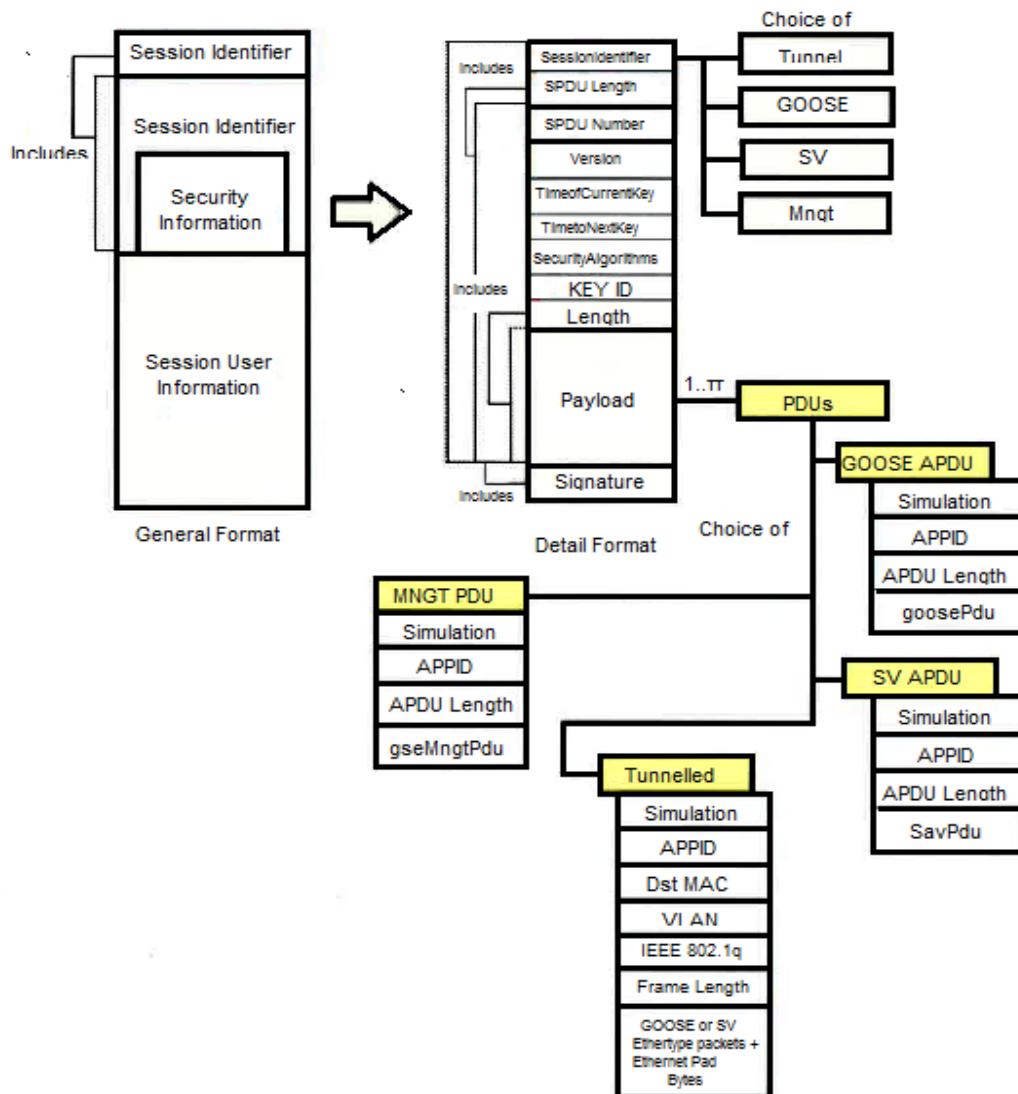


Figure 18. The structure of IEC 61859-90-5 session protocol (IEC 61850-90-5, 2012: 72).

- *Session Header* contains the following information:
 - *SPDU (Session Protocol Data Unit) Length*: The 1st version of UDP protocol allows a maximum packet of 65535 octets.
 - *SPDU Number*: The SPDU number is used by the subscriber for the detection of delivered duplicated or out of order packets.
 - *Session Protocol Version Number*
- *Security Related Attributes*:
 - *TimeofCurrentKey*: It represents the SecondSinceEpoch, and it has a size of four octets. SecondSinceEpoch is the interval in seconds counting from 1970-01-01 00:00:00 UTC.
 - *TimetoNextKey*: It has a two octet size, representing the time in minutes before the new key being used.
 - *SecurityAlgorithms*: It has a two octet size, and the most significant octet indicates the encryption type being used. None, AES-128 or AES-256.
 - *Key ID*: It has a four octet size, and it is assigned by KDC referred to the current key being in use.
- *User Data*: User data consist of user length and payload.
 - *User Length*: The maximum value of this attribute depends on the maximum size of the SPDU length.

4.8.4. Tunneled payload

According to IEC/TR 61850-90-5 (2012: 80-84), the payload section allows multiple PDUs to be aggregated in one SPDU. As it is seen above, there are four types of payloads, GOOSE, SV, Tunneled, and MNGT. Regarding the tunneled payload, PDU needs to include the following critical information required to re-emit the appropriate frames at the end of the tunnel(s).

- The multicast destination MAC address (*destinationMACAddress*), containing the original MAC address that the tunneled PDU (GOOSE/SV) was sent to.

- *TPID (Tag Protocol Identifier)*: It indicates the EtherType assigned for IEEE 802.1q Ethernet frames and it has a standard value (0x8100).
- The *TCI (Tag Control Information)* shows the user priority and its value should be set during the configuration to distinguish the SVs and time critical protection GOOSE messages from low priority messages.

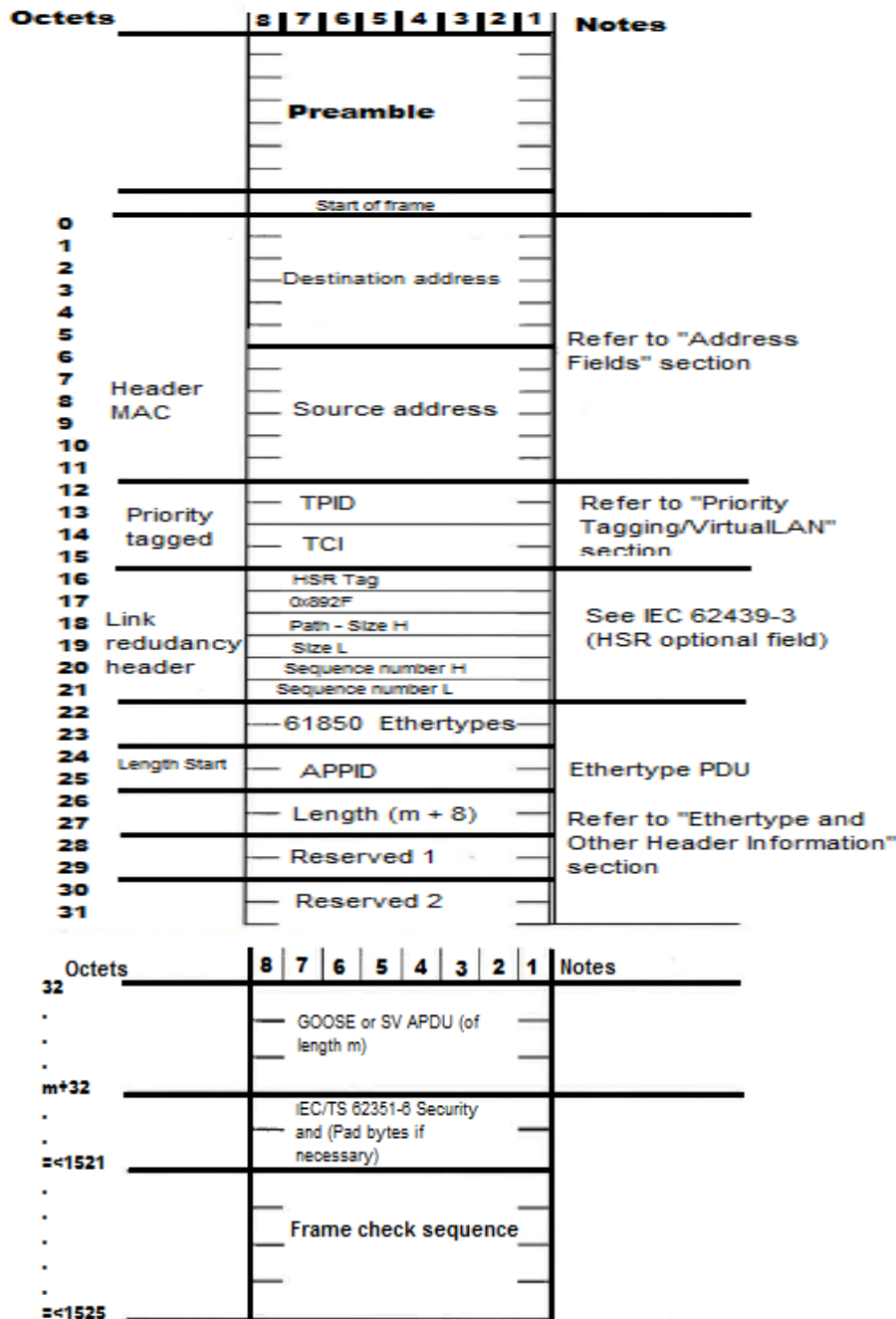


Figure 19. IEEE 802.3 frame format for SV & GOOSE (IEC 61850-90-5 2012: 81).

- The Ethertype PDU starting at the 22nd octet of the Ethernet frame is adopted by the IEEE 802.3 standard, indicating the type of protocol being encapsulated in the frame's payload. The types of protocols allowed to be tunneled are GOOSE Type 1/1A and SV. (IEC/TR 61850-90-5, 2012: 82)
- *Signature*: The signature will be used for the authentication/integrity of the octets from the *SI* till the end of the user's data payload. Within the signatures calculation, it is not included the signature's production.

4.8.5. KDC profile

According to the IEC/TR 61859-90-5 (2012: 85-87), the standard uses two types of key request and exchange mechanism. The first is for the identification of the payload and the second for the determination of the type of the returned key. The communication includes three phases, phase-1 for connection establishment and authorization, phase-2 for encryption and phase-3 obtaining the appropriate key.

The standard uses a payload extension mechanism in order to allow other protocols or organizations to use the IEC 61859-90-5 payload extension. The general format must include the payload identifier extension (default value), the payload extension length, the Object_identifier_tag, and the Object_Identifier (a set of octets representing ASN.1 encoded Object Identifier). The type of payload following depends on the value of the identifier.

- According to the IEC/TR 61859-90-5 (2012: 88), to achieve interoperability, each payload must include the following definitions:
 - The *VERSION* of the payload.
 - *DEST_MULTICAST_ETHERNET_ADDRESS*: It has six octets value and shall be specified per Ethernet transmission.

- *IP_ADDRESS*: This component specifies the IPv4 or IPv6 destination address for which a key is requested.
 - *DATASET*: This component allows the differentiation of an IEC 61850 DataSet Reference (DSRef), as specified in IEC 61850-7-2.
 - *61850_UDP_ADDR_GOOSE and 61850_UDP_ADDR_SV*: The GOOSE and SV payload for the IP version are the same.
 - *61850_UDP_Tunnel*: The Tunnel payload is similar to the above payloads except that dsRef is missing because multiple Ethernet multicast frames can be sent in one Tunnel SPDU.
 - *61850_ETHERNET_GOOSE and 61850_ETHERNET_SV*
- Regarding key download payload, RFC 3547 specifies the general format of key distribution to group members, allowing a range of 128-255 for private use. The KD (Key Download) payload identifier shall define several identifiers for the IEC 61850-90-5. Next, are shown the numbers being used.
- 192, indicating 61850_ETHERNET_GOOSE_OR_SV key identifier type
 - 193, indicating 61850_90_5_SESSION key identifier type
 - 194, indicating 61850_8_1_ISO9506 key identifier type
 - 195, indicating 61850_UDP_IP_AGGR key identifier type
 - 196, indicating 61850_UDP_MNGT key identifier type4

Additionally, there is the requirement for distribution of the current and next key among the authenticated clients. KDC generates the Current and Next Key ID. The key values are used in the Session protocol to identify the key is used. These values consist of four octets and are unique within the lifetime value and download type.

4.9. T-Profiles

As specified in IEC/TR 61859-90-5 (2012 94-97), there are three different A-Profiles. For each A-Profile correspond three independent T-Profiles (*Transport Profiles*). The relation of A and T profiles is shown in the next figure. It can be seen that the several T-profiles have common elements for the network and data link layer. Nevertheless, there are some differences regarding the transport layer.

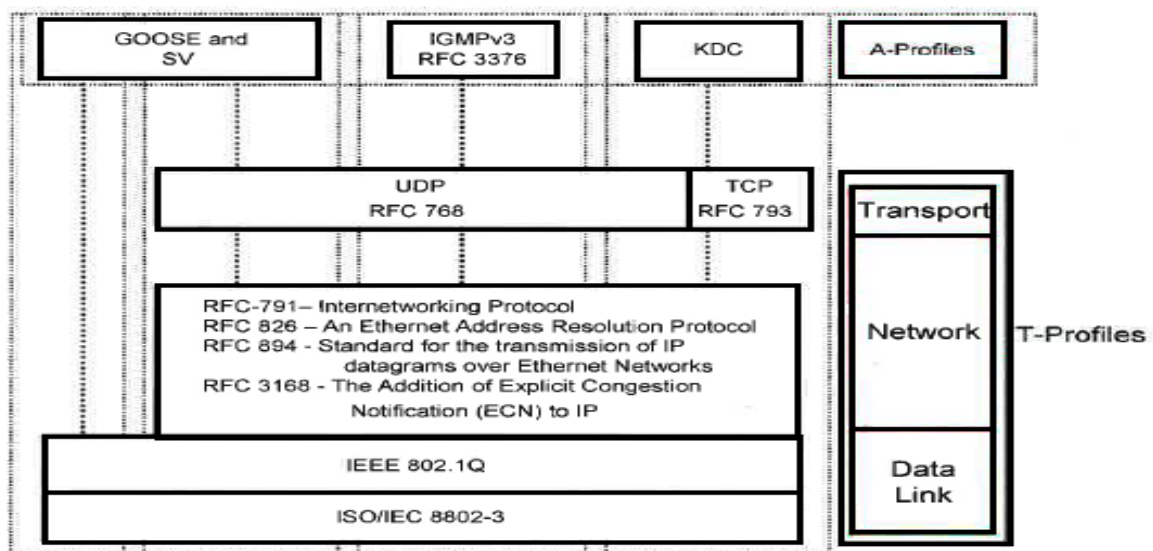


Figure 20. Association of A-Profile to T-Profiles. (IEC 61859-90-5, 2012: 95)

The supported A-profiles are:

- *T-Profile to support GOOSE and SV A-Profile over Ethernet:* It is used to create an IEC 61859-90-5 *Tunnel*, according to the specifications of Session layer within this document.
- *T-Profile to support GOOSE and SV A-Profile (UDP):* For the UDP T-Profile, the destination port shall be port 102 (predefined), and the source port shall be assigned locally.
- *IEEE quality of service (IEEE 802.1Q):* Applications conformed to this standard shall provide a transport service data interface in order to specify the destination IP Address, VLAN and the Ethernet Class of Service.

- *T-Profile to support KDC (TCP and UDP)*: The destination port shall be port 898 (predefined), and the source port shall be assigned locally.

According to IEC/TR 61859-90-5 (2012: 97-99), although the Network Layer protocols are common within the T-Profiles, the network layer can be distinguished by the IPv4 or IPv6 support. Implementations conformed to this standard shall support at least IPv4.

Bits: 4	8	16	20	32
Version	H.Length	TOS	Total Length	
Identification		Flags	Fragment Offset	
Time To Live	Protocol	Header Cheksum		
32 bits Source Address				
32 bits Destination Address				
Options				

Figure 21. The format of IP header. (IEC/TR 61859-90-5, 2012: 98).

Figure 21 shows the IP header fields as defined in RFC 791. Congested packets can be signaled using data contained in the ToS (*Type of Service*) field. The ToS field is divided into DSCP (*Differentiation Service Code Point*) and ECN (*Explicit Congestion Notification*) sub-fields.

Any packet may have a particular treatment along its end-to-end path through the network. This is accomplished through the DSCP signaling, which allows the change of behavior at any hop or router. DSCP can be set in the source before transmission or anywhere in the network depending on the implementation. Various router vendors recommend settings for DSCP values according to the traffic class. Additionally, it is recommended the notification of the lost packets in case of congestion indicated by IP packets delivered to the subscriber.

Both A- and T-Profiles are marked as EF (*Expedited Forwarding*), scheduled as Low Latency Queues traffic to expedite their proceeding versus other types of traffic.

Implementations conformed to this technical report shall support unicast and multicast IP addresses. The Time-to-Live parameter should be set to a value (32) or greater to allow UDP/IP packet routing.

4.10. The Effects on the IEC 61850-5

The effects on the IEC 61850-5 include the addition of the following performance classes since part-5 does not have adequate time class specifications to support synchrophasor implementations mentioned within this document. (IEC/TR 61850-90-5, 2012: 99-101).

Table 4. Performance classes to be added to part-5. (IEC/TR 61850-90-5, 2012: 100)

Performance class	Requirement	Transfer time class	Transfer time	Typical for interface (IF)
P13	Delay acceptable for protection functions using the measurements in the substation	TT6	< 3ms	IF8
P14	Delay acceptable for other functions using the measurements in the substation	TT5	< 10 ms	IF8
P15	Delay acceptable for protection functions using the measurements in between substation	TT6	< 3 ms	IF11
P16	Delay acceptable for protection functions using the measurements in the substation	TT5	< 10 ms	IF11

This message type consists of output data from synchrophasor devices independent from calculations and synchronizing methods. The data appear in continuous streams of synchronized measurements from each IED. Transfer time results in a constant delay in the stream of synchronized data, delaying the functions that use the measurements (visualization, protection, etc.). So, the requirements for the transfer time depend on the application. For example, protection applications require short transfer time.

5. CONDUCTING THE PRACTICAL PART_1

5.1. Introduction

For the accomplishment of this thesis, it was necessary to pass through multiple steps in order to acquire the necessary theoretical and practical background. After that, it was possible to reach the final target that was the transmission of GOOSE messages via the *tunneling* and *routing* method and the evaluation of the different technologies.

The first step was to pass through the IEC 61850 documentation, consisting of ten parts, plus the editions 90-1 and 90-5. In addition, I studied the DEMVE training material, visiting in parallel the DEMVE laboratory in the Technobothnia, trying to familiarize with IEDs. The work with IEDs included, connection to the devices using their local IP address and configuration of their parameters using the corresponding software (Vampset, PCM600). Additionally, I worked on some exercises prepared by Mike Mekkanen. Next, I used the Omicron IEDScout device for the generation of GOOSE messages and configuration of their parameters, while Wireshark software was used to detect and analyze the messages. In the last step, I configured the Viola devices and due to technical problems I achieved my target by the BeagleBone platform plus the *HAMACHI* and the *LibIEC61850* library.



Figure 22. The DEMVE project, a section of the Technobothnia laboratory.

5.2. Description of the Practical Part_1

The purpose of this thesis was to achieve inter-substation communication based on communication solutions introduced by the IEC 61850-90-1 and 90-5 parts. The LAN infrastructure of the reference substation had to be extended via the routing and the tunneling/VPN method and reach the remote substation. The applied methods had to be evaluated based on the acquired round-trip times.

The public network (Internet) is considered vulnerable, so when we talk about vital facilities such as substations the security of the communication should be seriously taken into consideration. A solution for secure communication and encryption of data was the use of VPN-tunnel (*Virtual Private Network*), a method suggested in the 90-1 issue of the standard for tunneled (indirect routing) communication.

After discussions with my instructor, Mike Mekkanen we conclude that the use of the Viola systems will be ideal to reach our goal. The Viola M2M Gateway is a network device that enables *VPN* connection between the company's network and remote Viola Arctic devices. It can also be used to control and monitor Arctic devices on local or remote networks. The concept of the Viola system is described in the figure below, while the concept of the VPN-tunneling it is already introduced in the 2nd chapter of this thesis.

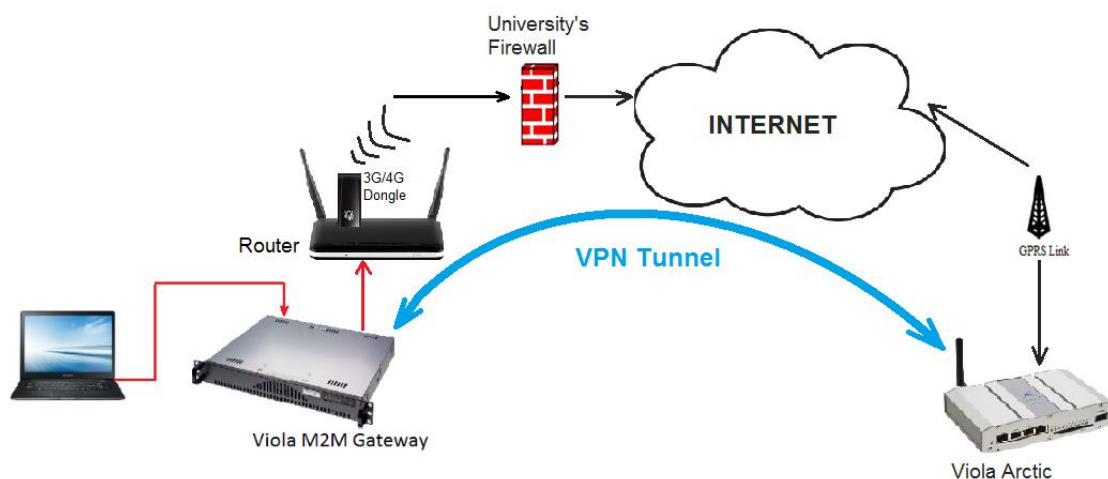


Figure 23. The concept of the practical part_1.

As the picture above depicts, the M2M gateway, has the inherited capability to establish a VPN connection with an Arctic device that may be located in remote area. In a typical application, each Arctic device has its own unique LAN that is tunneled (routed) over the open-VPN to the M2M gateway. Via the practical implementation, it was pursued a further investigation regarding VPN tunneling communications.

The idea was to simulate the transmission of a GOOSE message via the *PING* software utility, over VPN tunnel. The ping is available on Linux and Windows, and it checks if a destination computer/device is reachable via pinging and waiting for the echo to return to the source. The size of the message's payload is not standard, but it is case-dependent. For this experiment, it was chosen a size of 1525 bytes, representing a large data load having higher demands on transfer time. The concept of transfer time is already introduced in the 3.5 subchapter of the thesis.

5.3. Presentation and Configuration of the Devices in the Practical Part_1

5.3.1. The Viola M2M Gateway



Figure 24. The M2M Gateway (Viola Systems M2M Gateway 2012:1).

According to the *Viola Systems M2M Gateway* (2012:1), the *Viola M2M Gateway* is an industrial level device with pre-installed software for communication between a central device and Arctic devices that may be located in remote sites. Via the Viola communication solutions the local network can be expanded over wireless (3G/ EDGE/

GPRS/LTE), so the remote Arctic devices will be accessed as they were on the local network.

The operation of the M2M requires a computer with access to the internet and an HTML browser. M2M enables a secure communication via VPN tunnels or static IP addresses to Arctic device(s). The VPN connection is initiated by an Arctic device, and subsequently, the M2M decides if it will allow the connection or not, based on its configuration. Among others, the M2M offers a variety of advanced features such as routing, internal firewall, VPN connection, and remote management of a device.

The first step for the configuration of the M2M gateway is to connect the power supply and the RJ45 cables to access the LAN & WAN. A PC is connected to the LAN port while the WAN port accesses the internet through the connection to the router. The user is now ready to use the device and adjust settings through the graphical user interface, (figure 25). The *Network Configuration* is selected to set the local IP of the device, while for the public IP is selected the DHCP choice.



Figure 25. The graphical user interface of the M2M gateway.

Regarding the configuration of the open VPN, must be executed after the configuration of the Arctic device. From the M2M's graphical user interface, we select *OpenVPN Configuration* and next click on the *OpenVPN Easy mode* icon. We give the public IP of the M2M gateway, which the Arctic device is going to use when connecting to M2M.

Subsequently, we select *Create Client* which creates an authentication certificate. This certificate needs to be exported from M2M and uploaded to the remote Arctic device, certifying that the device is trusted and authentic. Figure 26 shows a created client (remote Arctic), while the *Tunnel IP* refers to the *virtual-IP* (IP-Address of the remote Arctic).

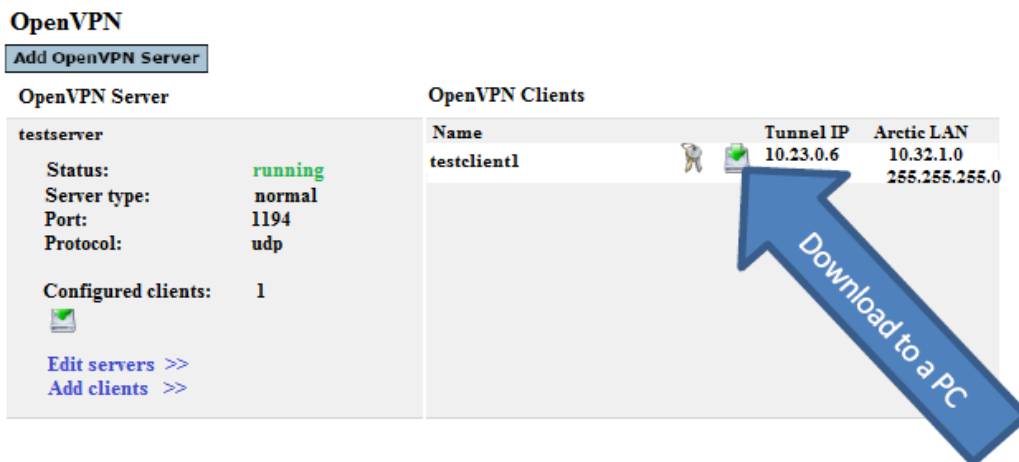


Figure 26. Download of the client authentication certificate.

Next, the certificate is downloaded from the M2M gateway and uploaded to the Arctic device. The LAN IP address of the Arctic has to be changed now to the *virtual-IP* set beforehand from the M2M.

Once we have downloaded the open-VPN program, we connect the Arctic to PC and start the open-VPN program running as administrators. The VPN client has started but it has not created a tunnel yet to the server (M2M). We click '*connect*' to establish the connection, and once the VPN icon turns green, the connection has been established. To verify that everything is operating correctly, we have to ping the Arctic's virtual IP from a PC or the M2M.

5.3.2. The Viola Arctic 3G/LTE Gateway



Figure 27. The Arctic Gateway (Viola Arctic Gateway Manual 2014:1).

According to the *Viola Arctic Substation Gateway* (2014:1), the Arctic is an industrial level wireless gateway using the high-speed 3G networks (HSPA+). When it is connected over the Ethernet consists an ideal solution for the monitoring and control of substation devices when high data bandwidth and low latency is required.

Arctic achieves high communication security using an internal firewall and robust data encryption protocols. It offers static IP addressing independent of mobile operator and VPN connectivity with Viola M2M Gateway. It supports multiple VPN encryption protocols increasing the data security. Regarding system reliability, it incorporates Dual SIM, while SIMs from different operators can be used in a single device increasing the flexibility and the modifications of the communication solution.

The Arctic Viola requires a power supply of 12-48Vdc, but before the power connection, the two SIM cards need to be installed in the SIM card holders. For the configuration, it is required a PC connected via an Ethernet-cable. Once the device is connected, it is necessary to set the hostname and the time. After that, a look at the *Status Screen* of the device provides useful information about its operation (Figure 28), such as hardware/firmware version, time, LAN connection, mobile internet connection (GPRS/3G/LTE), VPN, and firewall status.

Hardware and firmware version											
Product name: Arctic LTE Lite C-1260 (C-1260)											
Firmware version: arctic3 firmware 3.2.6 (build 4363)											
Hardware version: 0x20											
Hardware serial number: 11252266											
Device serial number: ARCMX28-454-128-027E6A											
Uptime											
4 hours, 41 min											
Network interfaces											
Interface	IP addresses	MAC address	MTU	Bytes		Packets		Errors		Dropped	
				RX	TX	RX	TX	RX	TX	RX	TX
lan0	10.10.10.9/24 fe80::206:70ff:fe02:7e6a/64	00:06:70:02:7e:6a	1500	13125374	286494679	155728	246868	0	0	0	0
wwan0	10.37.191.116/30	ea:25:27:7a:01:08	1500	282920522	9640029	246060	154619	0	0	0	0
Routing Table											
Kernel IP routing table											
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	iface				
192.89.123.231	0.0.0.0	255.255.255.255	UH	0	0	0	wwan0				
192.89.123.230	0.0.0.0	255.255.255.255	UH	0	0	0	wwan0				
10.37.191.116	0.0.0.0	255.255.255.252	U	0	0	0	wwan0				
10.10.10.0	0.0.0.0	255.255.255.0	U	0	0	0	lan0				
0.0.0.0	10.37.191.117	0.0.0.0	UG	0	0	0	wwan0				
Link Status											
For up-to-date mobile network go to modem info page .											
lan0: negotiated 100baseTx-FD, link ok, MDI-X (auto)											
wwan0: Signal level: -71 dbm (67% normal), current service: LTE, last information update: 2015-10-20 15:48:25											
VPN Status											
No VPN tunnels configured											
Firewall Status											
Track applications - on											
Filter - on											
LAN-In accept - on											
Pass vpnin - on											
GUI anti-lockout - on											
Pass lanash - on											
LAN-LAN accept - on											
Pass lanvpn - on											
Pass vpnlan - on											
LAN-Out accept - on											
D-NAT - off											
S-NAT - on											
Qos - off											
Deny ipv6 - off											
Reduce OpenVPN bridge multicast - off											
Total fail - 0											
Total ok - 27											
Serial Port Status											
RS1: console, 1152200, 8, N, 1, tx:1202 rx:0 RTS/DTR											
RS2: disabled											
Viola Patrol Status											
Disabled											

Figure 28. The Arctic's Status screen.

LAN and WAN parameters can be set at *networking* section. Regarding WAN settings a variety of options is offered, such as WAN speed, VLAN tagging on WAN interface, use of WAN plus cellular network as a backup. Since the device incorporates two SIM cards, one can be set as primary WAN1 and the other as secondary WAN2, used only in case of 1st network failover. In case that required, PIN codes for the SIM cards can be

used. Additionally, Arctic Viola incorporates pinging functionality, having as ping target the M2M VPN IP or the M2M public IP.

Viola Arctic incorporates advanced settings regarding the choice of the network service/frequency (Figure 29). The user can manually select the preferred mobile frequency according to the demands. This feature will allow us later to evaluate which band has the better performance regarding the VPN connection.

Advanced	
Network	2G Preferred
Service Frequency	Automatic 3G Only (UMTS/HSDPA) 2G Only (GPRS/EDGE) 3G Preferred 2G Preferred GSM 900 MHz GSM 1800 MHz GSM 900/1800 MHz GSM All frequencies WCDMA All frequencies WCDMA 900 MHz WCDMA 2100 MHz WCDMA 900/2100 MHz
Operator	Automatic Selects the fastest available
Idle Timeout	To allow only certain operator define the PLMN code here. Normal
Duration	The Mobile WAN connection is restarted when it has been unused for given ti
Reconnect Interval	The Mobile WAN connection is restarted when it has been connected for given
	How many seconds to wait between failed connection
	Submit Reset

Figure 29. The Arctic's advanced network service/frequency.

Regarding VPN services, Viola Arctic offers a choice of three VPN tunnel types: *OpenVPN*, *SSH-VPN*, *L2TP-VPN*, and *IPsec*.

- The OpenVPN tunneling is recommended for the connection of Arctic with M2M Gateway. It is light-weight and implements encryption. Additionally, the M2M Gateway can create OpenVPN client configuration for a remote device operation and management. The setup options are The *Easy OpenVPN* setup (automatic) and the *Advanced OpenVPN* setup (manual setup).
- The SSH-VPN is used for backward compatibility reasons. In the case of use, the M2M's peer name and the Arctic's hostname shall be identical.
- The use of the L2TP-VPN is considered insecure since it does not use encryption. Although it is not recommended for VPN implementation, it is justified in cases where fast and lightweight VPN is required, and an additional safety layer exists.

Before a VPN connection to be established between the M2M gateway (server) and the Arctic (client), the M2M gateway needs to generate a certificate including the Arctic's credentials, in order to be recognized as a trusted and an authentic device for communication. The generated certificate is later uploaded to Arctic device in order to recognize the M2M as an authentic device for communication.

Arctic saves the system log and indicates it when requested. The Viola's technical support requires the system log in case of troubleshooting since it provides a variety of information, such as Arctic's boot time, the VPN tunnel status, registration to network, signal level, mobile operator, the mobile frequency being in use and other important data. The system log can be downloaded as a text file to a PC or it can be sent to a remote server from the Arctic device.

5.3.3. The D-Link Wireless N300 Multi-WAN Router



Figure 30. The Wireless D-Link Router. (D-Link DWR-116 User Manual 5).

According to *D-Link DWR-116 User Manual* (2013: 1-2), the DWR-116 wireless multi-WAN router offers a quick and easy installation, allowing the access to mobile broadband networks anywhere in the world. Once connecting the USB modem (dongle) the user can share the 3G/4G LTE Internet connection via a secure 802.11n wireless network or the 10/100 Ethernet port. The DWR-116 router is ideal in situations where the conventional network is inaccessible.

5.3.4. The D-Link router configuration

According to *D-Link DWR-116 User Manual* (2013: 8-80), in order to access the configuration interface of the router, we open an internet browser and insert the default IP address of the router (*192.168.0.1*). The DWR-116 offers two ways to setup the Internet connection, the manual internet connection setup wizard (*step-by-step process*) and the Web-based Internet Connection Setup Wizard. Next, we create a password to access the router.

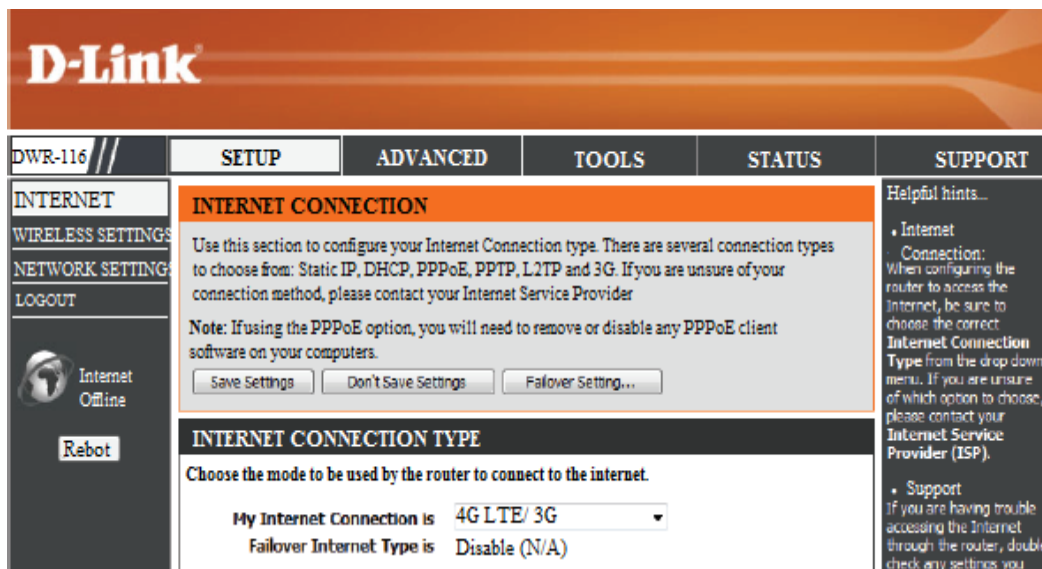


Figure 31. The setup of the DWR-116 internet connection.

The DWR-116 supports several kinds of WAN interfaces, allowing the user to choose a WAN or a WWAN (3G/4G LTE) connection. The configuration allows the setup of a primary WAN and a backup WAN in the case of primary's failover. The chosen internet connection was the *3G/4G LTE* since it is suitable for combination with a USB Dongle (Figure 32). The SIM card being used for the Dongle was from the Finnish *Elisa* mobile company.

INTERNET CONNECTION TYPE
Choose the mode to be used by the router to connect to the Internet.

My Internet Connection is 4G LTE /3G
Fallover Internet Type is Disable (N/A)

4G LTE /3G INTERNET CONNECTION TYPE
Enter the information provided by your Internet Service Provider (ISP).

Dial-Up Profile : Auto-Detection Manual
Country : Finland
Telecom : Elisa
3G Network : WCDMA/HSPA
Username : (optional)
Password : ***** (optional)
Verify Password : ***** (optional)
Dialed Number : *99#
Authentication : Auto
APN : internet (optional)
Pin Code :
Reconnect Mode : Auto Manual
Maximum Idle Time : 600 seconds
Primary DNS Server :
Secondary DNS Server :
Keep Alive : Disable Use Ping
Ping IP Address : 8.8.8.8
Ping Interval : 60 seconds
Bridge ethernet ports : Enable

Figure 32. The DWR-116 Internet configuration.

D-Link

DWR-116 // SETUP ADVANCED TOOLS STATUS

VIRTUAL SERVER

The Virtual Server option allows you to define a single public port on your router for redirection to an internal LAN IP Address and Private LAN port if required. This feature is useful for hosting online services such as FTP or Web Servers.

Save Settings Don't Save Settings

Well known services WEB (80) Copy to ID --
Use schedule -- select one -- ON---

VIRTUAL SERVERS LIST

ID	Service Ports	Se	Enable	Schedule Rule#
1			<input type="checkbox"/>	Add New Rule...
2	102	192.168.50.230 : 102	<input checked="" type="checkbox"/>	Add New Rule...
3			<input type="checkbox"/>	Add New Rule...
4			<input type="checkbox"/>	Add New Rule...

Internet Online
Reboot

Figure 33. The DWR-116 advanced settings for port forwarding.

Regarding PORT forwarding, via the *ADVANCED / VIRTUAL SERVER* settings, it is possible to define a single public port and redirect it to an internal LAN IP address. This

feature is useful for redirecting of traffic or a message/packet to the desired device in our network (figure 33). We first enter the IP address of the device on the local network we want to allow the incoming service, and next, define the port number that we would like to open and enable the rule. Since the M2M is accessing the internet through the router, the incoming traffic should be redirected from the router to the M2M's local IP.

5.3.5. The Huawei E392 TDD-LTE USB Stick

According to *Product Description HUAWEI E392 TDD/LTE USB* (2011), the HUAWEI E392 is a multi-mode modem supporting 2G, 3G, and 4G frequencies: LTE TDD/FDD, HSPA+, UMTS, EDGE, GPRS, and GSM standards. Before the operation, we insert the SIM card in the card holder and next the device is connected to a PC via the USB interface. The E392 driver and the client software are installed on the PC automatically.



Figure 34. The HUAWEI E392 USB Stick (HUAWEI Product Description 6).

The device switches automatically between different networks to provide the optimum internet connection. Regarding the data packets, the E392 supports the data service based on LTE/EDGE/GPRS/HSPA+/UMTS. It supports speeds up to 100 Mbit/s (download) and 50 Mbit/s (upload) under a 4G LTE network.

The HUAWEI was combined with the router in order to access the mobile internet. The reason of this combination was the configuration options that the router offers while the dongle does not. For example, if it is required the redirection of traffic to a particular device on the local network, the router can accomplish that through the port forwarding. That would be impossible by the dongle itself since it is not configurable.

5.4. Analysis of the Technical Problems Faced in the Practical Part_1

The technical problems or misbehaving of devices forbade me from accomplishing the experiment, although I followed the instructions carefully and I was very close from acquiring the results. Additionally, I would like to refer that my networking knowledge was not that deep, so it was not easy to understand and solve all the issues might appear during the practical part. The technical issues are analyzed below:

1. The main problem being faced was the misbehaving of the Viola M2M gateway. The device was not connected to the internet despite the careful study of the instructions in the user manual. So the device had to be reset and fix the problem.
 - Regarding the M2M's reset, proper instructions were missing from the user manual, and the technical support claimed that it was enough to press the reset button on the device, which was insufficient. After further search and trials, I found out that a monitor has to be connected to the M2M as well as a mouse and keyboard to interact with the GUI of the device and reset it to factory's default settings. Additionally, before the reset, a password was required to start the configuration, which was also missing from the manual and not provided immediately from the technical support when I asked for help.
 - After the reset, the device was operating normally, and it could access the internet, but despite that, the establishing of a VPN connection to the Arctic device was impossible.
 - After powering off, the device was unable again to connect to the internet and a new reset was required. This procedure on a daily basis made the continuing of the experiment very tough.
 - Additionally, the SSH facility provided by the device could not recognize the *ping* command. So, in order to ping from the M2M, it was necessary to use the PuTTY software. The pinging was needed to detect if the device can access the internet or not.

2. Regarding the Arctic Viola gateway, some minor problems were faced, without contributing to the experiment's fail.

- The main issue was that the signal's level was maintained at quite low levels, and it is interrupted every few seconds during the access of the mobile WAN. All the possible network frequencies had been tried, without leading to a remarkable improvement (Figure 29). Additionally, the device was placed in several points of the Technobothnia laboratory to improve the signal's gain, but the results were poor. In the manual is suggested the use of an external antenna to improve the signal.
- During the configuration, the device was logging off every 3-4 minutes, despite the set of the session time at the maximum value (1440 min). That made the configuration and study of the menu inconvenient.

3. The communication with Viola's technical support was constant, and the log files from both devices were sent to them, without being able to identify the problem.

4. Another constraint that can be mentioned is that the university's internet could not be used to check connectivity of Viola devices since its firewall is set to block pinging, so the only way to access the web was via the mobile internet.

6. CONDUCTING THE PRACTICAL PART_2

Due to the technical difficulties analyzed above, I decided to conduct the experiment by a combination of software and microprocessor. The target was to establish a client/server communication link where the client and server were representing two remote substations. Once again the exchanged messages were ping messages having a data-load of 1525 bytes, representing GOOSE messages. The different methods had to be evaluated based on the acquired round-trip times.

The idea was same with the first part except for the fact that there were no dedicated devices with inherited VPN-connection capability. At first, the devices were two PCs running the *LibIEC61850* library and the *HAMACHI* software to establish a VPN link. Next, they were used two Beagle-platforms to ping to each other via the 4G mobile network. The details of the implementation are explained below.

6.1. Introduction to Libiec61850-0.9.0.2 Library

According to LibIEC61850 (2016), the *LibIEC61850 Project* is an open source library written in C, aiming to provide a client and server library for the IEC 61850 MMS, SV, and GOOSE communication protocols. The software is available under the GPLv3 license, and it is developed by the electrical engineer and software developer Michael Zillgith (Germany).

The *LibIEC61850* library supports real-time intra-substation communication via GOOSE. The goal of the developer is to provide a portable implementation of all useful IEC61850 services, capable of running on embedded systems and micro-controllers. So far, the implementations run on embedded Linux systems, and PCs running Windows or Linux. In the webpage it is also included a set of simple examples, aiming to help users with their own IEC 61850 applications.

The API (*Application Programming Interface*) of libIEC61850 can be divided into the client and the server part, sharing both common elements. For both client and server exist two individual APIs, a low-level MMS API and a higher-level IEC 61850 API.

Regarding the server API, the library provides two different server APIs. The first is a generic MMS server API without the support of the IEC 61850, being suitable for generic MMS server devices implementations. The second server API is specific for IEC 61850 implementations, supporting the IEC 61850 control model, as well as the automatic generation of the MMS device model from the IEC 61850 data model. Additionally, it supports data sets and reporting, having a very low overhead.

Regarding the client API, the library provides also two different client APIs. The first is a generic MMS client API without supporting the IEC 61850, being suitable for generic MMS client applications. The second API is specific for IEC1850 applications, designed as close as possible to the IEC 61850 ACSI and supports model discovery, handling of data sets, configuration, and reception of reports, read/write data attributes.

Some of the features provided by the library are:

- Full support of ISO protocol stack on top of TCP/IP
- C code generation from SCL files for static implementation of the IED model
- Application program interface for MMS client
- Application program interface for IEC 61850 client/server
- Client & Server support for all IEC 61850 control models
- Standalone GOOSE publisher/subscriber code
- Standalone SV publisher/subscriber code
- Hardware abstraction layer and applications for WIN32, POSIX(Linux), and BSD systems (Mac OS X)
- Tool for converting SCL files into static IED models

6.1.1. Building the library and the examples

According to *libIEC61850* (2016), the *libiec61850* supports two different ways for building the library. First, the *make*-based system, working well with GNU-based toolchain, and second the *cmake* based system. So far, the embedded Linux systems are using the *make* system for cross-compiling. The examples included in the library are tested in platforms such as Linux/x86, Windows 7, Mac OS X 10.9, etc.

To build the examples, we start the *command line* and type the directory where our library is saved, (e.g., Desktop). Next, we type *make*. This command will build the examples for the host platform (Linux or Windows). To test the client/server communication we combine a client example with the corresponding server example. Use of third-party tools like Omicron IEDScout is also applicable. Ubuntu system allows cross-compiling for Windows by the install of the Ubuntu package *mingw32* and next typing: *make TARGET=WIN32*.

➤ **Running the examples**

First, the library has to be downloaded from the *libIEC61850* web-site. Regarding Linux, the server examples must be started as *root* since they are bound to the default 102 MMS port, and Linux allows this only for users having root permission. The command for starting a server example on an Ubuntu or other Linux system is:

```
sudo ./server_example1
```

➤ **Creating your own IEC 61850 server device**

The server API provides three different methods to set up the data model for the server. The first method creates a static model that cannot change during the runtime of the application. In this method the data model provided as SCL (ICD) file is converted to C code and it is compiled into the application. In the second method, the data model is generated during the runtime by API calls. This method is ideal for dynamic devices such as gateways, simulation tools, protocol converters, etc. The last method uses a configuration file format to provide the application with the IEC 61850 device model.

➤ **Generating the static model source code from an ICD (SCL) file**

The Java JRE 6 needs to be installed in order to create the model source code. Open the *command line*; open the library directory, and next go to the tools → model_generator directory, and enter:

```
java -jar genmodel.jar my_model.icd
```

The *my_model.icd* corresponds to an .icd file from the ready examples we have chosen to use. This command generates two files, the static_model.h, and static_model.c. These two files must be copied to the project directory (the folder containing our project), where the build system can find them. The static_model.c file defines the data structure for building up the IED data model, containing as well pre-configured values provided by the SCL file. The static_model.h file intends to be included by the code and defines handles for the efficient access of data model.

6.1.2. Analysis of the practical implementation

The library was mostly implemented in order to acquire experience and enhance my knowledge on the establishment of a communication link based on the IEC 61850 concepts. Nevertheless, once the user has the programming skills, he can use the library to build his own applications and extend the communication beyond the SAS's limits.

So the library examples were implemented for intra-substation communication (within the local network), and no “*tunneling*” or “*routing*” was used for communication since the ready examples are intended to be used within the SAS' limits (LAN).

Our practical part was implemented using two PCs running Ubuntu Linux, where a GCC toolchain and the *make* tool were pre-installed. At first, the library was downloaded from the *libiec61850.com* web-site. Since a client/server communication was required, one computer was running the server example, while the other the client.

Below are analyzed the steps for establishing a client/server communication.

- The *command line* was started typing the directory where the library was saved. By executing the *make* command, the library was build.
- Next, arbitrarily it was chosen one server and one client example included in the library file. The *server_example_goose* was chosen for the server, and the *client_example1* was chosen for the client.
- Before the connection of the client to the server, the IP address of the server was provided in the *.c* file of the client, (figure 35).

```
int main(int argc, char** argv) {
    char* hostname;
    int tcpPort = 102;      /* default PORT number*/

    if (argc > 1)
        hostname = argv[1];
    else
        hostname = "192.168.0.104"; /* IP address of the server*/
}
```

Figure 35. Definition of the server's IP address in the client's *.c* file.

- Similar changes were done in the server's *.icd* file regarding its IP and gateway.
- Next, the static model source code was generated from the *.icd* file of the server. From the command line → go to library directory → *tools* → *model_generator* and type: *java -jar genmodel.jar simpleIO_direct_control_goose.icd*

The above command generates two files: the *static.model.h* and the *static_model.c*. These files were copied to the folder containing our project.

- The last step was to run the examples. Two command lines were started from the two PCs. One PC was running the client, while the other the server, by typing:

```
sudo ./client_example1
sudo ./simpleIO_direct_control_goose
```

```

root@mike-ThinkPad-X230: ~/Desktop/libiec61850-0.9.0.2 (2)/examples/iec61850_client
client_example1.c CMakeLists.txt
root@mike-ThinkPad-X230:~/Desktop/libiec61850-0.9.0.2 (2)/examples/iec61850
nt_example1# ./client_example1
read float value: 7.999995
failed to write simpleIOGenericIO/GGIO1.NamPlt.vendor!
RptEna = 0
received report for simpleIOGenericIO/LLN0.RP.EventsRCB01
GGIO1.SPCS00.stVal: 0 (included for reason 4)
GGIO1.SPCS01.stVal: 0 (included for reason 4)
GGIO1.SPCS02.stVal: 0 (included for reason 4)
GGIO1.SPCS03.stVal: 0 (included for reason 4)
received report for simpleIOGenericIO/LLN0.RP.EventsRCB01
GGIO1.SPCS00.stVal: 0 (included for reason 5)
GGIO1.SPCS01.stVal: 0 (included for reason 5)
GGIO1.SPCS02.stVal: 0 (included for reason 5)
GGIO1.SPCS03.stVal: 0 (included for reason 5)
received report for simpleIOGenericIO/LLN0.RP.EventsRCB01
GGIO1.SPCS00.stVal: 0 (included for reason 4)
GGIO1.SPCS01.stVal: 0 (included for reason 4)
GGIO1.SPCS02.stVal: 0 (included for reason 4)
GGIO1.SPCS03.stVal: 0 (included for reason 4)
received report for simpleIOGenericIO/LLN0.RP.EventsRCB01
GGIO1.SPCS00.stVal: 0 (included for reason 4)
GGIO1.SPCS01.stVal: 0 (included for reason 4)
GGIO1.SPCS02.stVal: 0 (included for reason 4)
GGIO1.SPCS03.stVal: 0 (included for reason 4)

```

Figure 36. Terminal line, client_1 receiving reports from the server.

Figure 36 depicts the establishment of the communication link. The server and the client operate according to the configuration of the programmer. In the current implementation, the client receives reports from the server.

Figure 37 depicts the pinging of the server from the client's terminal. The acquired times are depicted in graphs and analyzed in the last sub-chapter.

```

64 bytes from 192.168.0.104: icmp_seq=55 ttl=64 time=0.051 ms
64 bytes from 192.168.0.104: icmp_seq=56 ttl=64 time=0.061 ms
64 bytes from 192.168.0.104: icmp_seq=57 ttl=64 time=0.051 ms
64 bytes from 192.168.0.104: icmp_seq=58 ttl=64 time=0.050 ms
64 bytes from 192.168.0.104: icmp_seq=59 ttl=64 time=0.054 ms
64 bytes from 192.168.0.104: icmp_seq=60 ttl=64 time=0.049 ms
64 bytes from 192.168.0.104: icmp_seq=61 ttl=64 time=0.059 ms
64 bytes from 192.168.0.104: icmp_seq=62 ttl=64 time=0.047 ms
64 bytes from 192.168.0.104: icmp_seq=63 ttl=64 time=0.062 ms
^[[B64 bytes from 192.168.0.104: icmp_seq=64 ttl=64 time=0.050 ms
64 bytes from 192.168.0.104: icmp_seq=65 ttl=64 time=0.054 ms
64 bytes from 192.168.0.104: icmp_seq=66 ttl=64 time=0.050 ms
^C
--- 192.168.0.104 ping statistics ---
66 packets transmitted, 66 received, 0% packet loss, time 64999ms
rtt min/avg/max/mdev = 0.024/0.048/0.069/0.012 ms
root@mike-ThinkPad-X230:~/Desktop/libiec61850-0.9.0.2 (2)/examples/iec61850_c
le1#

```

Figure 37. Pinging the address of the server from the client.

6.2. Introduction to Hamachi

According to *HAMACHI LogMeIn Getting Started Guide* (2016: 3-22), Hamachi is a virtual networking application, requiring no configuration; hence it can be set up in

minutes, enabling the secure remote access to the owner's network from anywhere an Internet connection exists. In other words, Hamachi establishes a virtual connection over the Internet, imitating the connection that would be if the computers were in a local network *LAN*. Hamachi is an on-demand service, being used according to the user's needs and it is compatible with Windows (Vista, XP, 7, 8, and 10), Mac OS 10.6, and Linux distributions (Ubuntu 16.04, CentOS 7.2).

A Hamachi client is functional only by using the *LogMeIn* ID, as a member of a *LogMeIn* account. Regarding the Hamachi virtual IP address, every client has a globally unique virtual IPv4 address in the 25.x.x.x range. Via this address client can be accessed from any other Hamachi network using at least one hub-and-spoke or a mesh network. There are three types of subscriptions to access the service; free, standard and premium. The free subscription is for networks incorporating till five devices and for a trial period of two weeks. For a larger number of devices, the user has to pay an annual fee in order to join or create a Hamachi network. The standard version allows an attaching of 32 clients/network while the premium 256 clients/network.

A few examples of Hamachi advantages are:

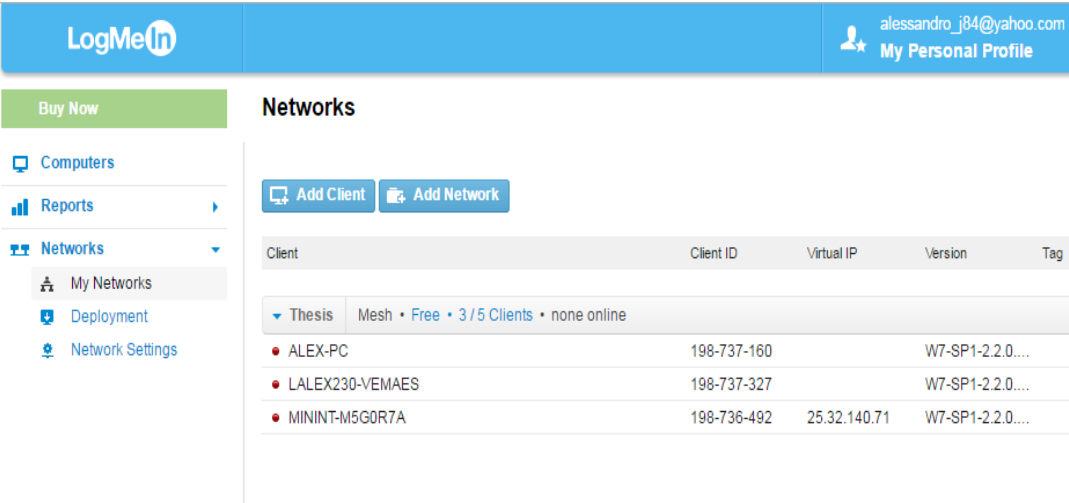
- *Extension of a Mobile Office LAN Network.* Mobile workers in a company will still have access to the company's resources (printers, mail servers, etc.); even they move away from the company's LAN. By the Hamachi application, mobile workers and all the shared resources become members of the same Hamachi network, communicating with each other, regardless of their physical location.
- *Access to Company's Network from Home.* Some workers are more productive when they work at home. Using the Hamachi as clients, home workers can establish a secure tunnel to their company's network, accessing the shared IT resources.
- *Managing Multiple Networks.* An administrator can set up and maintain many Hamachi networks for several customers. The *LogMeIn Central* can be used to create Hamachi networks, applying default or custom-based settings for each customer individually. The main administrator has the overall management over, clients, network activity, and other administrators in the group.

The network types that Hamachi provides are Hub-and-Spoke, Mesh, and Gateway. They distinguish mainly by the network topology.

- Regarding *Mesh networks*, every client/member is connected to every other client.
- Within *Hub-and-Spoke Networks*, one or more computers have the role of the hub, while clients connect as spokes. Spokes (clients) may connect to hubs, but not to each other. It is the ideal choice when strict control is required regarding the connection between network members.
- The *Gateway Network* can offer transparent access to the whole network via a centralized Hamachi gateway. In a gateway network, clients such as mobile workers can see one computer operating as a gateway for the entire LAN, providing access to all network resources.

The Hamachi's security is end-to-end. Two nodes exchange data only after mutual authentication and session key agreement. Regarding data encryption and decryption, the AES-256-CBC cipher is used after the establishment of a session key.

6.2.1. Installation and configuration of Hamachi



Client	Client ID	Virtual IP	Version	Tag
▼ Thesis Mesh • Free • 3 / 5 Clients • none online				
• ALEX-PC	198-737-160		W7-SP1-2.2.0....	
• LALEX230-VEAES	198-737-327		W7-SP1-2.2.0....	
• MININT-M5GDR7A	198-736-492	25.32.140.71	W7-SP1-2.2.0....	

Figure 38. The LogMeIn administration web page.

Before accessing the Hamachi services, a *LogMeIn* account needs to be created through the LogMeIn website. Next, it is possible the creation of a network and attachment of clients to it. The user he has the overall management of the network and the clients regardless of their physical location. The management includes edit or delete of an existing network, installation of the Hamachi client to a local or remote computer, and management of request from clients to join a Hamachi network.

Figure 38 depicts a free Hamachi subscription, with three active clients. The account was created in order to conduct the experiment, and it consists of one mesh network (*Thesis*), and three attached clients corresponding to three different computers. This interface provides all the services mentioned above. By *Add Client* the Hamachi software can be installed on the current computer or it can deploy to other (remote) computers. By *Add Network* we can create a Mesh, a Hub-to-Speak or a Gateway network. Via *edit the Network*, we can manage the join requests, change subscription, request a password for joining the network or delete the network.

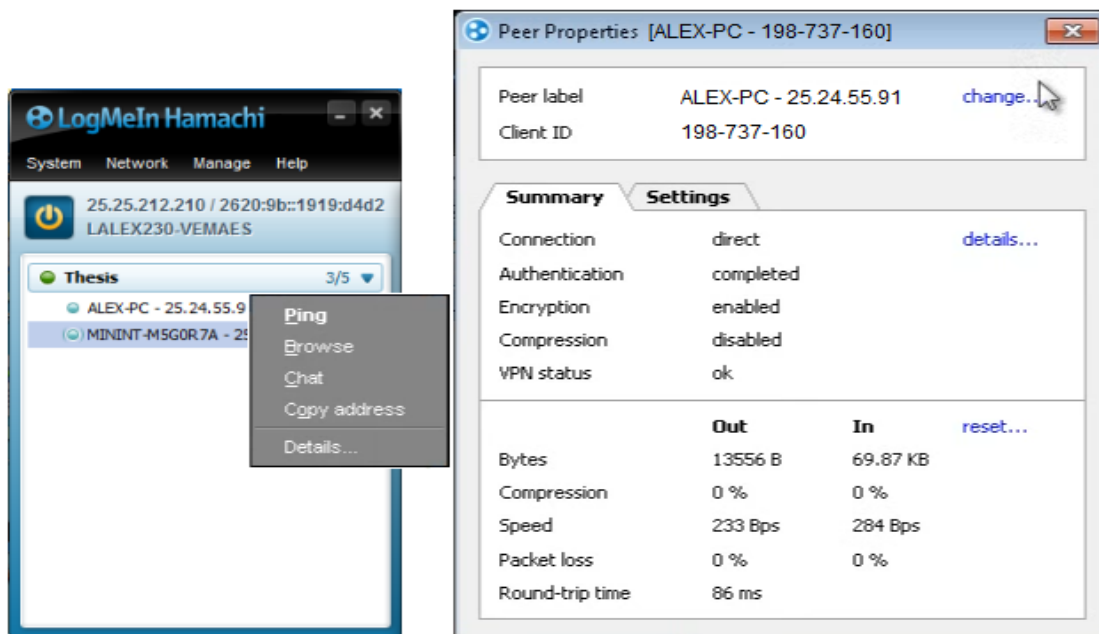


Figure 39. The LogMeIn Hamachi client.

Figure 39 depicts the Hamachi interface after the software was downloaded on the PC. It indicates the created network (*Thesis*), plus three clients with their names and their virtual IPs. The clients correspond to other PCs. The first one (*LALEX230-VEMAES*) is the administrator, while the other two joined the network by his permission. By pressing right click on a client, the administrator can choose to ping or to chat with it.

6.2.2. Analysis of the practical implementation

After the installation and creation of the Hamachi network, the practical part was ready to start. Two PCs were used running the Hamachi software. Once the Hamachi network (named *Thesis*) was established; the PCs were members of the same Hamachi network, and regardless of their physical location, they could communicate similarly as they were to the same local network. One of the PCs had the administrative role, while the other was a client joined the network by request to the administrator.

The ping command was executed from administrator to client or vice versa (figure 40). The acquired round-trip times are represented by graphs and analyzed in the last sub-chapter.

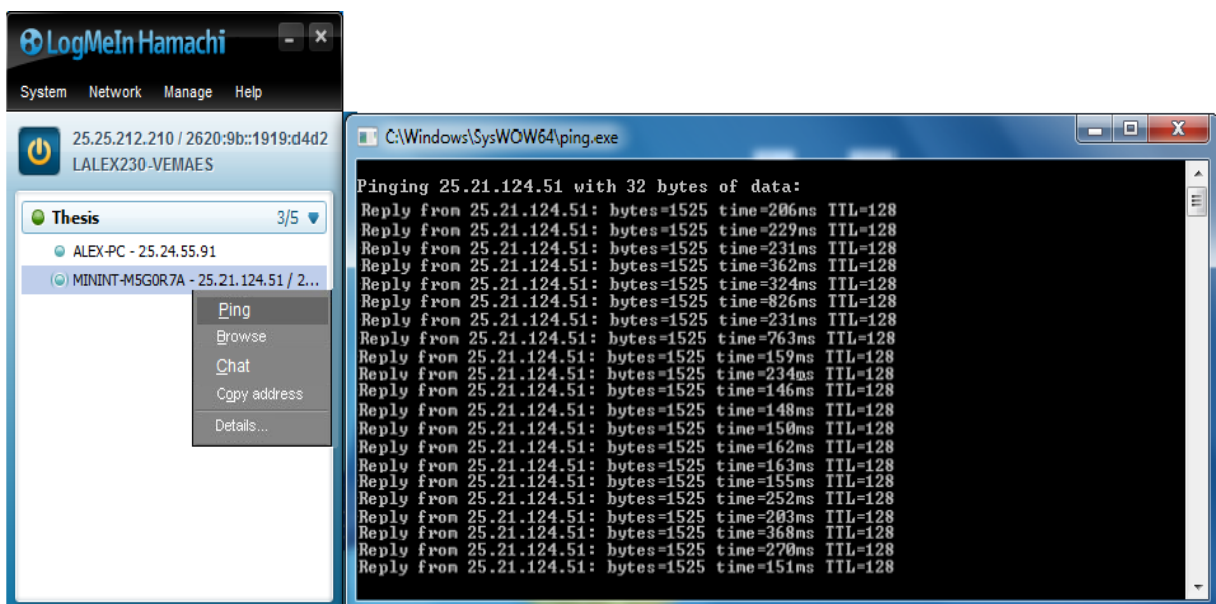


Figure 40. Pingging the Hamachi client

6.3. Introduction to Beagleboard.org

According to *BeagleBoard.org* (2016), the BeagleBoard is a non-profit foundation based in the United States, pursuing to provide education regarding embedded computing (design and use of open-source hardware and software). It emerged from a group of passionate individuals, including some Texas Instruments employees, interested in embedded devices.

The initial development was pursuing to improve the Linux distributions to support ARM devices. After the great success and support by many Linux distributions, development focused on enabling simplified physical computing on advanced GUI and networked devices with a simple learning experience and support of development environments such as Ubuntu, Windows Embedded, Android, etc.

All the designs are open-source and components can be provided by any manufacturer with compatible hardware. For example, nowadays the funding regarding board prototypes is provided by manufacturing partners such as Texas Instruments.

The low-cost produced boards are fan-less, single-board tiny computers based on Texas Instruments low-power processors featuring the ARM Cortex-A series core. Some of the Beagle boards/devices are: (BeagleBoard, BeagleBoard-X15, BeagleBoard-xM, BeagleBone, and BeagleBone-Black). The BeagleBoard.org provides a forum for hardware owners and open-source software developers to exchange ideas, knowledge, and experience.

The general-purpose processor of Beagles dominates versus the other commercial low-cost computing platforms. Some of the possible applications include low-cost Linux PC, robotics, home automation, network sniffer, vehicle telematics and automation, USB traffic monitor, gaming console, web services development, multimedia codec, security camera analyzer/streamer/recorder/monitor, framework development mobile digital TV, and many others.

6.3.1. Configuration of the BeagleBone-Black

The experiment was conducted via a BeagleBone-Black, which is a community-supported development board for developers or hobbyists. For the installation of the device, BeagleBoard website provides a step-by-step guide.

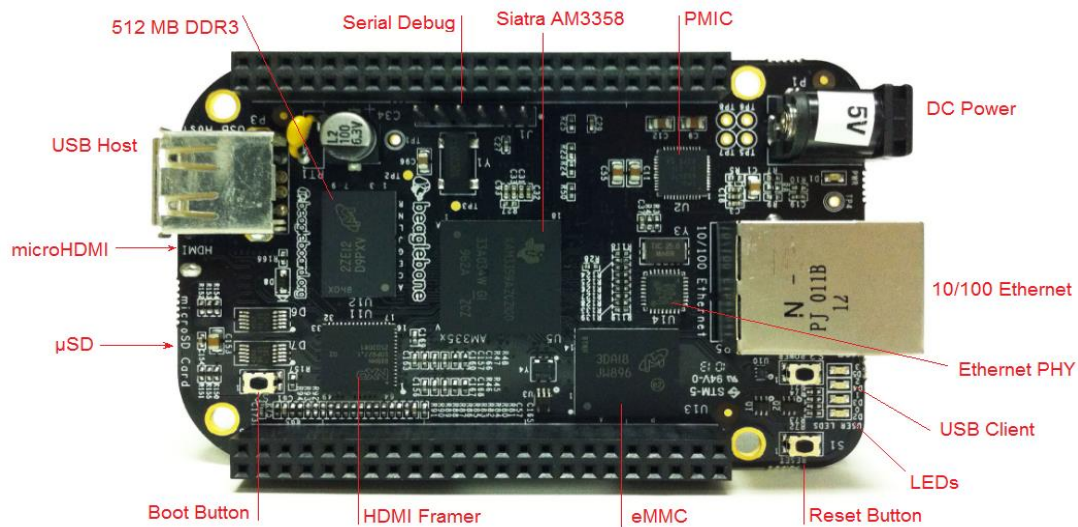


Figure 41. Beaglebone-Black platform. (BeagleBoard.org, 2016).

The BeagleBone-Black features are:

- *Processor:* AM335x 1GHz ARM Cortex-A8
 - 512MB DDR3 RAM
 - 4GB 8-bit eMMC on-board flash storage
 - 3D graphics accelerator
 - 2x PRU 32-bit microcontrollers
- *Connectivity:* USB client (for power & communications)/ USB host/ Ethernet/ HDMI or 2x 46 pin headers.
- *Software Compatibility:* Debian, Android, Ubuntu, plus much more.
- *Power Supply:* 210-460 mA/ 5V.

According to *BeagleBoard.org* (2016), a typical configuration requires the connection of the board to a PC via a USB-cable. So the PC provides the power and the developing interface for the Beagle. Once connected, the device will boot Linux operating system from the on-board 2GB or 4GB eMMC. Another option for booting is through the pre-configured microSD card.

Next step is to install the drivers for our operating system to provide us network-over-USB access to the Beagle. From the website we download drivers according to our PC's operating system (e.g. Window 64-bit /32-bit, Linux or MAC OS), and next, execute.

We click this address <http://192.168.7.2> on Chrome or Firefox browser in order to launch the page running on-board. The page shows the Beagles' capabilities. For navigation use the arrow keys on the keyboard.

From the left menu, we click on the *Cloud9 IDE*. That forwards us to an integrated developing environment running on the Beagle (Linux terminal). By typing *ifconfig*, we can see the eth0 IP address.

That was the initial installation/configuration as provided by the BeagleBoard.org. Once the user wants to use the platform for its own applications, he has to download the proper drivers and execute the proper commands in the terminal, making the platform fully operational.

- *Manipulating the IP routing table*: At this moment the Beagle cannot access the internet. So, we must provide the default gateway from the *sbin* subdirectory of our Ubuntu Linux:

```
/sbin/roote          # displays the current routing table
/sbin/roote add default gw 192.168.7.1 eth0 # IP-address and an interface name
nano/etc
nameserver 8.8.8.8
```

➤ *Configuring the 4G-Dongle:*

The following protocols need to be installed and make the dongle fully operational. The *PPP (Point to Point Protocol)* is used in the data link layer to establish a connection between two nodes. The *USB mode_switch* is used to convert the 4G dongle from the initial flash storage device into 4G modem. *Wvdial* is a Point-to-Point Protocol, dialing the modem and initiating PPPD to access the 4G Internet. The PPPD (Point-to-Point daemon protocol) is used to manage network connections between nodes on UNIX operating systems. To install them we use the below command:

```
sudo apt-get install ppp usb-modeswitch wvdial
sudo reboot      # Command for restarting the system
```

➤ *Changing the USB's ModeSwitch-program Configuration.*

```
# the lsusb command displays a list of the USB connected devices, (we check
# for the HUAWEI dongle and get its ID)
lsusb
cd /tmp # We enter the temporary folder
# We extract the file with various modems and the required information with
#the command below; the xxxx and yyyy denote the Dongle's ID number
tar -xzf /usr/share/usb_modeswitch/configPack.tar.gz xxxx \: yyyy
# the next command opens the unpacked file in a text editor; copy the
#TargetVendor and TargetProduct
leafpad xxxx yyyy
# the below command opens the usb_modeswitch.conf file
sudo Leafpad /etc/usb_modeswitch.conf.
# Paste the above copied lines in the new emerging window.
# Start the USB Mode Switch and change to the correct 4G modem mode.
```

- *Creation of a WvDial config file to connect our service provider (Elisa)*

The SIM card and the Dongle were provided by ELISA-mobile. At first, we check the SIM card in a mobile-phone and disable any PIN request if present.

Open the file containing the WvDial configuration data.

```
sudo leafpad /etc/wvdial.conf
```

#The emerging window requires several inputs such as USR/PSW, APN (mob.uwasa.fi in our case), and phone number provided by the operator

#Before connection we check the device to be in modem-mode.

```
sudo usb_modeswitch -c /etc/usb_modeswitch.conf
```

```
wvdial 3gconnect # The command to access the internet.
```

- *Obtaining the File's Permission*

UNIX systems incorporate a file control mechanism specifying who can access a particular file/folder and the ways they can act on it. The file control mechanism is divided into *classes* and *permissions*. Classes determine who can access a file, while permissions determine which kind of action the user can perform on file (read, write or execute). So, the following commands have to be entered through the command line in order all the used programs to acquire the rights to start the connection:

```
chmod +s /usr/sbin/pppd
```

```
chmod 777 /usr/sbin/pppd
```

```
chmod 777 /etc/ppp/chap-secrets
```

```
chmod 777 /etc/ppp/pap-secrets
```

```
chmod 777 /etc/ppp/peers/
```

```
chmod 777 /etc/ppp/peers/wvdial
```

```
chmod 777 /etc/ppp/peers/wvdial-pipe.
```

In the experiment, they have been used two Beagle platforms, and for both of them, the above configuration steps were followed in order to make them fully operational. The platforms were accessing the internet via a 4G modem, incorporating a SIM card

provided by Elisa Finland mobile company. For each modem was assigned a static IP to access the internet. Once the platforms were ready, they could ping to each other via the 4G internet. The second way of communication was through the Hamachi program which was running in two PCs; each controlling one of the Beagles. Both platforms joined the same Hamachi network so that they could ping each other over the virtual network established by the Hamachi program.

6.4. Analysis of the Results

For the graphical representation of the acquired times they have been used two functions of the Matlab software; the `hist(X)` and the `ecdf(X)`.

The `hist(X)` function creates a histogram depicting the distribution of the results over the time. The bars have a uniform width, while their height indicates the number of results in the bar.

The `ecdf(X)` function is called empirical CDF and indicates the proportion of X values less or equal to x . The plot of the `ecdf(X)` depicts each unique value versus the probability of values are less or equal to it. In other words, the `ecdf` plot is similar to a probability plot, but instead of a straight line, the fitted distribution forms a staircase-like function. We can use the plot to assess the distribution of data, estimate percentiles or compare different distribution-graphs with each other.

➤ GOOSE round-trip times based on the implementation of the libiec61850 library

The library was running on two PCs, one representing the client and the other the server. The PCs were members of the same network, and they were connected through a switch, so that's why the acquired round-trip times are very small (figure 42). Theoretically, the less the intermediate nodes, the smaller is the round trip

time, although the round-trip time is mostly affected by the proceeding taking place in the publisher and receiver of the GOOSE message. So, when the PC is running simultaneously other applications, it might have an impact at the round-trip time.

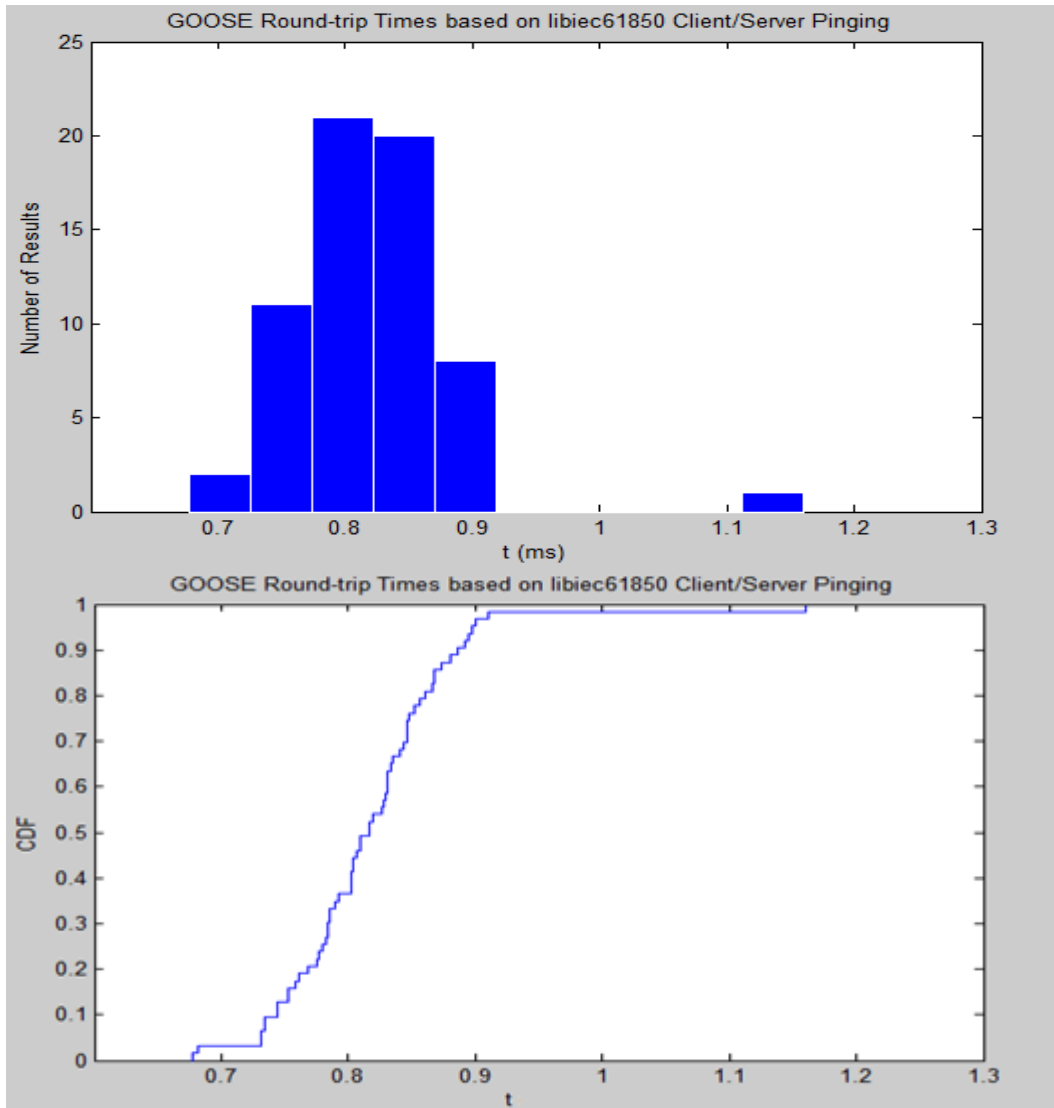


Figure 42. GOOSE round-trip times by the implementation of the libiec61850 library.

Looking at the histogram, it obviously approaches a normal distribution, with the majority of samples (times) to be within 0.8-0.9 ms. From the ecdf graph, we can see that around 98% of the samples (times) ≤ 0.9 ms.

- GOOSE round-trip times based on the PC-to-PC pinging via Hamachi

Looking at the histogram, we can see that the majority of samples (times) are equal or less to 240 ms. Analyzing the ecdf we can see that 70% of the round-trip times are ≤ 200 ms, while around 90% are equal less to 300 ms.

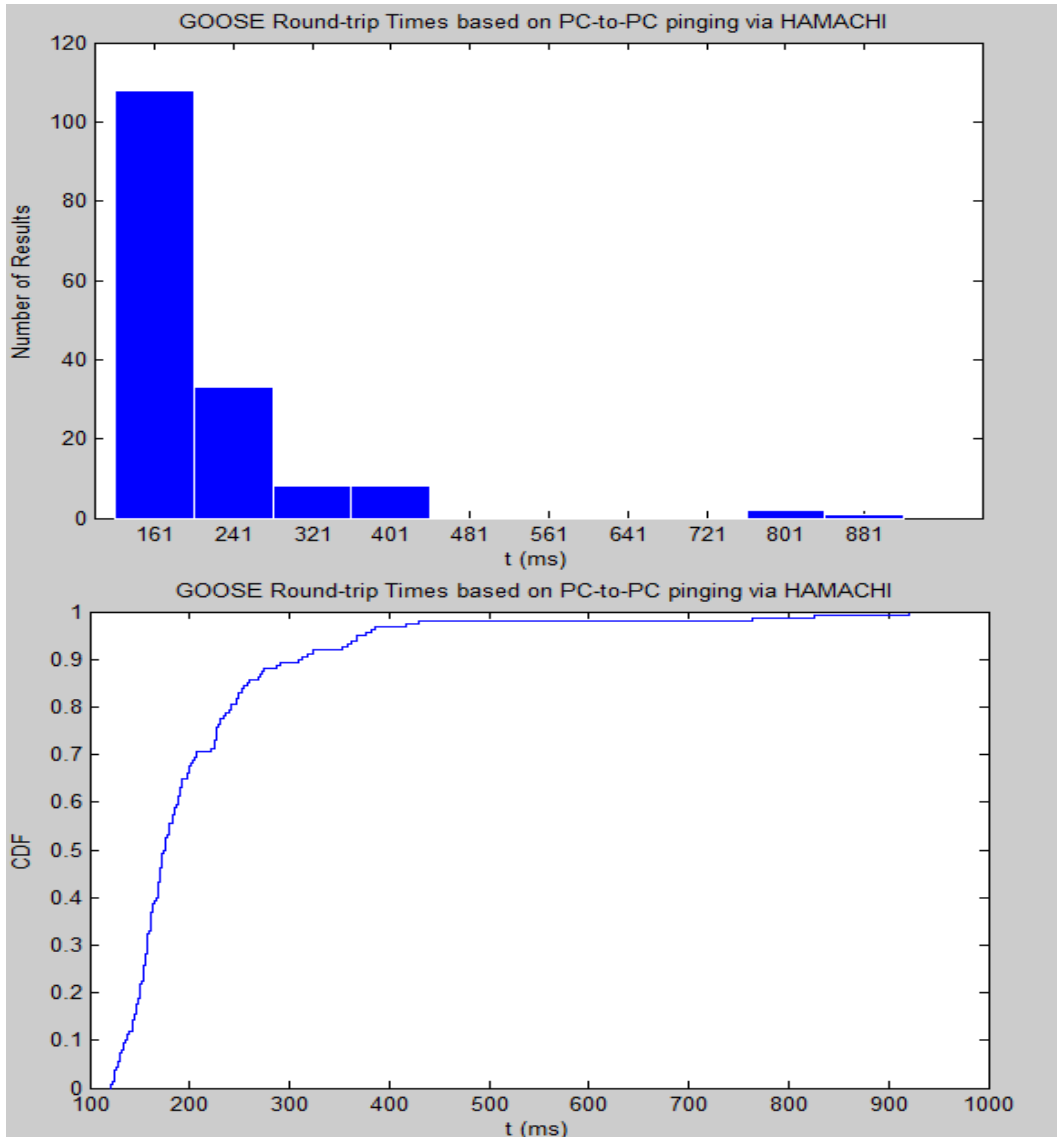


Figure 43. GOOSE round-trip times for PC-to-PC pinging via Hamachi.

Although the table 3 on page 52 regarding communication requirements depicts an individual case, we can use it to compare our results. Regarding wide area

communications, the table depicts a range of 50-500 ms end-to-end latency. Roughly speaking, this means a round-trip time in the range 100-1000 ms.

In our case, the acquired times are within this range, meaning that the Hamachi-based tunneling has good perspectives to be used for wide area communication. Nevertheless, from the table we cannot certainly say what types of messages this range concerns; low, medium, or high-speed messages.

- GOOSE round-trip times based on the Beagle-to-Beagle pinging via 4G modem

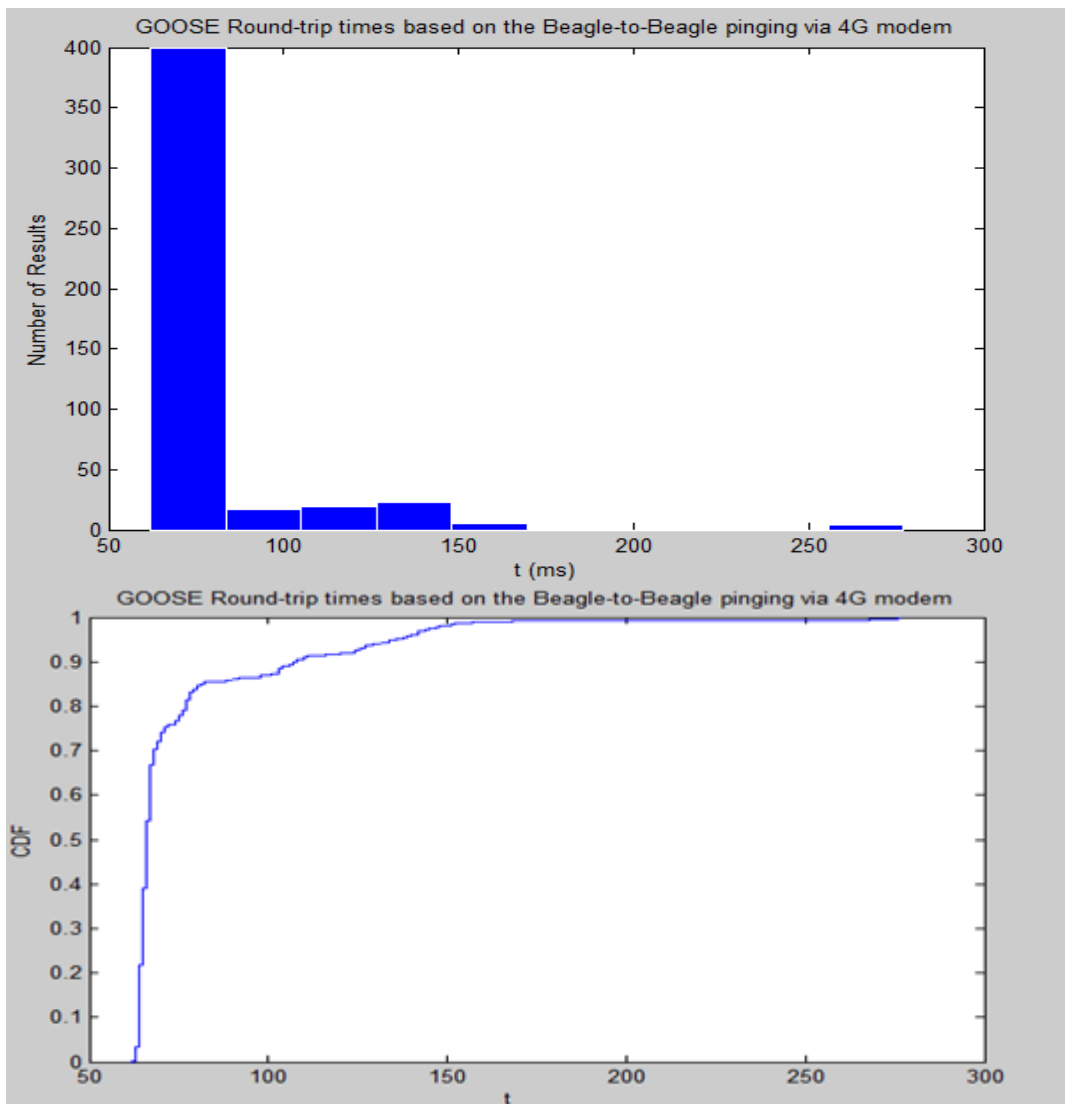


Figure 44. GOOSE round-trip times for Beagle-to-Beagle pinging via 4G modem.

Looking at the histogram, we can see that the majority of samples (times) are equal less to 100 ms. Analyzing the ecdf we can see that 85% of the round-trip times are equal less than 75 ms, and around 98% fall below 150 ms.

The times are within the range of the table-3 regarding wide area communication. In addition, it is obvious that from the particular experiment we acquired the fastest round trip times, meaning that packets “*travel*” faster in comparison to other methods. Additionally, the plot approaches better the characteristics of an ideal network, since the graph of an ideal network is a vertical line, set to a specific time!

The higher speeds can be justified partly to the better performance of the 4G mobile network in comparison to the tunneling method. Additionally, the Beagle platform is an embedded system, and it has several advances in comparison to a PC. An embedded system handles only a few tasks at a moment, or it is designed for a specific task, contrary to a PC that may run tenths of processes at a moment and probably affecting the round trip time.

➤ GOOSE round-trip times based on the Beagle-to-Beagle pinging via Hamachi

At first glance, it is obvious that samples (times) are uniformly distributed within the 100-860 ms range (Figure 45). It means that some of the packets are transferred very fast while some others very slowly.

The combination of Hamachi and Beagle did not have the expected results. It is the worst case from the three examined. Around 62 % of samples are equal less than 500 ms, while around 90% fall below 800 ms; this is translated into very large round-trip times, beyond the range indicated in the table_3.

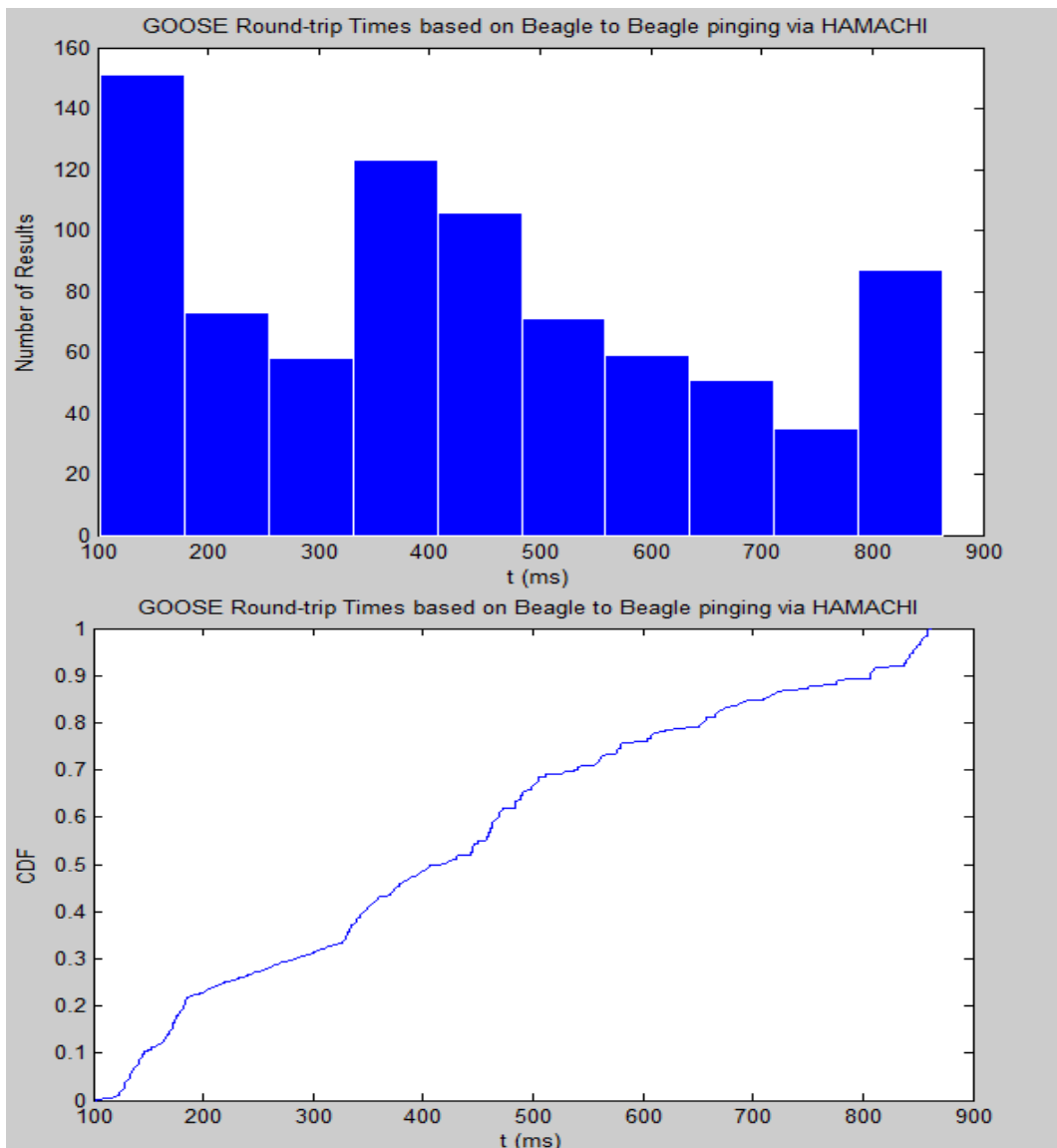


Figure 45. GOOSE round-trip times for Beagle-to-Beagle pinging via Hamachi.

The main reason that could lead to these high times is that in the concrete experiment they were added two more PCs during the communication. The communication path had the form $\text{Beagle1} \rightarrow \text{PC1} \rightarrow \text{Hamachi Tunneling} \rightarrow \text{PC2} \rightarrow \text{Beagle2}$. It is obvious that the more the nodes, the more the delay! So the addition of the PCs had as result in the large increment of the round-trip times; for some messages the time it was 10-fold larger in comparison to the previous methods.

Another reason could be the temporary congestion of the VPN network, especially when the service is provided free of charge (our case). Additionally, the uploading

speed in a VPN connection is equal to the uploading speed of the worst node in the path, and this negative feature might also affect the round trip time.

➤ Comparison of the three methods

From the plot (figure 46), it is obvious that the Beagle-to-Beagle pinging via 4G modem offered the fastest message transmission. So the communication over the 4G mobile network can offer higher speeds outmatching from the tunneling method. Regardless of that, the current technology cannot be used to transmit routable-GOOSE messages, since they are defined theoretically in the 90-5 part, but in practice, they do not exist devices/IEDs to support them.

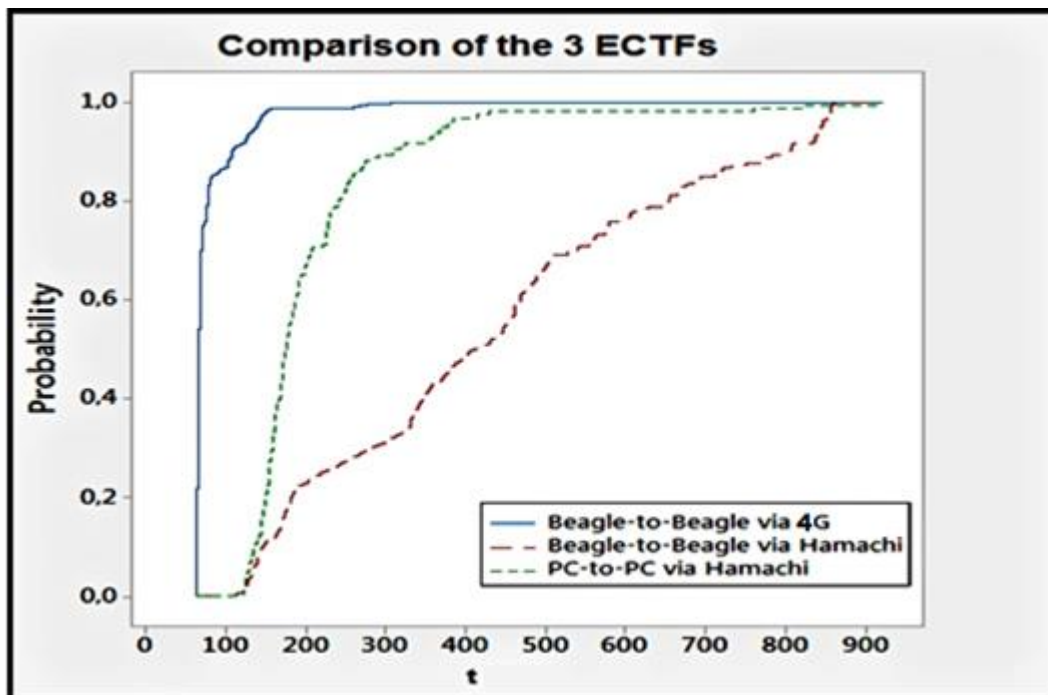


Figure 46. Comparison plot of the three methods.

The PC-to-PC pinging via Hamachi tunneling also offered quite satisfactory results, although PCs except for the particular task run as well and other processes adding extra processing latency to transmission. At the moment the tunneling consists an ideal method for wide area communication since the current IEDs cannot support

the routing of the Ethernet-based GOOSE messages. On the contrary, IEDs can be connected via a tunnel and communicate, offering substantially indirect routing.

The Beagle-to-Beagle communication based on the 4G network for establishing the Hamachi tunnel had the lowest performance in comparison to the other methods. A method to improve the performance could be if the platforms were able to establish a tunnel without the contribution of the PCs.

6.5. Argumentation, Future Work & Optimization

From the acquired times we can see that both methods offered round-trip times within the range provided by the example on the Table_3 (100-1000ms). Nevertheless, since real IEDs cannot access the internet directly, the tunneling method is the ideal solution at the moment for wide area communication. So, via tunneling, it was achieved an indirect routing of the Ethernet-based GOOSE and SV messages.

After the accomplishment of the practical part, the conclusion is that the routing and tunneling can be generally used to exchange messages over wide area networks. Nevertheless, the range provided by the Table_3 regarding wide area communication is not enough to specify if this communication concerns also the exchange of protection messages since they have very strict time requirements

The optimization steps proposed below can contribute to better results for future experiments concerning IEC 61850 communication.

Since each experiment incorporates several steps and many types of applicable devices/software; there are several ways to optimize the acquired results. Below are proposed several optimization steps for similar experiments.

- The pinging time was used for the sake of simplicity, assuming that it is equal to the round-trip time. Although it approaches the round-trip time, it is not exactly the

same, so instead of pinging, the Omicron IEDScout program should be used to generate real GOOSE messages.

- The Omicron IEDScout is the ideal tool for the simulation of GOOSE messages. It can generate GOOSE messages, sniffing the traffic in real time and depicting the message content. We must keep in mind that GOOSE cannot be routed; hence the test can only be applied within the local network's limits.
- A network analyzer program should be used (e.g. Wireshark) to capture the transmitted GOOSE messages. Via a network analyzer, it is possible the accurate measurement of the round-trip times from the analyzed messages.
- There are several types of VPN, (IPsec, L2TP, PPTP, SSL, OpenVPN, SSH), having their pros and cons. So, further investigation is required to reveal which one provides the highest speeds, maintaining in parallel the required security. For example, Hamachi service uses the PPTP protocol which has several security vulnerabilities; hence for real applications, this fact should be taken seriously into consideration.
- Additionally, free VPN services should be avoided since they do not offer the best performance, and in the case of a congested network could add a lot of latency.
- The use of an embedded system especially designed for a particular experiment would be beneficial, since the processing will focus on the execution of the specific task, leading to optimum performance. For example a platform capable of establishing or running a VPN connection without the help of a PC.
- Once an engineer has the programming skills (C or Java), he can use the libiec61850-0.9.0.2 library to build his own applications according to the requirements of the project. The library except for PCs is meant to be used mostly for embedded systems, hence an application especially designed to run on an embedded platform will avoid the possible latency added when it runs on a PC.

7. CONCLUSIONS

It is predicted that the IEC 61850 will be the dominant standard to undertake the communication aspects of smart grids, supporting in parallel the concept of the decentralized energy production. The above statement means that the standard has to surpass the substation's limits, to support wide-area communication via the *tunneling* or by the addition of *routable* profiles to the Ethernet-based messages.

The fundamental aim of this thesis was to investigate the performance of the *tunneling* and *routing* methods suggested by the 90-1 and 90-5 parts of the IEC 61850 standard for extension of the communication beyond the substation's limits.

The content of the thesis may be divided into theoretical and practical part: The theoretical part provides an overview of the main literature of the IEC 61850 standard, the history, objectives, benefits, concepts, as well as already implemented communication protocols (IEEE/IEC/ISO) related to the IEC 61850. Emphasis is given to the standard's communication architecture, focusing on the 90-5 part, meant to provide routable profiles for GOOSE and SV messages. The practical part consists of the methodology followed to conduct the experiment, presentation of the devices and software being used as well as their configuration.

The acquired round trip times from the practical implementation were depicted in Matlab plots, used to compare and evaluate the Hamachi/VPN and the BeagleBone/4G for their performance. The reference point for the comparison was the time range provided by the Table_3. From the two methods, the 4G mobile network offered the fastest communication in comparison to the tunneling, although both methods provided round-trip times within the range mentioned in the Table_3. So, taking into considerations the drawbacks of the applied methods, and the assumptions being made, we came to the conclusion that the routing and tunneling can be generally used to exchange messages over wide area networks. Nevertheless, several optimization steps were proposed, such as the use of IEDScout program for generating real GOOSE messages, while a network analyzer should be used for an accurate round trip time measurement.

LIST of REFERENCES

- Adamiak, Mark, Baignet, Drew, Mackiewicz, Ralph Zhangand (2004). *IEC 61850 Communication Networks and Systems in Substations: An Overview for Users*, SIPSEP, Monterrey, Mexico. 61-68 p.
- BeagleBoard.org. *BeagleBoard.org*. [online]. [Cited 26 November 2016]. Available from World Wide Web: <<https://beagleboard.org/>>.
- Comer E., Douglas (2000). *Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture*. 4th Ed. New Jersey, USA. 133-134 .p ISBN 0130183806.
- DEMVE Training Material (2014). *IEC 61850 Horizontal Communication*. Vaasa University of Applied Sciences (VAMK) 2014.
- D-Link User Manual (2013). D-Link DWR-116 Wireless N300 Multi-WAN Router. 133-134 p.
- Gunter A., Zhangand (2009). *IEC 61850 - Communication Networks and Systems in Substations: An Overview of Computer Science*. University of Illinois. p.6 Urbana-Champaign.
- Hamachi(2016). *LogMeIn HAMACHI Getting Started Guide*.
- IEC 61850 (2003-2012). *Communication networks and systems in substations*. IEC 61850 Standard, (10 Parts).

IEC 61850-90-1 (2010). *Use of the IEC 61850 for the communication between substations.*

IEC 61850-90-5 (2012). *Use of the IEC 61850 to transmit synchrophasor information according to IEEE C37.118.*

IEC (2014). International Electrotechnical Commission. *Electropedia: The World's Electrotechnical Vocabulary* [online]. [Cited 23 March 2014]. Available from World Wide Web: < URL:<http://www.electropedia.org/?ref=extfooter>>.

IEEE C37.118 (2005). *IEEE Standard for Synchrophasor Data Transfer for Power Systems.*

IT Wissen. *Das große Online-Lexikon für Informationstechnologie.* [online]. [Cited 01 November 2016]. Available from World Wide Web: < URL: <http://www.itwissen.info/definition/lexikon/transmission-control-protocol-internet-protocol-TCP-IP-TCP-IP-Protokolle.html/>>.

LibIEC61850. *Open source library for IEC 61850.* [online]. [Cited 08 August 2016]. Available from World Wide Web: <URL:<http://libiec61850.com/libiec61850/>>.

NettedAutomation. *The Net is the Automation.* [online]. [Cited 27 November 2016]. Available from World Wide Web: < URL: <http://www.nettedautomation.com/solutions/iec61850/IEC61850-SV-Application.pdf> >.

OpenVPN Technologies. *Your private path to access network resources and services securely.* [online]. [Cited 02

November 2016]. Available from World Wide Web: < URL: <https://openvpn.net/index.php/open-source.html/>>.

PAC World Magazine. *Performance Measurements for IEC 61850 IEDs and Systems*. [online]. [Cited 09 December 2016]. Available from World Wide Web: < https://www.pacw.org/issue/december_2010_issue/performance/implementing_firewalls_for_modern_substation_cybersecurity.html>.

PAC World Magazine. *IEC 61850 Edition 2 and Engineering*. [online]. [Cited 09 December 2016]. Available from World Wide Web: <https://www.pacw.org/en/issue/december_2014_issue/iec_61850_edition_2_and_engineering/maintenance_testing_program_choices_tbm_cbm_and_pbm/complete_article/1.html>.

Phasor-RTDMS. *Phasor Real Time Dynamics Monitoring System*. [online]. [Cited 05 January 2016]. Available from World Wide Web: < URL: http://www.phasor-rtdms.com/phasorconcepts/phasor_adv_faq.html >.

Pinto Faria, Ricardo André (2011). *A Wireless Sensor Network for Electrical Distribution Substations*. Lisbon, Portugal: Universidade Tecnica de Lisboa. Master Thesis. 5 p.

Siemens (2010). *Efficient Energy Automation with the IEC61850 Standard 2 p*.

Taikina-aho, Markku (2011). *Redundant IEC61850 Communication Protocols in Substation Automation*. Vaasa, Finland:

Vaasa University: Electrical and Energy Engineering.
Master Thesis. p. 42.

Tech-FAQ. *The OSI Model - What It Is; Why It Matters; Why It Doesn't Matter*. [online]. [Cited 29 October 2016]. Available from World Wide Web: < URL: <http://www.tech-faq.com/osi-model.html>>.

TechTarget. *TechTarget Global Network of Technology-Specified Websites*. [online]. [Cited 25 October 2016]. Available from World Wide Web: < URL: <http://www.techtarget.com/network/>>.

Viola Systems (2015). *Application Note for 3G/LTE Gateway Configuration Guide. 8-80 .p*

Yeh, T.-H., Hsu, S.-C., Chung, C.-K. and Lin, M.-S. (2015). Conformance Test for IEDs Based on IEC 61850 Communication Protocol. *Journal of Power and Energy Engineering*, 3, .p 289-296.