

TRABAJO ESPECIAL DE GRADO

DISEÑO DE UN PROTOTIPO DE PROTECCIÓN Y TELEPROTECCIÓN DE LÍNEA DE TRANSMISIÓN DE ULTRA ALTA TENSIÓN BAJO EL ESTÁNDAR UNIVERSAL DE SUBESTACIONES ELÉCTRICAS IEC 61850

Presentado ante la Ilustre
Universidad Central de Venezuela
Para optar al título de Ingeniero Electricista
Por Pablo Eduardo Romero

Caracas, 2016

TRABAJO ESPECIAL DE GRADO

**DISEÑO DE UN PROTOTIPO DE PROTECCIÓN Y
TELEPROTECCIÓN DE LÍNEA DE TRANSMISIÓN DE ULTRA
ALTA TENSIÓN BAJO EL ESTÁNDAR UNIVERSAL DE
SUBESTACIONES ELÉCTRICAS IEC 61850**

TUTOR INDUSTRIAL: Ing. Luís Betancourt
PROFESOR GUÍA: Ing. Dan El Montoya

Presentado ante la Ilustre
Universidad Central de Venezuela
Para optar al título de Ingeniero Electricista
Por Pablo Eduardo Romero

Caracas, 2016

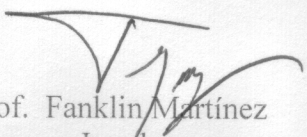
CONSTANCIA DE APROBACIÓN

Caracas, 08 de junio de 2016

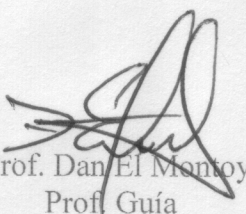
Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Pablo E. Romero O., titulado:

**“DISEÑO DE UN PROTOTIPO DE PROTECCIÓN Y TELEPROTECCIÓN
DE LÍNEA DE TRANSMISIÓN DE ULTRA ALTA TENSIÓN BAJO EL
ESTANDAR UNIVERSAL DE SUBESTACIONES ELÉCTRICAS IEC 61850”**

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Comunicaciones, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.


Prof. Franklin Martínez
Jurado


Prof. Francisco Varela
Jurado


Prof. Daniel Montoya
Prof. Guía

DEDICATORIA

A mis abuelos Eduardo Oestreicher y Hebe García Oestreicher
en honor a nuestra promesa.

A mi esposa e hijos, Natalia Medrano, Ailén y Nahuel Romero
por su apoyo y tiempo incondicional.

AGRADECIMIENTOS

A mis padres, Diana Mirtha Oestreicher y Eduardo Abel Romero, que siempre confiaron en mí.

A mis hermanas: Ingrid y Astrid Romero.

A toda mi familia.

A los ingenieros de la familia que me dieron todo su apoyo: Mary Luz Alonso, Michele Vicario (Padre e hijo), Pilar Medrano, Rafael Arruebarrena, Sergio Medrano

A mis compañeros de carrera, Jhonny Rinaldi, Gabriel Gil, Rafael Ortiz.

A mis tutores industrial y académico; José Luis Betancourt y Dan El Montoya.

A mis compañeros de trabajo, por su lealtad incondicional.

Pablo Eduardo Romero

**DISEÑO DE UN PROTOTIPO DE PROTECCIÓN Y
TELEPROTECCIÓN DE LÍNEA DE TRANSMISIÓN DE ULTRA
ALTA TENSIÓN BAJO EL ESTÁNDAR UNIVERSAL DE
SUBESTACIONES ELÉCTRICAS IEC 61850**

**Prof. Guía: Ing. Dan El Montoya. Tutor Industrial: Ing. José Luís Betancourt.
Tesis. Caracas, UCV, Facultad de Ingeniería Eléctrica. Ingeniero Electricista.
Opción: Comunicaciones. Institución: Corpoelec, 2016, 98h. +anexos**

Palabras Claves: Teleprotección; IEC 61850; GOOSE.

Resumen. Se realiza el diseño de un prototipo de protección y teleprotección de línea de transmisión de ultra alta tensión, siguiendo el estándar universal para subestaciones eléctricas especificado en la norma IEC 61850. Se simula su funcionamiento ubicándolo en la subestación La Arenosa, a la espera de un código permisivo para desconexión de carga por contingencia en la línea a 765 kV, específicamente en el vínculo: San Gerónimo – La Arenosa. Con la ocurrencia de un evento, los Dispositivos Electrónicos Inteligentes (IED) ubicados en la subestación San Gerónimo, cuya función es sensar valores de corriente, tensión y reactivos de la línea, notifican al servidor GOOSE que realice el envío de mensajes de alta prioridad al prototipo ubicado en la subestación La Arenosa y este, a su vez, ejecute una desconexión selectiva de carga según la información contenida en el mensaje. Todo el procedimiento no debe exceder el tiempo máximo exigido para la maniobra, tomando en cuenta los retardos de procesamiento de los IED involucrados, el medio de transmisión, el retardo del prototipo y tiempo de apertura del relé. En el análisis de resultados se comprueban empíricamente los cálculos obtenidos en la teoría, relativos al tiempo de transmisión.

ÍNDICE GENERAL

CONSTANCIA DE APROBACIÓN.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTOS.....	v
ÍNDICE GENERAL.....	vii
LISTA DE FIGURAS.....	x
LISTA DE TABLAS.....	xii
SIGLAS.....	xiii
ACRÓNIMOS.....	xiv
INTRODUCCIÓN.....	1
CAPÍTULO I.....	4
PLANTEAMIENTO DEL PROBLEMA.....	4
JUSTIFICACIÓN.....	5
OBJETIVOS.....	6
ANTECEDENTES.....	7
CAPÍTULO II.....	8
GENERALIDADES Y ASPECTOS TEÓRICOS.....	8
2.1. Descripción general de la línea troncal de transmisión de 765 kV.....	8
2.1.1. Esquema general de una subestación de ultra alta tensión.....	9
2.1.2. Sistema de protecciones.....	10
2.1.2.1. Sistemas de protección de líneas de transmisión.....	11
2.1.2.2. Sistemas de protección a interruptores.....	13
2.1.3. Causas de funcionamiento anormal en los sistemas de potencia.....	13
2.2. Descripción del estándar IEC 61850.....	14
2.2.1. Arquitectura de la subestación.....	14
2.2.2. Niveles e Interfaces Lógicas.....	15
2.2.3. Nodos lógicos (LN) y Funciones.....	17
2.2.4. Modelo jerarquizado de información.....	17
2.2.5. Clases genéricas de datos.....	18

2.2.6. Dispositivo Lógico.....	18
2.2.7. Modelo de Servicios Abstractos de Comunicación (ACSI).....	19
2.2.8. Eventos Genéricos de la Subestación (GSE).....	19
2.2.9. Lenguaje de configuración (SCL).....	22
2.2.10. Modelo (SCL).....	22
2.2.11. Descripción de los tipos de archivo SCL.....	23
2.2.12. Estructura del estándar.....	24
2.3. Prototipo de Teleprotección Remoto (TPR).....	25
2.3.1. Arduino UNO R3.....	25
2.3.2. Ethernet Shield Arduino.....	30
2.3.3. Entorno de Desarrollo Integrado (IDE).....	33
2.3.4. Estructura de un programa.....	34
2.4. Dispositivo Electrónico Inteligente (IED) simulado.....	35
2.5. Herramientas para el análisis de información.....	36
2.5.1. libIEC61850.....	36
2.5.2. Wireshark (Analizador de protocolos Ethernet).....	37
CAPÍTULO III.....	39
MODELADO Y DISEÑO DEL PROTOTIPO.....	39
3.1. Alcance y requerimientos.....	39
3.2. Modelo conceptual.....	39
3.3. Diseño del prototipo TPR.....	41
3.3.1. Hardware.....	42
3.3.2. Software.....	45
3.4. Análisis de tramas Ethernet tipo GOOSE bajo la norma IEC 61850.....	45
3.4.1. Desarrollo de algoritmos para publicación de mensajes GOOSE.....	51
3.4.2. Desarrollo de algoritmos para la interpretación de tramas en el Prototipo.....	57
3.4.3. Herramienta de monitoreo.....	67
CAPÍTULO IV.....	70
SIMULACIÓN Y ANÁLISIS RESULTADOS.....	70
4.1. Equipos utilizados.....	71
4.2. Esquema de montaje.....	71
4.3. Procedimiento.....	72
4.4. Fiabilidad del Prototipo.....	82
4.4.1. Muestreo en diferentes condiciones de espacio y tiempo.....	82
4.4.2. Muestreo con presencia de tráfico en la red.....	83

CONCLUSIONES.....	86
RECOMENDACIONES.....	87
REFERENCIAS BIBLIOGRÁFICAS.....	88
BIBLIOGRAFÍAS.....	90
ANEXOS.....	92

LISTA DE FIGURAS

Figura 1: Mapa de Subestaciones a 765 kV.....	8
Figura 2: Unifilar Sistema de Transmisión a 765 kV.....	9
Figura 3: Sección de una subestación de ultra alta tensión.....	10
Figura 4: Niveles SAS.....	16
Figura 5: Modelo abstracto de SAS.....	19
Figura 6: Priorización en la transmisión de mensajes tipo GOOSE.....	20
Figura 7: Tiempos de actuación de los mensajes.....	20
Figura 8: Desempeño de los mensajes en la pila de protocolos.....	21
Figura 9: Estructura del estándar.....	24
Figura 10: Esquemático de la placa Arduino UNO R3.....	26
Figura 11: Descripción de los pines de la placa Arduino UNO R3.....	27
Figura 12: Componentes principales de la placa Arduino UNO R3.....	28
Figura 13: Diagrama de bloques del AVR.....	28
Figura 14: Registros Hexadecimales.....	29
Figura 15: Circuito Ethernet Shield.....	30
Figura 16: Descripción de los pines de la placa Ethernet Shield.....	31
Figura 17: Trama UDP.....	32
Figura 18: Funciones del microchip WZ5100.....	33
Figura 19: Interfaz de usuario (GUI).....	35
Figura 20: Flujo de tramas en el tiempo.....	37
Figura 21: Analizador de protocolos Wireshark.....	38
Figura 22: Diagrama funcional del Sistema de Desconexión de Carga del SEN....	40
Figura 23: Esquema de actuación en el marco del unifilar del SEN.....	41
Figura 24: Placa Arduino UNO R3.....	43
Figura 25: Tipos de memoria en la placa Arduino UNO R3.....	43

Figura 26: 16 Relay Module.....	44
Figura 27: Detalle de una trama GOOSE.....	47
Figura 28: Jerarquía de objetos.....	52
Figura 29: Tiempos de transmisión de mensajes GOOSE.....	53
Figura 30: Representación coma flotante de 32bits, IEEE 754.....	54
Figura 31: Pila de protocolos sobre el modelo de capas OSI.....	58
Figura 32: Monitor Serie.....	60
Figura 33: Estructura de una trama GOOSE.....	62
Figura 34: Detalle de Ethertype.....	62
Figura 35: Secuencia de datos del APDU.....	64
Figura 36: Esquema de conexión del prototipo TPR.....	66
Figura 37: Algoritmo de captura de tramas GOOSE.....	67
Figura 38: Interfaz de monitoreo para la simulación.....	68
Figura 39: Comparación de los dataset en tramas GOOSE.....	69
Figura 40: Esquema general de montaje.....	72
Figura 41: Ráfagas de tramas GOOSE.....	73
Figura 42: Activación de la Línea.....	74
Figura 43: Flujo de Tramas.....	74
Figura 44: Detalle de la Trama.....	75
Figura 45: Definición del tiempo de transmisión (tomado de IEC 61850-2).....	77
Figura 46: Resultados de la Simulación.....	79
Figura 47: Gráfica de dispersión de la muestra.....	83
Figura 48: GUI herramienta de software PackETH.....	84
Figura 49: GUI herramienta de software PackETH.....	85

LISTA DE TABLAS

Tabla 1: Grupos de Nodos Lógicos.....	18
Tabla 2: Especificaciones técnicas placa Arduino UNO R3.....	26
Tabla 3: Estructura de una trama GOOSE.....	47
Tabla 4: Valores para Ethertype.....	62
Tabla 5: Tiempos de las ráfagas de las tramas.....	76
Tabla 6: Tipo de mensaje y de las tramas.....	78
Tabla 7: Análisis de dispersión de la muestra.....	82

SIGLAS

- CORPOELEC: Corporación Eléctrica Nacional, S. A.
- SEN: Sistema Eléctrico Nacional
- S/E : Subestación
- SRG: Subestación San Jerónimo
- ARE: Subestación La Arenosa

ACRÓNIMOS

ACSI:	Abstract Communication Service Interface
ALU:	Unidad Aritmético Lógica
APDU:	Application Protocol Data Unit
AVR:	Advanced Virtual RISK
BER:	Bit Error Rate
BSD:	Berkeley Software Distribution
CID:	Configured IED Description
EEPROM:	Electrically Erasable Programmable Read-Only Memory
FTDI:	Future Technology Devices International
GOOSE:	Generic Object Oriented Substation Event
GPL:	General Public License
GSE:	Eventos Genéricos de la Subestación
GSSE:	Generic Substation Status Event
GUI:	Graphic User Interface
HMI:	Human Machine Interface
I/O:	Input/Output
IDE:	Integrated Development Environment
IEC:	International Electrotechnical Commission
IED:	Intelligent Electronic Device
IID:	Instantiated IED description
ISO:	International Organization for Standardization
LAN:	Local Area Network
LD:	Logical Device
LN:	Logical Node
MAC:	Media Access Control

MMS:	Manufacturing Message Specification
OSI:	Modelo de Interconexión de Sistemas
PLC:	Power Line Carrier
PTPR:	Prototipo de Teleprotección Remota
SAS:	Substation Automation System
SCD:	Substation Configuration Description
SCD:	Substation Configuration Description
SCL:	Substation Configuration description Language
SCL:	Substation Configuration Language
SED:	System Exchange description
SRAM:	Static Random Access Memory
SRV:	Servidor
SSD:	System Specification Description
SV:	Samples Values
TCP/IP:	Transfer Control Protocol / Internet Protocol
UDP:	User Datagram Protocol
VID:	Identificador de Red Virtual
WAN:	Word Area Network
XML:	Extended Mark-up Language

INTRODUCCIÓN

La Corporación Eléctrica Nacional (CORPOELEC) es una sociedad anónima del gobierno de la República Bolivariana de Venezuela adscrita al Ministerio del Poder Popular para la Energía Eléctrica. Constituida a partir de la fusión de catorce (14) empresas filiales con la entrada en vigencia del Decreto con rango, valor y fuerza de Ley N° 5.330 del dos (2) de mayo del año 2007, la empresa se organiza para dar respuesta fundamentalmente a tres de sus procesos medulares: generación, transmisión y distribución de la energía eléctrica en todo el territorio nacional.

El sesenta y cinco por ciento (65%) de la energía que se genera en Venezuela proviene del Complejo de Generación Hidroeléctrica del Bajo Caroní ubicado en la región de Guayana, el cual está compuesto por un sistema de cuatro represas, a saber: Simón Bolívar (Guri), Tocoma, Caruachi y Macagua. Actualmente, el complejo cuenta con una capacidad instalada de 10.000 MW y desde la región de Guayana se exporta al centro occidente del país 6.300 MW a través de un sistema troncal de transmisión de ultra alta tensión con niveles de 765/400/230 kV. Este sistema interconecta siete (7) subestaciones a lo largo del país. De esta forma el Sistema Eléctrico Nacional (SEN) es un conjunto de sistemas interconectados, sincronizados a la frecuencia establecida en el complejo generador del bajo Caroní.

Las líneas de transmisión de 765 kV usualmente están expuestas a ciertos agentes del medio ambiente que pueden generar importantes perturbaciones en el transporte de la energía, como lo son las descargas atmosféricas, el contacto o cercanía con la vegetación y animales, la contaminación, los cambios de temperatura, entre otros. Por otra parte, el flujo de potencia fuera de los rangos permitidos puede

ocasionar deformaciones físicas en la línea, como elongaciones, puntos calientes o degradación de partes y mecanismos de aislación y sujeción. Por esto es importante la presencia de dispositivos que sean capaces de aislar las fallas y controlar la energía que se transporta, procurando no alterar la estabilidad del sistema interconectado nacional.

Los sistemas de protección y teleprotección actualmente se comunican mediante enlaces de fibra óptica, radio enlaces y sistemas de onda portadora (PLC, del inglés Power Line Carrier), proporcionando mecanismos confiables de protección en las líneas. Sin embargo, la falta de estandarización de protocolos y normas es un problema que dificulta la interoperabilidad y modernización de equipos de operación y supervisión.

CORPOELEC desde el año 2013, inició el “Proyecto de Modernización del Sistema de Transmisión a 765 kV”, proyectado a cinco (5) años, que incluye la sustitución de doscientos (200) transformadores de corriente, cinco (5) reactores, setenta (70) interruptores de potencia, trescientos (300) tableros de protecciones en las siete subestaciones de transmisión, trece (13) líneas troncales y diez (10) líneas generadoras. Entre las mejoras que aportará el proyecto, se destaca la modernización de los sistemas de control que incorporarán un nuevo estándar universal que da respuesta a las limitaciones de integración expuestas anteriormente.

El estándar IEC 61850 es una norma que nace por la necesidad de interconectar equipos de distintos fabricantes en una subestación o subestaciones remotas. El estándar apunta fundamentalmente a sustituir el mapeo de más de cincuenta (50) protocolos existentes, muchos de los cuales son propietarios.

IEC 61850 es parte de la Comisión Electrotécnica Internacional (IEC). Su implementación debe realizarse sobre topologías de red redundantes y tolerantes a

fallas, permitiendo configuraciones de doble estrella jerárquica, doble anillo o híbrida. El perfil de comunicaciones, según aplique, puede ser como los descritos en el modelo OSI: TCP/IP, UDP o Ethernet introduciendo soporte para ACSI (Servicios abstractos de comunicaciones), los cuales deben ser mapeados mediante protocolos MMS (Messaging), aplicaciones cliente-servidor y valores analógicos muestreados SV (Samples Values) y de tipo GOOSE (Generic Object Oriented Substation Event). Los mensaje de tipo GOOSE son utilizados para enviar información crítica de estado entre los distintos IED (Dispositivo Electrónico Inteligente) de la subestación o IED remotos. Como el tiempo comunicación debe establecerse en el orden de los milisegundos, los mensajes GOOSE se publican en multidifusión (Multicast) sin direccionamiento IP, moviéndose solamente a través de la primera y segunda capa del modelo OSI (capa física y de enlace) para ser capaces de ejecutar comandos y funciones en equipos de protección y teleprotección en el mínimo tiempo posible.

A pesar de la gran aceptación de los fabricantes, la implementación del estándar IEC 61850 se limita mayormente a la supervisión y no a la operación de equipos, pero a medida que los fabricantes masifican su soporte, IEC 61850 se consolida como estándar universal por sus potencialidades operativas.

El presente trabajo, busca demostrar con el diseño de un prototipo de bajo costo, la posibilidad de su implementación operativa, simulando un contexto real de comunicación entre las subestaciones del vínculo más largo del sistema de transmisión a 765 kV venezolano (San Gerónimo – La Arenosa).

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

Las comunicaciones entre múltiples IED (por ejemplo, la señalización e intercambio de información) típicamente han sido realizadas mediante una combinación de cableados rígidos entre dispositivos y comunicaciones seriales de baja velocidad. A menudo, la señalización se lograba conectando las salidas de un IED a las entradas de otro IED. Esta forma de interconexión es poco flexible y limitada en su alcance de control. Una lógica de control sofisticado entre IED podría requerir un gran número de interconexiones cableadas entre múltiples IED, siendo muy poco práctica su implementación.

Frecuentemente las comunicaciones seriales de baja velocidad son limitadas en un arreglo de comunicaciones tipo maestro/esclavo, donde las comunicaciones entre IED de tipo "punto a punto" son difíciles de realizar.

El problema de interconexión físico y lógico es una limitante que impide el la implementación de nuevas tecnologías, haciendo que la transitoriedad hacia ellas sea muy lenta porque cada modernización trae cambios radicales.

La utilización de un estándar universal posibilita la interoperabilidad de los equipos y con esto un proceso de modernización progresivo y dinámico.

JUSTIFICACIÓN

El sistema de transmisión a 765 kV data de unos treinta años y su modernización es indispensable, no solo para la ampliación de la capacidad de transmisión de potencia, sino también para convertirse en un sistema más flexible de cara a los avances tecnológicos.

La implementación de equipos que se comuniquen mediante la norma IEC 61850 producirá una notable simplificación de la arquitectura de datos de la subestación, expandiendo las capacidades de supervisión remota y comunicación entre otras subestaciones.

Otra ventaja es la simplificación de la topología de red, reduciendo las complejas conexiones de cableado rígido entre equipos de diferentes fabricantes.

El diseño del prototipo que realiza la lectura de señales críticas especificadas en el estándar IEC 61850 y ejecuta funciones en otros equipos, pretende impulsar el uso de las potencialidades operativas del protocolo de comunicación, buscando demostrar que es posible manipular tramas en la capa de enlace de red con un dispositivo programable de bajo costo.

OBJETIVOS

Objetivo general

Diseñar un prototipo de protección y teleprotección de línea de transmisión de ultra alta tensión bajo el estándar universal de subestaciones eléctricas IEC 61850 que se comuniquen en forma fiable con Dispositivos Electrónicos Inteligentes (IED) remotos, mediante un enlace de fibra óptica.

Objetivos específicos

1. Obtener los diagramas unifilares de dos subestaciones asociadas a la línea 765kV y modelarla según su configuración en un archivo de metadatos acorde al SCD (Substation Configuration Description) de la norma IEC 61850.
2. Diseñar el prototipo que permita analizar tramas Ethernet tipo GOOSE y SV sobre la capa física y de enlace del modelo OSI, permitiendo ejecutar comandos a los IED.
3. Programar en el prototipo una capa lógica de control que escoja el esquema de protección según la lectura de los mensajes GOOSE y SV.
4. Escoger un esquema específico de protección real y cargarlo al prototipo.
5. Emular una arquitectura de red con fibra óptica en un laboratorio de pruebas de CORPOELEC y realizar los estudios de respuesta del prototipo (tasa de bits de error, dominio de colisión, tiempos de respuesta de las tramas y equipos).
6. Concluir si los resultados se encuentran sobre los parámetros aceptables de confiabilidad y tiempo.

ANTECEDENTES

La búsqueda de mayor confiabilidad y explotación de las posibilidades que ofrece el estándar IEC 61850, llevaron a los principales fabricantes del mundo a dar soporte a los protocolos del estándar. Sin embargo, la referencia más cercana a este trabajo proviene de la línea de investigación abierta por el Dr. Steven Blair, de la Universidad de Strachclyde, Glasgow, Escocia, titulada *Rapid-prototyping protection schemes with IEC 61850* ^[1].

El trabajo del Prof. Blair consiste en un prototipo para el control de esquemas de protección sobre un microprocesador con arquitectura de 32 bits Raspberry Pi, y el desarrollo de unas librerías que decodifican las tramas críticas de información que especifica el estándar IEC 61850: Generic Object Oriented Substation Events (GOOSE) y Samples Values (SV). El prototipo es capaz de conectarse a una red LAN en una subestación, validar su SCD (Substation Configuration Description file, un archivo que describe los nodos de los IED conectados en la subestación) e intercambiar mensajes críticos entre ellos.

La diferencia con el trabajo del Dr. Blair en el diseño del prototipo con el Atmega328 Arduino R3, es que la arquitectura de 8 bits imposibilita la utilización de las librerías en 32 bits. Para leer las tramas Ethernet se le añade una tarjeta compatible de red (Ethernet Shield), la cual posee un chip Wiznet W5100 que permite el acceso a librerías de control para el filtrado de las tramas GOOSE y SV del protocolo IEC 61850. De esta forma, es posible la realización de lecturas y proceso de comandos a otros IED de la red.

CAPÍTULO II

GENERALIDADES Y ASPECTOS TEÓRICOS

2.1. Descripción general de la línea troncal de transmisión de 765 kV.

El sistema troncal de transmisión 765 kV, está compuesto por siete (7) subestaciones. Comienza por las subestaciones Guri B y Malena ubicadas en el estado Bolívar. De la subestación San Gerónimo en Guárico se derivan tres vínculos que terminan en la subestaciones: SUR/OMZ, La Horqueta y La Arenosa, que a su vez se interconectan entre sí y finalmente con la subestación (S/E Yaracuy, como se muestra en la Figura 1.



Figura 1: Mapa de Subestaciones a 765 kV

Sobre el vínculo Malena – San Gerónimo se importan, hacia el centro del

país 6.300 MW y la importación hacia la región Centro Occidente es aproximadamente de 2.700 MW.

En el diagrama unifilar de la Figura 2 se observa el comienzo de sistema de transmisión troncal en la S/E Guri B que finaliza en la S/E Yaracuy con un recorrido total de 1.320 Km.

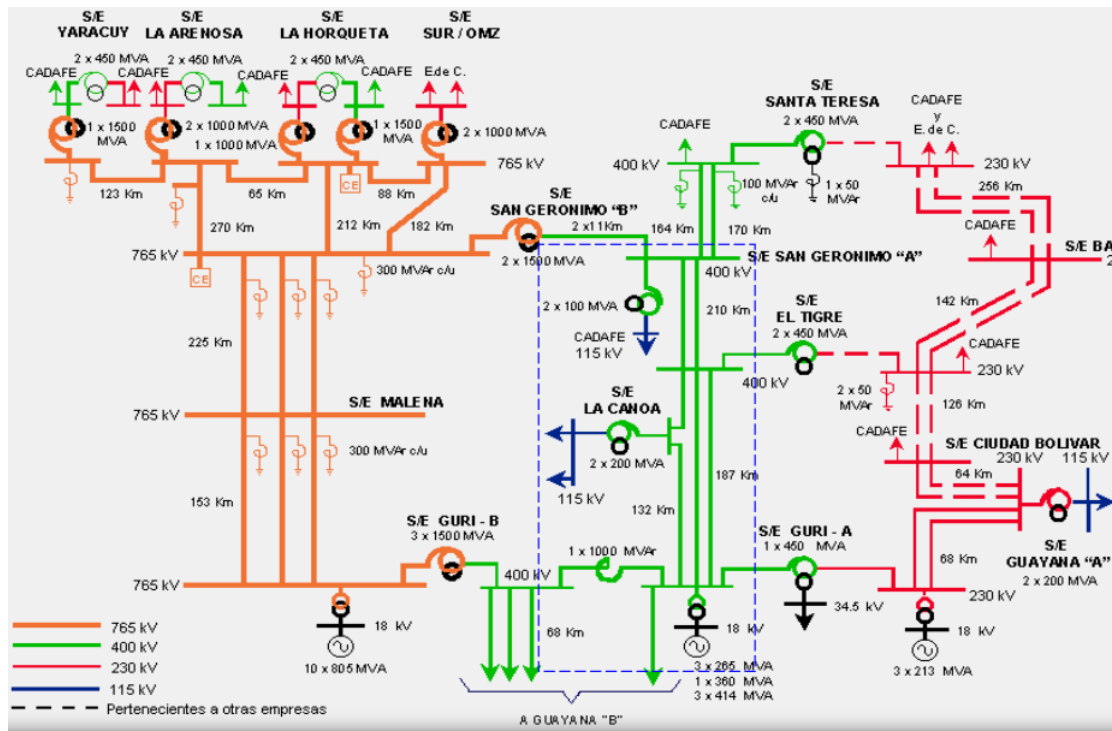


Figura 2: Unifilar Sistema de Transmisión a 765 kV

[2]

2.1.1. Esquema general de una subestación de ultra alta tensión

Las subestaciones de ultra alta tensión se caracterizan por manejar niveles de voltaje entre 230 kV y 765 kV.

Entre los principales componentes se encuentran: Interruptores automáticos, seccionadores, conmutadores de puesta a tierra, transformadores de corriente (TC), transformadores de potencial o transformadores de voltaje capacitor, capacitores de acoplamiento, filtros de línea, pararrayos y/o espinterómetros, reactores de derivación, reactores limitadores de corriente, barras y aisladores de estación, sistemas de puesta a tierra, capacitores en serie, capacitores en derivación entre otros.

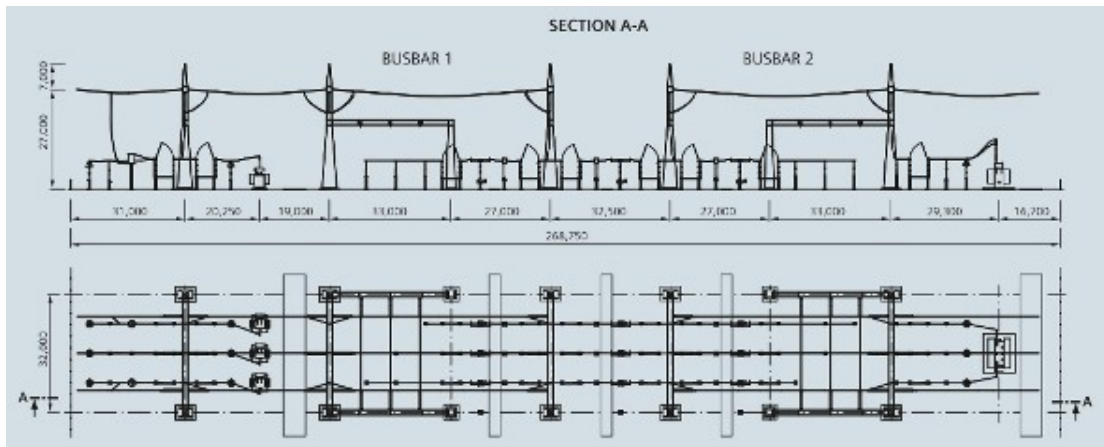


Figura 3: Sección de una subestación de ultra alta tensión

[3]

2.1.2. Sistema de protecciones

Las protecciones de los sistemas de potencia son componentes integrales que evitan la destrucción de un conjunto de equipos o dispositivos interconectados en una tarea común, por causa de una falla que podría iniciarse de manera simple y después extenderse sin control en forma encadenada. El sistema de protecciones debe aislar la sección donde se ha producido la falla buscando perturbar lo menos posible la red troncal de alta de tensión.

Los relés son dispositivos que controlan el estado de un interruptor mediante

una entrada eléctrica.

2.1.2.1. Sistemas de protección de líneas de transmisión

- a. Protección de sobrecorriente: Este relé opera en forma casi instantánea para un valor de corriente excesivo, indicando una falla en el aparato o circuito protegido. Su tiempo de operación está en el orden de 0,05 segundos (3 ciclos).
- b. Protección de sobrecorriente con retraso de tiempo: Este es un relé con una característica de tiempo definida e inversa, que opera cuando la corriente en el circuito excede un valor determinado, por lo general, a mayor corriente menor tiempo de la característica inversa.
- c. Protección de recierre para corriente alterna: Relé que funciona cuando el ángulo de fase entre dos tensiones o corrientes o entre una tensión y una corriente alcanza un valor determinado.
- d. Protección de distancia: Los relés de distancia son dispositivos empleados para la protección de líneas de transmisión de mediano y alto voltaje. Los relés de distancia tipo admitancia o similar de tres zonas, son para fallas entre fases y los relés de distancia tipo reactancia de tres zonas son para protección de fallas de una fase a tierra. Con facilidades para compensar el efecto de inductancia mutua. Se utiliza este tipo de protección, debido a que éstos funcionan en base a la característica de impedancia o reactancia de la línea y los demás parámetros de las líneas de transmisión que varían constantemente (voltaje,

corriente, potencia, carga). Además son de alta velocidad, no cambian su característica de funcionamiento con los cambios de generación y por su característica direccional.

- e. Protección de sobrecorriente direccional: Cuando la coordinación de las protecciones de sobrecorriente se hace complicada y en ocasiones imposible en líneas de transmisión con fuente de alimentación en ambos extremos, se pueden emplear relés de sobrecorriente supervisadas por una unidad direccional. La direccionalidad simplifica el problema de selectividad y seguridad. Este relé se aplica a líneas como respaldo para proteger equipos de la subestación, es otra forma de protección de respaldo de la propia línea. El relé direccional de sobrecorriente de tierra es alimentado por los secundarios de los transformadores de potencial instalados en el bus.
- f. Protección piloto: Es el mejor tipo para la protección de líneas. Además, emplea un canal de comunicaciones en conjunción con las protecciones para averiguar en el menor tiempo posible si una falla se encuentra en la línea protegida o fuera de ella.
- g. Protección diferencial de línea: Se utiliza para protección, control y supervisión de líneas y cables aéreos en todo tipo de redes, se puede utilizar hasta en los niveles de tensión más altos. Es adecuado para protección de líneas con carga elevada y líneas con varias terminales en las que los requisitos de disparo sean de uno, dos y/o tres polos.

2.1.2.2. Sistemas de protección a interruptores

- a. Arreglo mecánico para detectar discrepancia o discordancia de polos: Este tipo de arreglo actúa en el momento en que algún polo del interruptor queda en una posición distinta a los otros. En el caso de interruptores con disparo y recierre monopolar deberá adicionarse ese tiempo de discordancia en el temporizador.
- b. Protección contra falla del interruptor: La finalidad de esta protección es detectar la falla de apertura del interruptor al recibir la orden de disparo enviada por alguna protección. Para su operación se requiere otra protección que envíe la orden de disparo. Si al transcurrir cierto tiempo la protección no actúa y hay presencia de corriente en la línea, el esquema será activado.

2.1.3. Causas de funcionamiento anormal en los sistemas de potencia

El sistema de potencia puede ser afectado por muchas situaciones que produzcan una operación fuera de las condiciones normales. Estas posibles causas pueden ser:

- Falla de los componentes del sistema
- Situaciones de carácter imprevisto (por ejemplo, tormentas)
- Errores de operación. (manuales o automáticos)

Estas situaciones provocan efectos muy variados en el sistema de potencia

tales como:

- Mal servicio.
- Pérdida de la estabilidad.
- Daños de los equipos.

CORPOELEC es la empresa encargada de corregir las situaciones anormales que se presentan en la operación del sistema. Las interrupciones del servicio y la variación de los parámetros de la red (tensión, corriente, frecuencia, etc.) fuera de los límites permitidos, son consecuencia común de una operación incorrecta, causando enormes inconvenientes técnicos y económicos.

2.2. Descripción del estándar IEC 61850

La norma IEC 61850 surge con el objetivo de garantizar la interoperabilidad entre distintos equipos electrónicos inteligentes (IED), que componen un sistema de automatización de una subestación eléctrica.

Para lograr este objetivo el estándar desarrolla un modelo de datos que recoge toda la información que puede ser necesaria en un sistema de automatización de una instalación eléctrica, de modo que todos los IED que cumplen con las especificaciones del estándar organicen su información según el mismo modelo de datos. La interoperabilidad, sin embargo, no garantiza la intercambiabilidad, lo que significa que las funcionalidades para las que está preparado cada dispositivo no están estandarizadas.

2.2.1. Arquitectura de la subestación

Los sistemas que se tratan en la IEC 61850 son sistemas de automatización de las

subestaciones eléctricas (SAS). Por esto, los diferentes elementos que se deben modelar para su control y supervisión son los componentes típicos que se pueden encontrar dentro de una subestación, como: Seccionadores, interruptores, transformadores de corriente, tensión y potencia. Una vez modelados estos elementos, es necesaria su separación en distintos bloques eléctricos (bahías o bays), teniendo en cuenta que, por motivos de funcionalidad o fabricación, los dispositivos pueden ser controlados de manera conjunta.

2.2.2. Niveles e Interfaces Lógicas

Las limitaciones actuales de interoperabilidad y excesivo uso de cableado para interconectar equipos que no trabajan sobre la misma norma, son los problemas a vencer y el estándar IEC 61850 permite a los sistemas de automatización superar estas deficiencias.

Partiendo del modelo abstracto de información que se repite de manera genérica en todas las subestaciones eléctricas, la norma separa este sistema en tres niveles jerárquicos, según se muestra en la Figura 4:

1. Nivel de proceso (Process Level): Es el nivel correspondiente a los dispositivos electrónicos (IED) que permiten el acceso a los equipos desde niveles superiores. Un ejemplo podría ser un PLC que controla y gestiona la información de un interruptor.
2. Nivel de bahía o bloque eléctrico (Bay Level): Este nivel se corresponde con los IED que se encargan de controlar y proteger a los elementos de un determinado bloque eléctrico. Un relé es un ejemplo de este tipo de dispositivos.

3. Nivel de subestación (Station Level): Corresponde con el puesto de operación local de la subestación, desde el que el operador puede supervisar y gobernar los distintos equipos. Como se puede ver en la Figura 4, puede tratarse o bien de una interfaz hombre-máquina (HMI) o de un puesto de trabajo remoto.

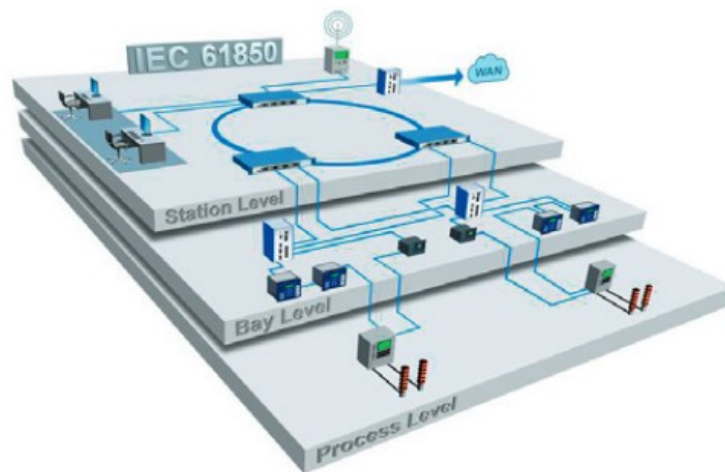


Figura 4: Niveles SAS

[4]

A la vez que se muestran los distintos niveles, también existen interfaces lógicas entre los dispositivos del SAS. Varias de estas interfaces pueden estar implementadas en un único conector físico y en una única LAN (Red de Área Local). De hecho, el estándar propone una configuración con dos buses que se repartirían las interfaces lógicas presentes en el sistema de comunicación:

- 1 Bus de subestación: Alberga los intercambios de información que comunican los niveles de bahía y subestación o nivel de subestación con un puesto de control remoto.

- 2 Bus de proceso: Alberga las interfaces lógicas correspondientes a la relación entre niveles de proceso y de bahía.

La disposición física de los buses depende de la división que se haya realizado previamente de los elementos de la subestación en distintos bloques eléctricos.

2.2.3. Nodos lógicos (LN) y funciones

- Nodos lógicos: Son las unidades lógicas básicas que componen una función que debe realizar el sistema, y se alojan en los IED. La norma estandariza las distintas clases de LN.
- Funciones: El sistema de automatización de la subestación eléctrica debe llevar a cabo una serie de funciones de protección, control o supervisión. La norma estandariza cuál es la estructura que deben seguir las diferentes funciones de un SAS, las cuales deben estar compuestas por nodos lógicos.

2.2.4. Modelo jerarquizado de información

La norma IEC 61850 presenta un modelo de información jerarquizado para la representación de un SAS. En él, cada nodo lógico contiene una serie de datos, los cuales se componen a su vez de atributos. De esta forma, se plantea una virtualización de la subestación eléctrica; es decir, que todo lo que hay dentro de la subestación se modelará mediante nodos lógicos. Un ejemplo genérico del modelo jerarquizado se puede observar en la Tabla 1.

Tabla 1: Grupos de Nodos Lógicos

Indicador	Grupos de Nodos Lógicos	Qty	
L	System Logical Node	3	
P	Protection Functions	28	- PDIR Direccional - PDIS Distancia - PSCH Esquema de protección - PTOC Sobrecorriente temporizada - PTOV Sobretensión - más ...
R	Protection Related Functions	10	
C	Supervisory control	5	
G	Generic Function References	3	
I	Interface and Archiving	4	- RDIR Elemento direccional - RDRE Registrador de oscilos - RBRF Falla interruptor - RFLO Localizador de fallas - RREC Auto recierre - más ...
A	Automatic control	4	
M	Metering and Measurement	8	- MMXU Unidad de Medición - MMTR Medición - MSQI Secuencia y desbalance - MHAI Armónicos e inter armónicos - más
S	Sensors, Monitoring	4	
X	Switchgear	2	
T	Instrument Transformer	2	- XCBR Interruptor - XSWI Seccionador
Y	Power Transformer	4	
Z	Further (power system) equipment	5	

2.2.5. Clases genéricas de datos

Los datos que componen los LN (nodos lógicos) tienen a su vez distintos atributos asociados. El apartado 7-3 de la norma define las clases genéricas de datos como estructuras para tipos de datos que comparten la organización y tipos de atributos, aunque tengan distinto significado.

2.2.6. Dispositivo Lógico

El dispositivo lógico LD (Logical Device) es un concepto necesario para completar el modelo de información del SAS, y se refiere a un elemento de jerarquía inmediatamente superior al nodo lógico. Es decir, un IED albergará distintos LD, que a su vez contendrán una serie de nodos lógicos, tal como se muestra en la Figura 5.

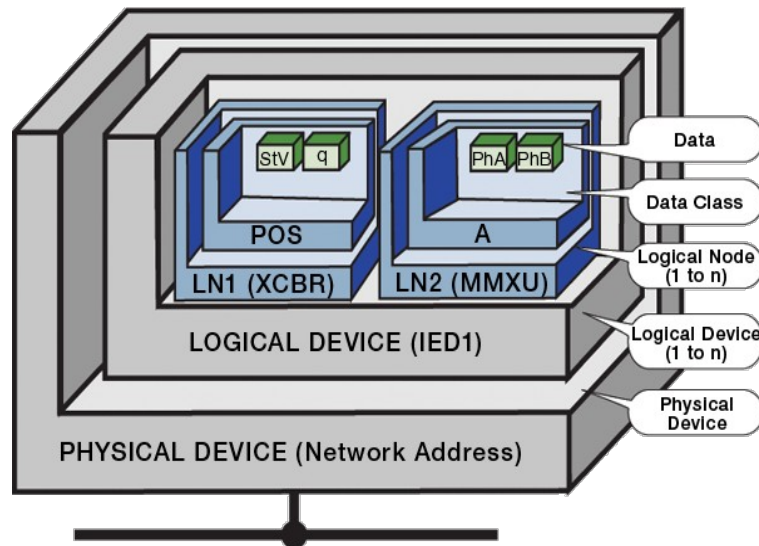


Figura 5: Modelo abstracto de SAS

[5]

2.2.7. Modelo de Servicios Abstractos de Comunicación (ACSI)

La norma define ACSI en términos de un modelo de clase jerárquico de toda la información que puede ser accedida a través de una red de comunicaciones y los servicios que operan sobre estas clases de parámetros asociados con cada servicio.

2.2.8. Eventos Genéricos de la Subestación (GSE)

Es el servicio que permite la comunicación de eventos genéricos de la subestación a varios dispositivos IED dentro del sistema, de manera simultánea, rápida y segura. Está relacionado con acciones automáticas que requieran el intercambio de información con una importante restricción temporal. Para poder proporcionar este servicio la norma establece dos bloques de control. El GoCB, que permite el envío de mensajes tipo GOOSE, basado en un mecanismo proveedor/suscriptor y el GsCB que permite el envío de mensajes GSSE de

características similares al GoCB pero con tratamiento de distinto tipo de información. La Figura 6 muestra cómo un switch realiza la priorización de mensajes tipo GOOSE.

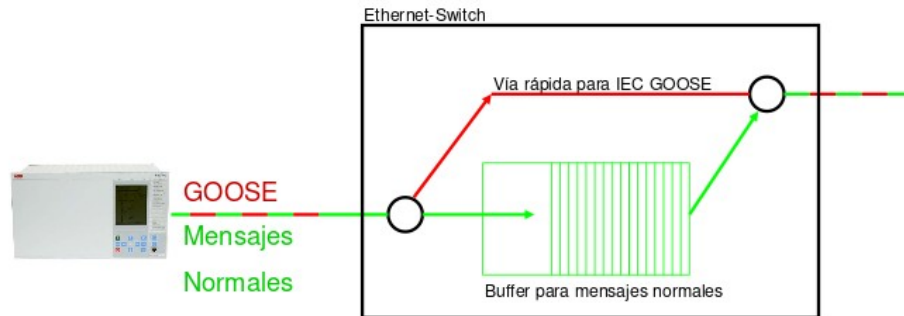


Figura 6: Priorización en la transmisión de mensajes tipo GOOSE

[6]

Este servicio permite la transmisión de valores instantáneos de medidas analógicas, minimizando el tiempo que transcurre desde el muestreo hasta la recepción del mensaje.

La Figura 7 muestra el tiempo de actuación de los mensajes:

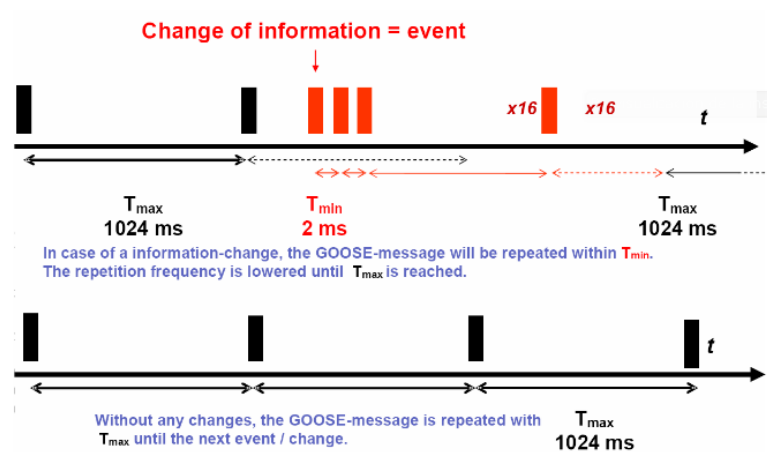


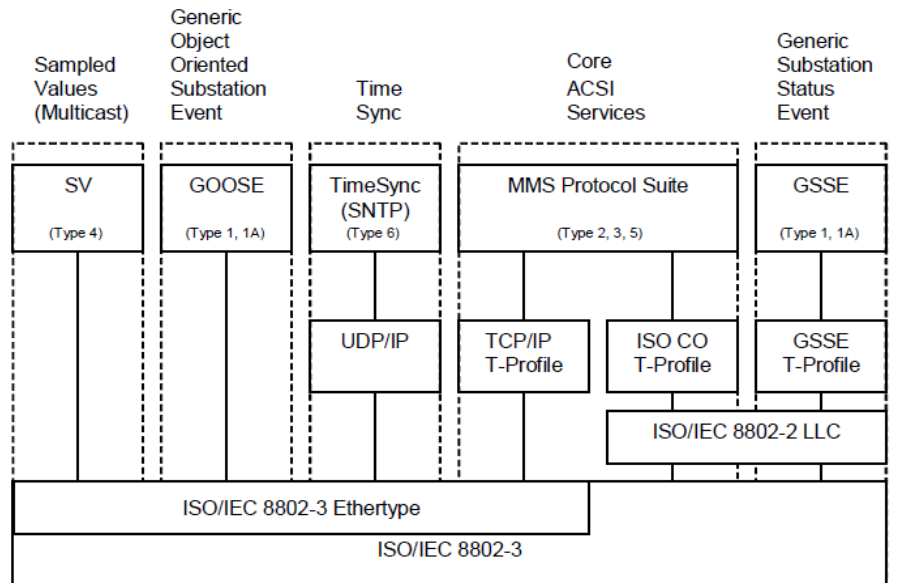
Figura 7: Tiempos de actuación de los mensajes

[7]

SV: Tipo 4. Mensajes de muestreo de datos. Los mensajes de muestreo contienen información de los valores presentes en un tiempo determinado, por ejemplo, voltaje, corriente, impedancia, etc.

GOOSE: Tipo 1, 1A. Prioritarios, de alta velocidad. Contienen generalmente funciones con instrucciones de disparo, apertura, cierre, etc.

MMS: Tipo 2, 3 y 5. Mensajes de mediana y baja velocidad con funciones de transferencia de archivos. La figura 8, muestra el desempeño de los mensajes en la pila de protocolos.



(Type x) is the Message type and performance class defined in IEC 61850-5

IEC 136/04

Figura 8: Desempeño de los mensajes en la pila de protocolos

[8]

2.2.9. Lenguaje de configuración (SCL)

El lenguaje de descripción de la configuración de subestación (SCL) es una herramienta que permite intercambiar las descripciones de las capacidades de los IED y las descripciones del sistema de automatización de la subestación.

SCL permite lo siguiente:

- a) Una especificación del sistema en términos de un diagrama unifilar, asignando nodos lógicos a cada parte e indicando a los equipos en el diagrama con sus funciones que desempeñan.
- b) Preconfiguración de los IED con un número fijo de nodos lógicos
- c) Preconfiguración de los IED de acuerdo a los procesos o ciertas estructuras.
- d) Un proceso de configuración completa de los IED modelando sus funciones y optimizando las comunicaciones entre los clientes.

El lenguaje SCL define:

- 1. La especificación funcional del Sistema de Automatización de Subestación (SAS).
- 2. La descripción de la capacidad de los IED.

2.2.10. Modelo (SCL)

El SCL está basado en el lenguaje (XML, extended Mark-up Language). La meta del SCL es tener una descripción formal del SAS en un nivel de ingeniería. El

modelo SCL tiene tres partes básicas:

- Subestación: Es la descripción de los equipos del patio de maniobra (topología) y sus funciones.
- IED: Dentro del IED se implementan los nodos lógicos y los objetos de la subestación.
- Comunicación: Son las comunicaciones a la red y los puntos de acceso, describe las conexiones entre IED en base a los nodos lógicos como son cliente/servidor.

2.2.11. Descripción de los tipos de archivo SCL

Los archivos SCL son utilizados para intercambiar datos de configuración entre herramientas diferentes, posiblemente de distintos proveedores. Existen cuatro propósitos diferentes para intercambiar datos de SCL, y por lo tanto cuatro tipos de archivos SCL para ser distinguidos en el intercambio entre herramientas. Esto se hace por medio de diferentes extensiones de archivos, las cuales se describen a continuación:

1. ICD (IED Capability Description). Contiene las características de cada dispositivo relacionadas con las funciones de comunicación y el modelo de datos.
2. SSD (System Specification Description): Contiene las capacidades funcionales de ingeniería de un IED. El esquema unifilar junto a las funciones que se realizarán en los equipos primarios en términos de nodos lógicos.
3. SCD (System Configuration Description): Se trata de un archivo que el integrador del sistema exportará como resultado de las ICDs y las SSDs, el

cual contiene la configuración del sistema: todos los IEDs, la configuración de las comunicaciones y la descripción de la subestación.

4. CID (Configured IED Description): Contiene, para un equipo, las configuraciones y datos necesarios que definen la interacción y comunicación entre el resto de los equipos del sistema.

2.2.12. Estructura del estándar

En la Figura 9 se muestra la estructura general del estándar, donde los aspectos del sistema se ubican en las partes 1 a la 5; la configuración en la parte 6; el modelo de datos y servicios abstractos y comunicación en las partes 7 – 1 a 7 – 4; el mapeo de la comunicación en las partes 9 – 1, 8 – 1 y 9 – 2 y en la parte 10 las pruebas de conformidad.

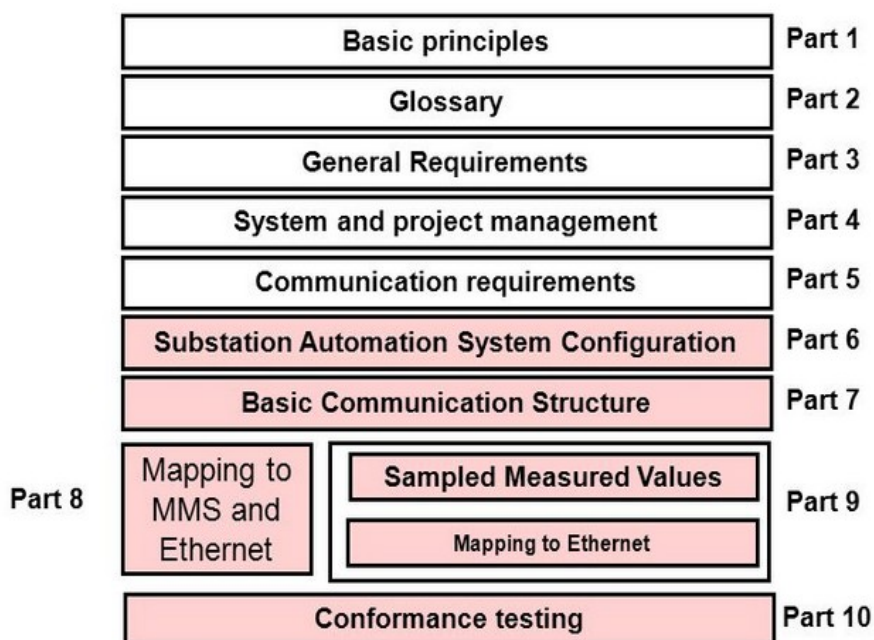


Figura 9: Estructura del estándar

[9]

2.3. Prototipo de Teleprotección Remoto (TPR)

Se escoge la Arduino como plataforma para el prototipo electrónico de protección y teleprotección por su versatilidad de programación, bajo costo y capacidad de acoplamiento con una interfaz de red a través de placa dotada con un chip W5100 (Shield Ethernet).

Arduino es una plataforma computacional física open-source, basada en una tarjeta de I/O y un entorno de desarrollo que implementa el lenguaje Processing/Wiring.

2.3.1. Arduino UNO R3

Arduino Uno R3 utiliza el microcontrolador ATmega328. En adición a las características de tarjetas anteriores, Arduino Uno utiliza el ATmega16U2 para el manejo de USB en lugar del 8U2 (o del FTDI encontrado en generaciones previas). Esto permite una tasa de transferencia más rápida y más capacidad de memoria.

Arduino es utilizado para crear objetos interactivos, teniendo la capacidad de leer datos de otros dispositivos en forma analógica o digital.

Arduino es hardware abierto, sus esquemáticos, componentes y circuitos electrónicos están publicados en Internet y son de libre acceso.

En la Figura 10 se muestra el circuito de la placa Arduino UNO R3 identificando los componentes y etapas del circuito.

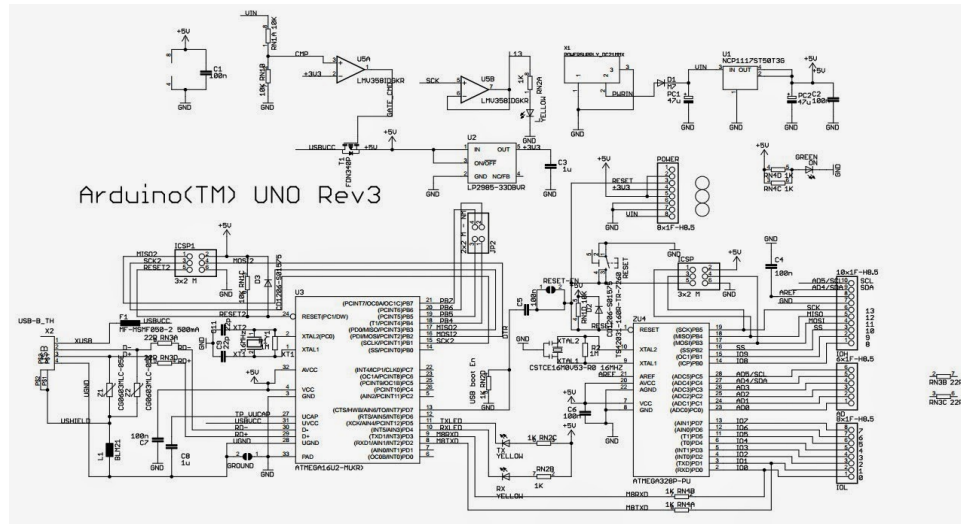


Figura 10: Esquemático de la placa Arduino UNO R3.

[10]

En la Tabla 2 se muestran las especificaciones técnicas de la placa Arduino UNO R3.

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Tabla 2: Especificaciones técnicas placa Arduino UNO R3

En la Figura 11 se muestra la disposición de los pines de la placa Arduino UNO R3 de los cuales algunos serán ocupados para alimentar los dispositivos de acople, otros para la transmisión de datos y señales digitales para el accionamiento de los relés.

Se puede observar cómo las disposiciones laterales de entradas y salidas analógicas y digitales funcionan como base de acople para para otras tarjetas compatibles.

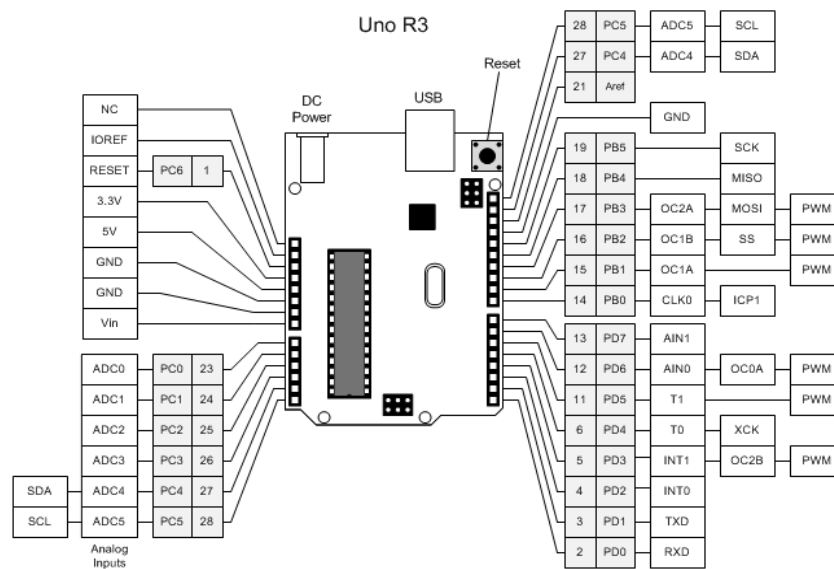


Figura 11: Descripción de los pines de la placa Arduino UNO R3

[11]

En este trabajo se utilizará una tarjeta de acople que extenderá funcionalidades al Arduino UNO R3, proporcionándole la capacidad de conectarse a redes WAN y LAN, mediante un puerto Ethernet. En la Figura 12 se muestra el microprocesador programable ATMEGA 328,



Figura 12: Componentes principales de la placa Arduino UNO R3

[12]

El AVR es parte de la familia de microcontroladores RISC del fabricante estadounidense Atmel, su arquitectura fue concebida por estudiantes del Norwegian Institute of Tecnology, posteriormente desarrollada por Atmel. En la Figura 13 se muestra el diagrama de bloques de la arquitectura del núcleo AVR.

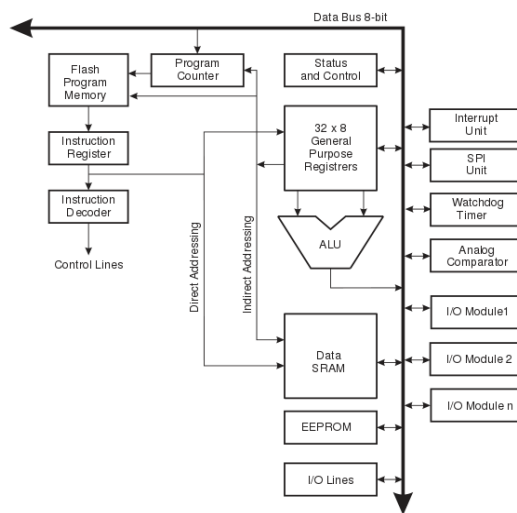


Figura 13: Diagrama de bloques del AVR

[13]

Se puede encontrar, entre otros, el bloque de la Unidad Aritmético Lógica (ALU), las etapas de memoria volátil, fija y externa (SRAM, EEPROM y Flash) , los módulos de entrada y salida I/O, la unidad de control de interrupciones, el reloj, el comparador analógico, el registro y decodificador de instrucciones.

El registro del AVR permite ejecutar instrucciones a alta velocidad. La memoria se puede extender mediante un almacenamiento externo si se quiere, pero para los efectos de este trabajo el tiempo es un factor de suma importancia, porque lo que las principales funciones deben residir en este registro de 32 direcciones de 8 bits cada una.

	7	0	Addr.	
General Purpose Working Registers	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

Figura 14: Registros Hexadecimales

[14]

Como se dijo anteriormente, el acceso al registro debe ser veloz y a través del decodificador de instrucciones hexadecimal. En la Figura 14 se observa el registro y las correspondientes direcciones de memoria en formato hexadecimal.

2.3.2. Ethernet Shield Arduino

El Arduino Ethernet Shield permite a una placa Arduino conectarse a la capa de enlace de una red LAN. Está basada en el chip Ethernet Wiznet W5100. El Wiznet W5100 provee de una pila de red IP capaz de procesar los protocolos TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas.

En la Figura 15 se pueden reconocer los terminales de transmisión y recepción de datos

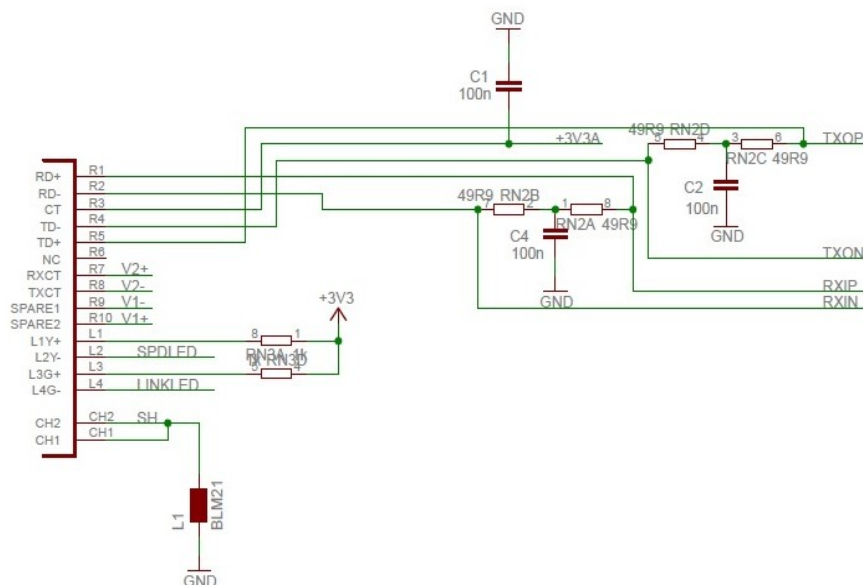


Figura 15: Circuito Ethernet Shield

[15]

En la Figura 16 se describe la distribución de los pines de conexión identificando componentes de la placa

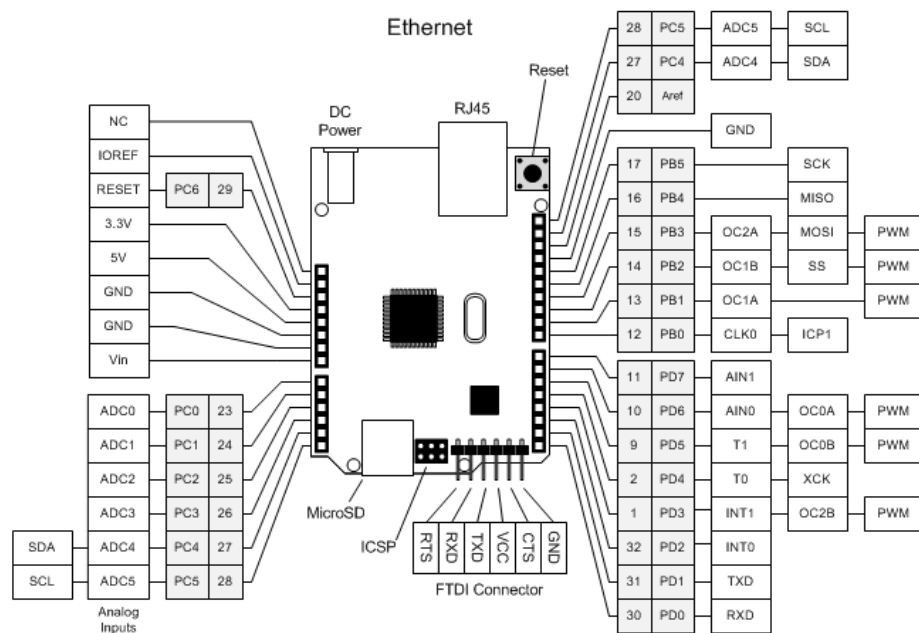


Figura 16: Descripción de los pines de la placa Ethernet Shield

[16]

El shield provee un conector Ethernet estándar RJ45 y un conector lector de tarjeta Micro SD.

El shield contiene indicadores LED para la información de estado del dispositivo :

- PWR: indica que la placa y la shield están alimentadas
- LINK: indica la presencia de un enlace de red y parpadea cuando la shield envía o recibe datos
- FULLD: indica que la conexión de red es full duplex
- 100M: indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
- RX: parpadea cuando la shield recibe datos
- TX: parpadea cuando la shield envía datos

- COLL: parpadea cuando se detectan colisiones en la red

Una de las claves de este trabajo es poder acceder a las funciones del W5100 a través de las librerías Ethernet con la finalidad de decodificar las tramas GOOSE. Ya que la capa de abstracción de las librerías del Arduino solo puede manejar datagramas UDP y como se vio anteriormente en el modelo de referencia OSI, el protocolo UDP está a nivel de la capa de transporte, mientras que los mensajes GOOSE solamente llegan hasta la capa de enlace por la necesidad de optimizar el tiempo. Por lo tanto fue estrictamente necesario tener acceso a las librerías que controlan el chip Ethernet Wiznet W5100.

La Figura 17 muestra el contenido del encabezado en una trama UDP

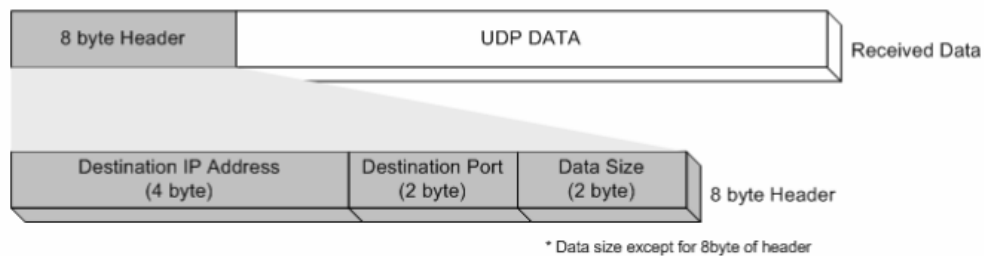


Figura 17: Trama UDP

[17]

Accediendo a las funciones del chip W5100 se logra controlar el flujo de datos de las tramas Ethernet e interpretar los mensajes de tipo GOOSE.

El Chip W5100 utiliza un buffer de almacenamiento temporal que permite manipular los datos. En la figura 18 se muestra algunas funciones básicas .

```

{
    /* first, get the received size */
    get_size = Sn_RX_RSR;
    /* calculate offset address */
    get_offset = Sn_RX_RD & gSn_RX_MASK;
    /* calculate start address(physical address) */
    get_start_address = gSn_RX_BASE + get_offset;

    /* read head information (8 bytes) */
    header_size = 8;

```

Figura 18: Funciones del microchip WZ5100

[18]

2.3.3. Entorno de Desarrollo Integrado (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Parte del Entorno de Desarrollo Integrado de Arduino (IDE, Integrated Development Environment) está basado en Processing y Wiring, el primero cuenta con las herramientas necesarias para codificar, compilar y escribir en el procesador el programa binario final. Los archivos del IDE tienen extensión “.ino” y son denominados “Sketchs”.

El segundo, (Wiring) es un entorno multiplataforma que interactúa con Processing y es capaz de controlar dispositivos en el entorno de los microprocesadores. Está basado en Java y es capaz de enlazar librerías en C y C++.

Cuando se carga un sketch (proyecto), se utiliza el bootloader de Arduino (pequeño programa que ha sido precargado al microcontrolador de la tarjeta). Esto permite cargar código sin usar ningún hardware adicional. El bootloader es activado durante unos pocos segundos mientras se reinicia el microcontrolador.

Es posible incorporar librerías extendiendo las funcionalidades básicas.

2.3.4. Estructura de un programa

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones.

```
void setup()
{
  estamentos;
}
void loop()
{
  estamentos;
}
```

En donde setup() es la parte encargada de recoger la configuración y loop() es la que contiene el programa que se ejecutará cíclicamente. Ambas funciones son necesarias para que el programa trabaje.

La función de configuración debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta sólo una vez y se utiliza para configurar o inicializar pinMode (modo de trabajo de las E/S), configuración de la comunicación en serie y otras.

La función bucle (loop) siguiente contiene el código que se ejecutará continuamente (lectura de entradas, activación de salidas, etc). Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

La Figura 19 muestra un ejemplo de la estructura básica del código

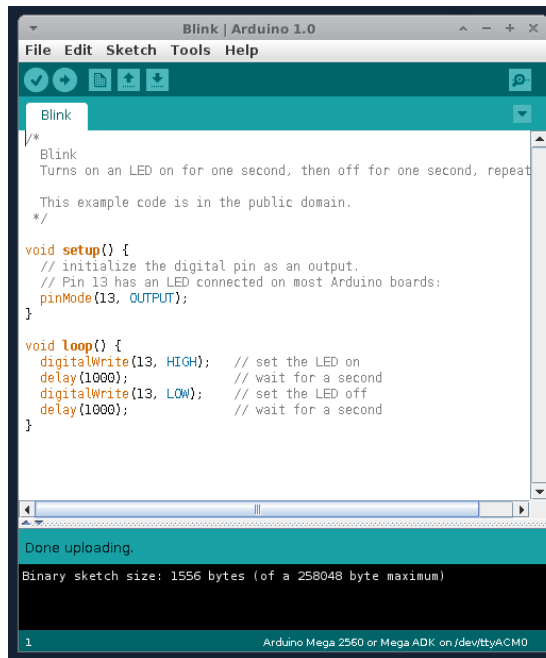


Figura 19: Interfaz de usuario (GUI)

[19]

2.4. Dispositivo Electrónico Inteligente (IED) simulado

Como se vió al principio del capítulo II, Un IED es una parte integral de la arquitectura SAS (Substation Automation System), que ejercerá funciones de muestreo, ejecutará comandos de actuación y advertirá a otros IED condiciones o eventos del sistema.

Uno de los objetivos específicos de este trabajo es poder identificar y procesar mensajes tipo GOOSE, provistos por un IED, dentro de la capa de enlace de la red. Para esto simularemos el comportamiento de un IED a través de un servidor de mensajes GOOSE por software. Para simular un IED, se utilizarán las librerías del

proyecto libIEC61850, un proyecto Open Source, que provee un servidor y cliente capaz de manejar los protocolos descritos en el estándar IEC 61850.

2.5. Herramientas para el análisis de información

2.5.1. libIEC61850

El proyecto libIEC61850 es una biblioteca disponible bajo la licencia GPLv3, que provee un desarrollo cliente/servidor para los mensajes MMS, GOOSE y SV, establecidos en el estándar IEC 61850, cuenta con las siguientes características:

- Pila completa de protocolos ISO sobre TCP/IP
- Aplicación estática de los modelos de equipos generados en código C a partir del archivo de lenguaje para la configuración de subestaciones (SCL, Substation Configuration Language).
- Creación dinámica del modelo de equipos mediante las llamadas a la API o el archivo de configuración
- Servicio de Asociación (con autenticación de contraseña)
- Lectura y escritura de los servicios MMS para variables simples y complejas.
- Servicios del modelo de navegación (GetServerDirectory y GetDeviceDirectory).
- Los servicios de datos establecidos, incluyendo la creación y eliminación dinámica de conjuntos de datos.
- El API cliente y servidor IEC 61850
- Soporte para la configuración de los servicios de control de grupo
- Un Hardware simulado en capa de abstracción con implementaciones POSIX

(Linux), WIN32 y en sistemas BSD (Mac OS X, FreeBSD).

- Herramienta de conversión y traducción de archivos SCL a modelos estáticos IED.

2.5.2. Wireshark (Analizador de protocolos Ethernet)

Wireshark es un analizador de protocolos inicialmente llamado Ethereal, de licencia libre GPL. Está basado en la librería libcap que es utilizada para capturar paquetes de datos en una red.

La función principal de Wireshark es facilitar el análisis de datos mediante un sistema de filtrado de tramas, que soporta la interpretación de más de 480 protocolos distintos.

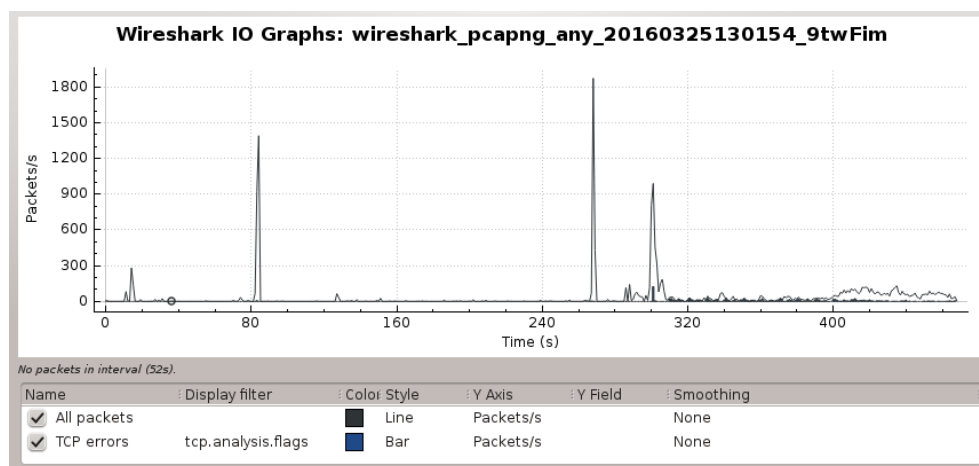


Figura 20: Flujo de tramas en el tiempo

Usualmente las interfaces de red descartan los paquetes que no tengan como destino su dirección MAC. Wireshark funciona colocando la interfaz de red en modo promiscuo y con esto se logra aceptar sin descarte todo el tráfico presente en una red.

La Figura 20 muestra el flujo de tramas Ethernet en el analizador de

protocolos.

Wireshark puede ser ejecutado en sistemas operativos tipo Unix, incluyendo Linux, Solaris, FreeBSD, NetBSD, OpenBSD, Mac OS X y también en otras plataformas como Microsoft Windows, U3 y en aplicaciones Portables (Apps).

La Figura 21 muestra un conjunto de tramas, la selección de una de ellas permite desplegar su representación en forma binaria o hexadecimal.

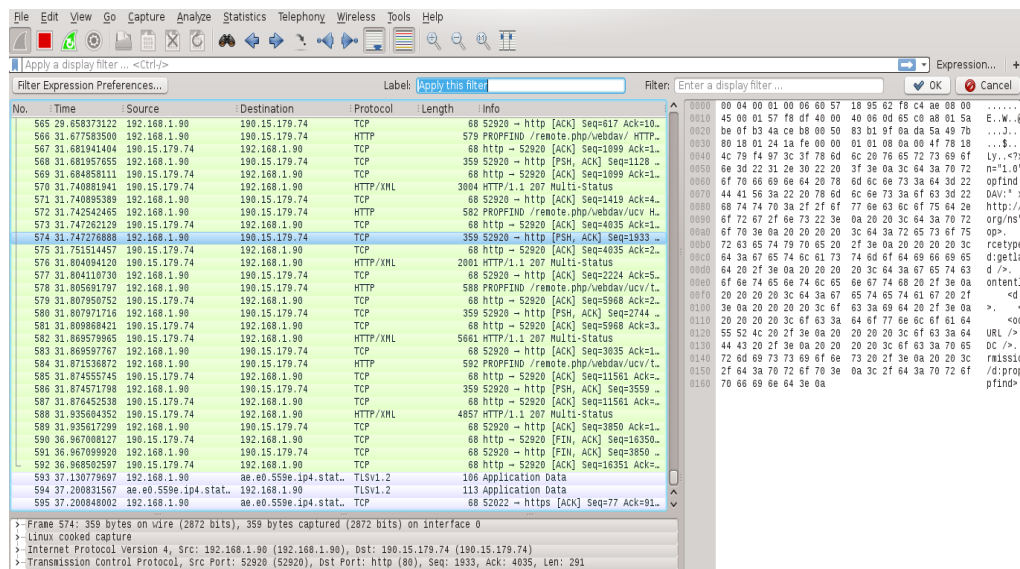


Figura 21: Analizador de protocolos Wireshark

CAPÍTULO III

MODELADO Y DISEÑO DEL PROTOTIPO

3.1. Alcance y requerimientos

Si bien el prototipo a desarrollar pudiera manejar un alcance mayor, este trabajo se enfoca principalmente a la interpretación de un mensaje de alta prioridad o trama Ethernet de tipo GOOSE, para accionar una función o comando determinado, cumpliendo con las especificaciones de la norma IEC 61850.

Una vez ejecutada la acción, el prototipo envía un mensaje de confirmación para poder medir el tiempo de respuesta. Esta información es utilizada para verificar que el prototipo procesa la información dentro de los tiempos exigidos por la norma.

3.2. Modelo conceptual

En la arquitectura del Sistema Automatizado de Subestaciones (SAS) en el contexto que se empleará el prototipo TPR, existe un IED proveedor de mensajes GOOSE, que para este caso se ubica en la S/E La Arenosa obteniendo muestras de los valores de transferencia de potencia existentes en la línea San Geónimo – La Arenosa a 765 kV con mensajes SV, al alcanzar el límite establecido permitido emite un mensaje GOOSE con una función de disparo teletransferido, al estar activado el código permisivo C1, actúan las autodesconexiones de los siguientes bloques: Sidetur, Acarigua, Anillo Yaguara – San Carlos, Bloque 1, 2 y 3 de Occidente, según

se muestra en el diagrama funcional del Sistema de Autodesconexión de Carga del SEN representado en la Figura 22.

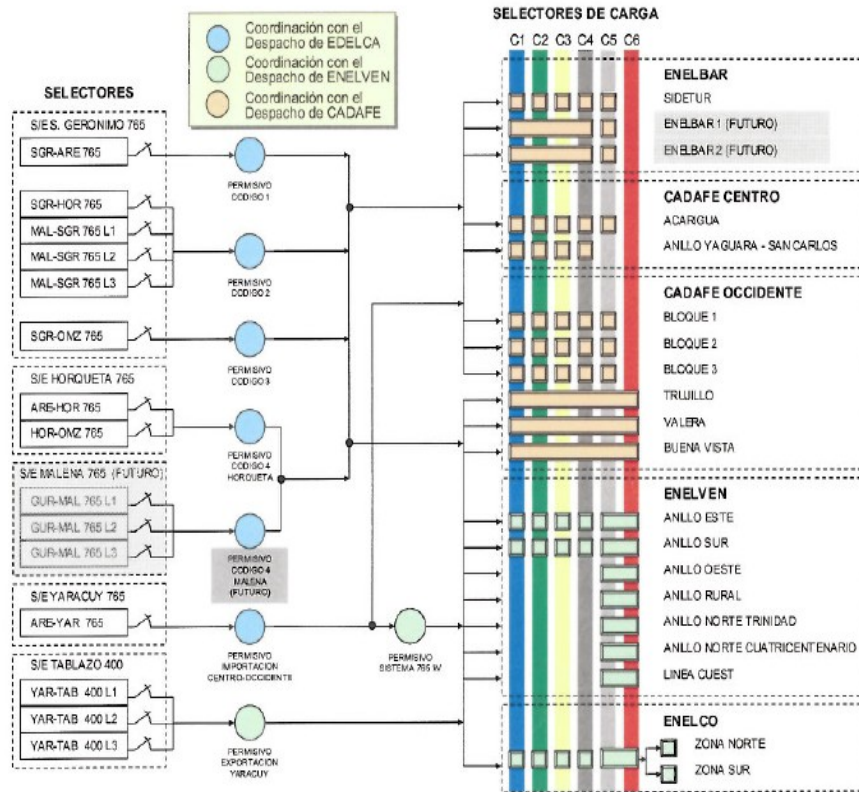


Figura 22: Diagrama funcional del Sistema de Desconexión de Carga del SEN

[20]

A efectos de simplificar el escenario de actuación del prototipo, se excluirán los bloques de Occidente en la simulación.

En el diagrama unifilar del SEN, mostrado en la Figura 23, se pueden observar los principales bloques que activará el prototipo cuando se produzca un disparo de línea por exceso de límite de transferencia.

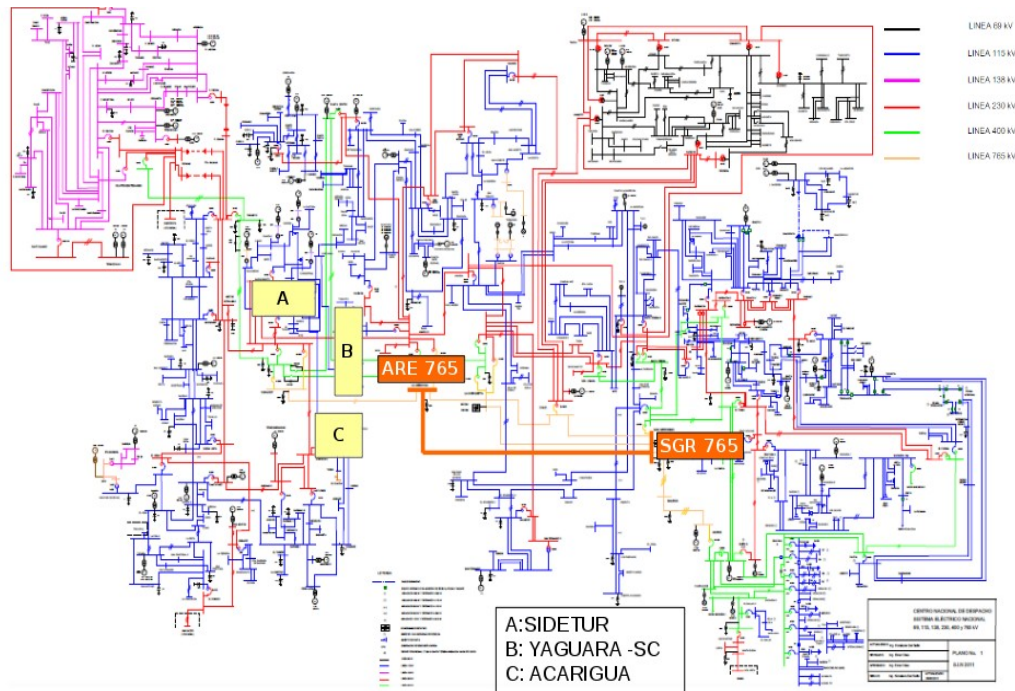


Figura 23: Esquema de actuación en el marco del unifilar del SEN

Se denotan los bloques de la siguiente forma y su potencia de carga nominal:

- Bloque A: Bloque de carga Sidetur (40 MW).
- Bloque B: Bloque de carga Acarigua (214 MW)
- Bloque C: Bloque de carga Anillo Yaguara – San Carlos (200 MW)

3.3. Diseño del prototipo TPR

El diseño del prototipo de teleprotección remota (TPR) consiste en utilizar una placa Arduino UNO R3 acoplada a un Shield Ethernet, el cual proporciona capacidades integradas para el mapeo de la capa física (PHY). Mediante este

procedimiento, en modo promiscuo, un algoritmo ubicado en el microprocesador ATMEGA 328 interpreta todas las señalizaciones presentes en la capa física mediante las funciones provistas por el chip W5100 con el fin de detectar y procesar las tramas de prioridad tipo GOOSE. Una vez detectada una función de actuación, el microprocesador activa un conjunto de relés que emiten las señales de tensión a los equipos responsables de realizar los disparos transferidos a las subestaciones que finalmente realizan las desconexiones de las cargas A, B y C. Se realiza el monitoreo de transferencia de potencia con un software de desarrollo propio y se analizan las tramas GOOSE mediante un analizador de protocolos. Se utiliza un algoritmo que genera datos aleatorios crecientes para simular el incremento de la potencia transmitida por la línea de ultra alta tensión

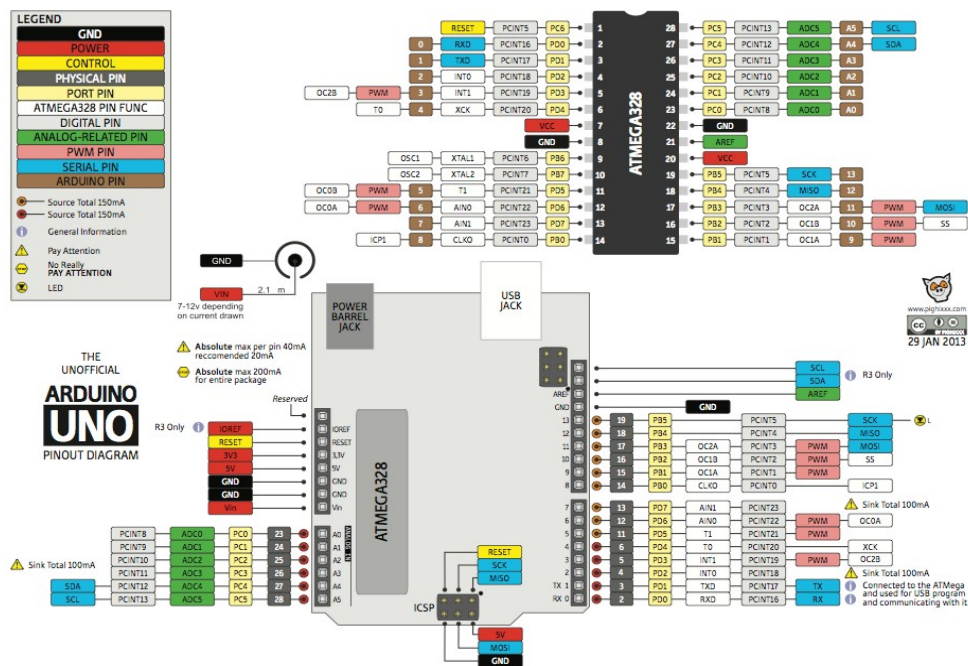
3.3.1. Hardware

- Placa Arduino UNO R3, con microprocesador programable ATMEGA328

Al realizar el acople con el Shiel Ethernet, los pines 10, 11, 12 y 13 son utilizados para la comunicación entre ambas placas.

Arduino utiliza un puerto serial para la comunicación en tiempo de programación. Durante la realización del programa fue posible verificar el contenido del buffer mediante esta vía.

En la Figura 24 se muestra la placa Arduino UNO R3, la leyenda en código de colores según las distintas funcionalidades y la descripción de los pines de control.



W5100 que es capaz de dar soporte para Stacks Protocols (Pilas de protocolos) como TCP, UDP, IPv4, ICMP, ARP, IGMP, PPPoE.

- 16 Relay Module

Modulo de 16 relés es utilizado para generar las señales de voltaje a los equipos responsables de enviar los disparos teletransferidos. Los relés pueden manejar tensiones alternas de 220V con un voltaje de control de 12V y una corriente continua de 15 a 20mA. En la Figura 26 se muestra su aspecto físico y los pines que conectan al Arduino UNO R3

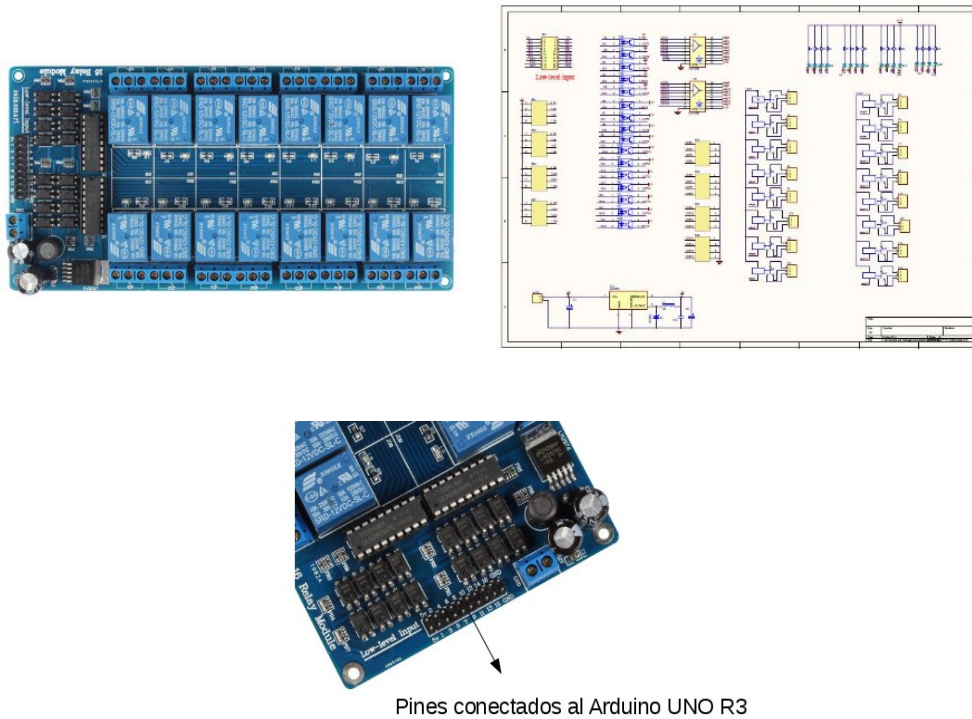


Figura 26: 16 Relay Module

[23]

El módulo se alimenta con un fuente de DC 12V y dispone de 16 relés que se conectan a la placa Arduino mediante unos pines de control a 5V de corriente continua.

3.3.1. Software

Para realizar las pruebas funcionales del Prototipo se requiere la utilización de varios tipos de Software:

- Para el prototipo se utilizará el IDE de Arduino como entorno de programación para el desarrollo del algoritmo de captura y control de tramas GOOSE.
- Para el monitoreo de transferencia de potencia de la línea de ultra alta tensión se utilizará un desarrollo propio en HTML5, PHP, y JavaScript combinado con el uso de librerías jQuery y HighCharts
- El servidor GOOSE que simulará el IED en la S/E San Gerónimo es una adecuación de un servidor GOOSE provisto por el conjunto de librerías libIEC61850.
- El servidor de demanda aleatoria es de desarrollo propio escrito en lenguaje C.
- El sistema operativo donde opera el servidor GOOSE, el sistema de monitoreo y analizador de protocolos es Slackware Linux, utilizando un servidor HTTP Apache 2.
- Para el análisis de las tramas GOOSE se utilizará el analizador de protocolos WireShark.

3.4. Análisis de tramas Ethernet tipo GOOSE bajo la norma IEC 61850

La velocidad de comunicación entre dispositivos electrónicos inteligentes

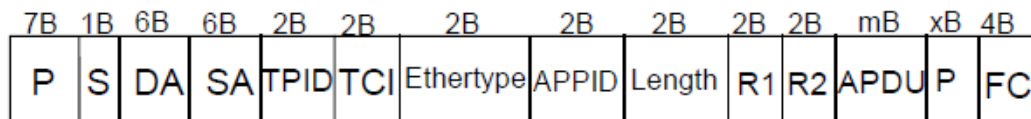
(IED) es un requisito indispensable para la comunicación mediante mensajes GOOSE de alta prioridad. El estándar presentado por la norma IEC 61850 establece los tipos de mensajes y clases de rendimiento.

Tipos de mensajes:

- 1a – Fast Messages – Trip
- 1b – Fast Message – Other
- 2 – Medium speed messages
- 3 – Low speed messages
- 4 – Raw data messages
- 5 – File transfer functions
- 6a – Time synchronisation messages a
- 6b – Time synchronisation messages b
- 7 – Command messages with access control

Para la implementación del Prototipo TPR se utilizarán mensajes tipo 1A “Trip”, los cuales son catalogados como los más rápidos e importantes dentro de la subestación.

El servidor GOOSE ubicado en la S/E San Gerónimo notificará con un mensaje tipo GOOSE 1A a la S/E La Arenosa. Si hay un disparo de línea por exceso de límite de transferencia, este mensaje de alta prioridad activará las desconexiones de carga correspondientes. A continuación, en la Tabla 3, se muestra la estructura de una trama GOOSE y en la Figura 27 el detalle de una sección de esta.

[illegible]

[25]

lo utilice para la sincronización.

- Limitador de Comienzo (S): Indica que la trama comienza a partir de él, su longitud es 1 byte.
- Dirección de destino (DA): Indica la dirección física (MAC) del dispositivo al que van dirigidos los datos.
- Dirección del emisor del mensaje (SA): Indica la dirección física (MAC) del dispositivo origen de los datos.
- Etiqueta de Identificación de Protocolo (TPID): Indica el tipo de trama Ethernet asignada para codificación de tramas Ethernet 802.1Q. El valor asignado es 0x8100 que permite identificar los mensajes Ethernet codificados para redes virtuales.
- Etiqueta de información de Control (TCI): Indica la prioridad del usuario, está establecida en la configuración y es utilizada para separar valores muestreados de baja prioridad (Samples Values). Consta de 3 campos, los primeros 3 bits corresponden a la prioridad de los mensajes determinada por el usuario. El nivel de prioridad va del 1 al 7 siendo este último el más alto. El CFI se denomina indicador de forma canónica, que para el estándar IEC 61850 debe ser cero. El octeto y medio restante es utilizado para el identificador de red virtual (VID), que debe ser configurado en la inicialización del sistema. Para el caso de no utilizar redes virtuales, debe configurarse como nulo.
- Ethertype indica con qué protocolo están encapsulados los datos que contiene el Payload, las tramas GOOSE se identifican con el valor hexadecimal 0x88B8.

- Identificador de aplicación (APPID): Se utiliza para seleccionar tramas que contienen mensajes GOOSE
- Longitud: Es el número de octetos, incluyendo la cabecera del APPID y la longitud del APDU. Debe ser menor a 1.492 bytes y la longitud total corresponde a $m+8$ bytes.
- Reservado 1 y 2: Reservados para uso futuro, se debe considerar 0x00.
- Protocolo de aplicación para la unidad de datos (APDU): El APDU de una trama GOOSE consiste en lo siguiente:
 - Número de estado (stNum): Se asigna este número cada vez un mensaje GOOSE cambia a consecuencia de un evento.
 - Número de secuencia (sqNum): Número asignado para aumentar la retransmisión de mensajes.
 - Modo de prueba (Test): Se establece si la trama está en modo de pruebas.
 - Tiempo permitido de vida (TAL): El tiempo máximo de vida del paquete después de la transmisión.
 - Revisión de configuración (confRev): Indica la versión del IED, tanto la del publicador como la del suscriptor deben ser las mismas.
 - Número de entradas del conjunto de datos (numDatSetEntries): Indica el número de datos presentes en el mensaje GOOSE recibido.
 - Referencia a bloque de control (gocbRef): Proporciona el nombre del bloque de control GOOSE.

- Conjunto de datos (Dataset): Proporciona el nombre del conjunto de datos en el IED GOOSE
 - Identificación GOOSE (Goid): Cada bloque GOOSE en cada IED tiene asociado un id diferente.
 - Marca de tiempo: Los mensajes vienen marcados con el tiempo exacto con el que fueron generados.
 - Datos (Data) Contiene la información del mensaje GOOSE.
-
- Relleno (P): Si es necesario se realiza un relleno de bytes
-
- Comprobado de secuencia de trama (FCS): Es utilizado para controlar la secuencia de las tramas.

El TPR mediante el chip W5100 estará capturando todas las tramas existentes en el medio físico, como los mensajes GOOSE se transmiten en modo MULTICAST, se deberá identificar en primer lugar si el IED publisher es de interés. Para esto es necesario analizar los bytes correspondientes a la MAC origen, pero como es el único IED presente en la red de simulación, se asumirá, por simplificación, que es válido.

El dato que se validará es el Ethertype, ya que gracias a este valor es posible determinar que efectivamente es una trama GOOSE. Por último se validarán el campo de identificación GOSSE (GOid) y el campo de datos (Data).

3.4.1. Desarrollo de algoritmos para publicación de mensajes GOOSE

Para generar los mensajes tipo GOOSE se utiliza una adaptación de un servidor GOOSE provisto en las librerías escritas en lenguaje C, libIEC61850.

Cumpliendo con el modelo establecido por la norma para los Sistemas de Automatización de Subestaciones (SAS), es necesario generar un archivo de extensión .ICD (IED Capability Description) el cual contendrá las características de los dispositivos relacionados con las funciones de comunicación y el modelo de datos. Este archivo está escrito en el lenguaje SCL (Substation Configuration Language) el cual está especificado en la norma IEC 61850.

Para la compilación del servidor GOOSE, se requieren fundamentalmente dos archivos:

- SGR765.icd
- servidorGOOSE.c

El conjunto de librerías libIEC61850 dispone de unas herramientas para generar el modelo SCL desde archivos ICD a librerías .h, para que el servidor GOOSE pueda interpretar la configuración.

Esta herramienta escrita en Java se denomina genmodel.jar y se ubica en `tools/model_generator/`

Para que el servidor GOOSE pueda compilarse correctamente se requiere la generación de las librerías provenientes del modelo ICD.

Un archivo Makefile lleva a cabo la gestión de la compilación, es utilizado por el compilador de C mediante el comando make en ambientes Unix/Linux.

para generar el modelo en tiempo de compilación, se realizará una adecuación al Makefile del servidor GOOSE y de esta forma se incluye el modelo SCL.

En la Figura 5 del capítulo I se observa un ejemplo de la estructura del modelo de datos IEC 61850 y su sistema jerárquico se muestra a continuación en la Figura 28.

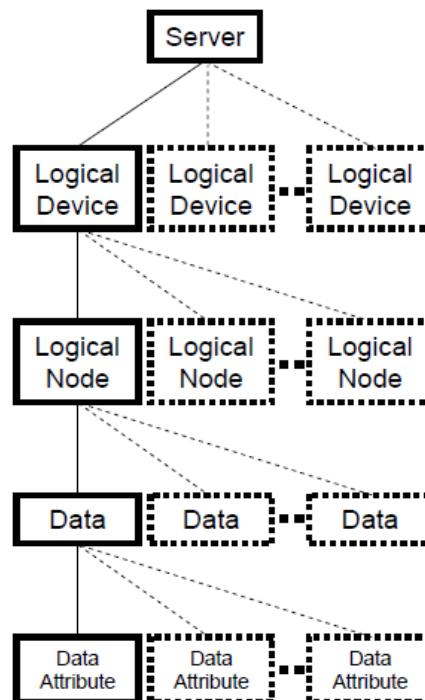


Figura 28: Jerarquía de objetos

[26]

En el archivo ICD se puede encontrar la definición de un Dispositivo Lógico (LD) llamado “GenericIO” mediante la etiqueta:

```
<LDevice inst="GenericIO">
```

Dentro de este dispositivo lógico se puede ubicar un Nodo Lógico (LN) llamado LN0

```
<LN0 lnClass="LLN0" lnType="LLN01" inst="">
```

A su vez, LN0 anidará dos datasets, uno será de tipo “Events” y el otro “AnalogValues”

```
<GSEControl appID="events" name="gcbEvents" type="GOOSE" dataSet="Events"
confRev="2" minTime="10" maxTime="3000" />
```

```
<GSEControl appID="analog" name="Voltaje" type="GOOSE" dataSet="AnalogValues"
confRev="2" />
```

Cuando el dataset es de tipo Events se debe declarar el tiempo mínimo “minTime” y el maxTime, la Figura 29 muestra que en condiciones normales la transmisión de mensajes se efectuará en cada maxTime y al momento de un evento la frecuencia de envío aumentará hasta que el tiempo mínimo entre dos mensajes sea minTime. La máxima frecuencia de envío corresponde al minTime.

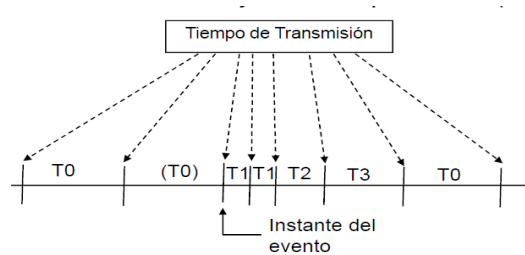


Figura 29: Tiempos de transmisión de mensajes GOOSE

[27]

El dataSet Events, en este caso, tendrá un valor de control “ctlVal” que será de tipo booleano. El estado de esta variable corresponde al estado general de la línea de transmisión SRG-ARE al asignarle el valor hexadecimal 0x00 (FALSE) el prototipo efectuará las desconexiones de carga selectivas correspondientes al código

C1. Nótese que otro atributo definido en el Data Object es la marca de tiempo. Es importante que la trama contenga esta marca para conocer el instante preciso en que se produce el evento a efectos de realizar auditorias posteriores.

```
<DOType id="SP1" cdc="SPC">
  <DA name="stVal" bType="BOOLEAN" fc="ST" dchg="true" />
  <DA name="ctlModel" type="CtlModels" bType="Enum" fc="CF" />
  <DA name="ctlVal" bType="BOOLEAN" fc="CO" />
  <DA name="t" bType="Timestamp" fc="ST" />
</DOType>
```

El conjunto de datos AnalogValues proveerá a la trama GOOSE los valores de tensión para cada fase de la línea mediante una etiqueta que describirá los atributos para cada DataObject mediante etiquetas DataAttribute.

```
<DOType id="A" cdc="MV">
  <DA name="fhA" bType="FLOAT32" fc="MX" />
  <DA name="fhB" bType="FLOAT32" fc="MX" />
  <DA name="fhC" bType="FLOAT32" fc="MX" />
  <DA name="fhN" bType="FLOAT32" fc="MX" />
  <DA name="Pot" bType="FLOAT32" fc="MX" />
</DOType>
```

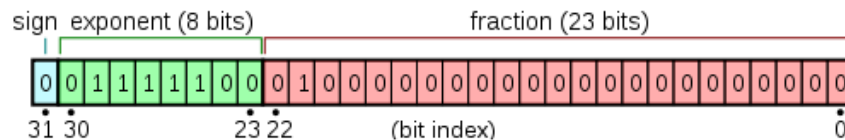


Figura 30: Representación coma flotante de 32bits, IEEE 754

[28]

bType es la propiedad que asigna el tipo de variable, en este caso los valores de tensión de cada fase serán de tipo coma flotante de 32 bits basados en el estándar IEEE 754.. La Figura 30, se muestra un ejemplo de esta representación. El primer bit se reserva para el signo, el exponente ocupa 8 bits y la parte fraccionaria 23 bits.

Con esta representación se consigue una precisión de 6 a 9 dígitos

significativos

$$(-1)^{b_{31}} \times (1 \cdot b_{22} \cdot b_{21} \dots b_0)_2 \times 2^{b_{30} b_{29} \dots b_{23} - 127}$$

$$(-1)^{sign} \times \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i}\right) \times 2^{(e-127)}$$

Respecto al código fuente del servidor GOOSE indicado anteriormente, se le realizaron unas adaptaciones para que pueda generar un conjunto de datos aleatorios crecientes, que simulen un aumento progresivo de la potencia demandada en el línea SGR-ARE a 765KV.

La traza de demanda de potencia aleatoria se escribirá en el archivo datos.dat mediante la función

```
void traza(float d)
{
    /* Se escriben los valores de potencia creciente aleatoria
    * en el archivo datos.dat
    */
    FILE *f;
    f = fopen("datos.dat", "w+");
    fprintf(f, "%f\n", d);
    fclose(f);
}
```

También se escribirá un archivo “activa.dat” que indicará el estado de la línea, si está en servicio o no. Esto se utiliza para que la herramienta de monitoreo muestre el estado de servicio de la línea y el registro de las trazas de potencia en el tiempo.

```
int linea_activa()
{
    /* Se escriben el estado de linea SGR-ARE en el archivo activa.dat */
    FILE *fp;
    char buffer[10];
    fp = fopen ( "/var/www/htdocs/PTRweb/activa.dat", "r" );
    fscanf(fp, "%s", buffer);
    printf("%s", buffer);
}
```

```

        if (buffer[0]=='1') return 1;
        else return 0;
        fclose ( fp );
    }

```

El servidor GOOSE colocará el contenido de las variables sensadas de cada fase y neutro (fhA, fhB, fhC y fhN) de la línea SGR-ARE a 765KV y el valor de la potencia transmitida, en este caso simuladas aleatoriamente, en mensajes GOOSE para que sea comunicada a otros IED que requieran la información dentro de la S/E San Gerónimo.

Las funciones que publican estos valores son las que se muestran a continuación:

```

/* Publicacion de los valores de voltaje y potencia*/
IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_fhA, fhA);
IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_fhB, fhB);
IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_fhC, fhC);
IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_fhN, fhN);
IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_Pot, Pot);

```

Si se supera el límite de transferencia de potencia establecido en 2.000MW entonces la función que dispara los eventos GOOSE de alta prioridad envía un mensaje GOOSE al Prototipo TPR ubicado en la S/E La Arenosa para que se efectúen las autodesconexiones de carga establecidas.

```

if (Pot>2000) {

IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_GenericIO_GGIO1_SPCS01_stVal,
0);

IedServer_updateUTCTimeAttributeValue(iedServer, IEDMODEL_GenericIO_GGIO1_SPCS01_t,

```

```
Hal_getTimeInMs();  
Pot=0;
```

En resumen, se generó un archivo .icd escrito en lenguaje SLC, posteriormente se modificó el Makefile que gestiona la compilación del servidor GOOSE para incluir el Data Model ICD escrito en SCL. Se modificó el código fuente del servidor GOOSE para publicar mensajes GOOSE con los valores aleatorios de potencia transferida y del estado del servicio de la línea.

3.4.2. Desarrollo de algoritmos para la interpretación de tramas en el Prototipo

Las tramas publicadas por el servidor GOOSE podrán ser observadas por todos los IED que formen parte de la subred, ya que se publicarán en modo multicast.

Una de las características más importantes de los mensajes GOOSE es que, por su importancia, deben tener prioridad absoluta y los equipos que los procesen deben sacrificar los protocolos utilizados en otras capas del modelo OSI en aras de ganar tiempo.

La Figura 31 muestra claramente como los mensajes GOOSE prescinden de cuatro pilas de protocolos del modelo OSI para ahorrar tiempo.

Un mensaje GOOSE no necesita tratamiento IP ni TCP, tampoco requiere pasar por las capas de sesión, presentación y aplicación para tener acceso a ellos.

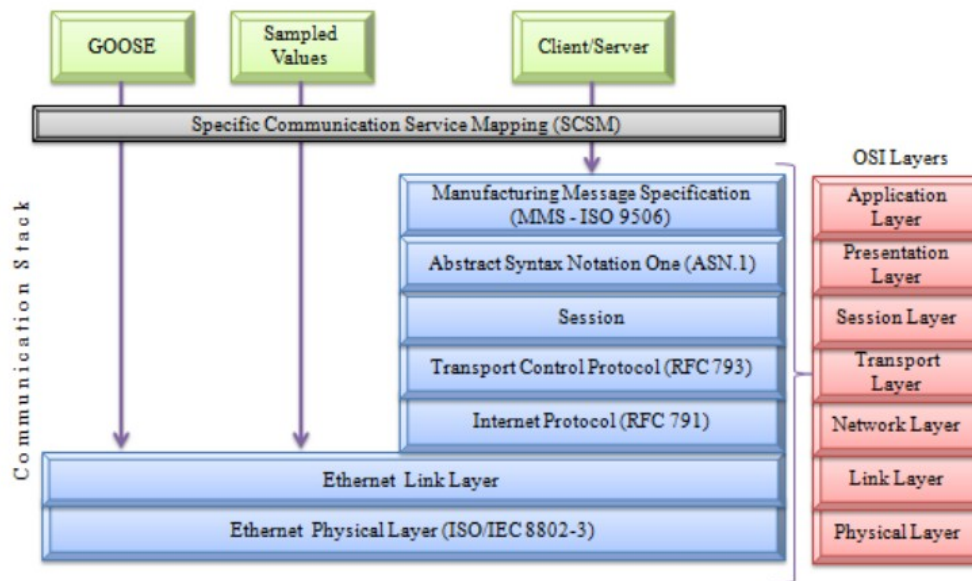


Figura 31: Pila de protocolos sobre el modelo de capas OSI

[29]

Para poder acceder a la capa física y de enlace, se utilizaron las funciones provistas en el chip W5100.

El código fuente, que se compila y aloja en la memoria flash del microprocesador ATMEGA328, se muestra a continuación.

```
#include <SPI.h>
#include <Ethernet.h>
#include <utility/w5100.h>
SOCKET s;
byte rbuf[800]; // rx buffer
int rbuflen; // longitud data
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0xC8, 0x4B };

void setup() {
  pinMode(9, OUTPUT);
  digitalWrite(9, LOW);
  pinMode(4, OUTPUT);
  digitalWrite(4, HIGH);
  W5100.init();
  W5100.writeSnMR(s, SnMR::MACRAW);
  W5100.execCmdSn(s, Sock_OPEN);
}

void loop() {
  rbuflen = W5100.getRXReceivedSize(s); //Inspeccionando tramas
```

```

        if (rbufllen > 0) {
            W5100.recv_data_processing(s, rbuf, rbufllen); //Captura de tramas en un
buffer
            W5100.execCmdSn(s, Sock_RECV);
            // Detección del mensaje GOOSE de alta prioridad
            if (rbuf[18]==0x88 && rbuf[19]==0xB8 && rbuf[rbuf[23]-4+21]==0x83 &&
rbuf[rbuf[23]-4
                +22]==0x01 && rbuf[rbuf[23]-4+23]==0x00)
            {
                digitalWrite(9, HIGH);
            }else if(rbuf[18]==0x88 && rbuf[19]==0xB8 && rbuf[rbuf[23]-4+23]==0xFF) {
                digitalWrite(9, LOW);
            }
        }
    }
}
void printGOOSE(int lon)
{
    int octeto=1;
    int longitud=0;
    for (int i=0; i<lon+2; i++) {
        if (longitud==40) {
            longitud=0;
            Serial.print("\n");
        }
        longitud++;
        if (rbuf[i]==0x88 && rbuf[i+1]==0xB8 && rbuf[rbuf[23]-4+23]==0xFF) {
            Serial.print(rbuf[rbuf[23]-4+23], HEX);
            if (rbuf[rbuf[23]-4+23]==0xFF) {
                digitalWrite(9, LOW);
            }else digitalWrite(9, HIGH);
        }
        if (octeto==2) {
            Serial.print(" ");
            octeto=1;
        }
        p(rbuf[i]);
        octeto++;
    }Serial.println("");
}
}

void p(char X) {
    char buf[9];
    sprintf(buf, "%04x", X);
    char *b=buf+2;
    Serial.print(b);
}
}

```

La primera parte del algoritmo realiza la inclusión de tres librerías necesarias:

```

#include <SPI.h>
#include <Ethernet.h>
#include <utility/w5100.h>

```

SPI.h: Esta librería proporciona funciones necesarias para visualizar datos por el puerto serial de la placa Arduino, para poder explorar y dar formato a las tramas capturadas se requiere fundamentalmente de la función Serial.print.

La conexión mediante el puerto serial permite transferir datos desde el Arduino a una herramienta en el entorno de programación llamada “Monitor Serie”. Se puede encontrar en Herramientas/Monitor Serie del IDE de Arduino.

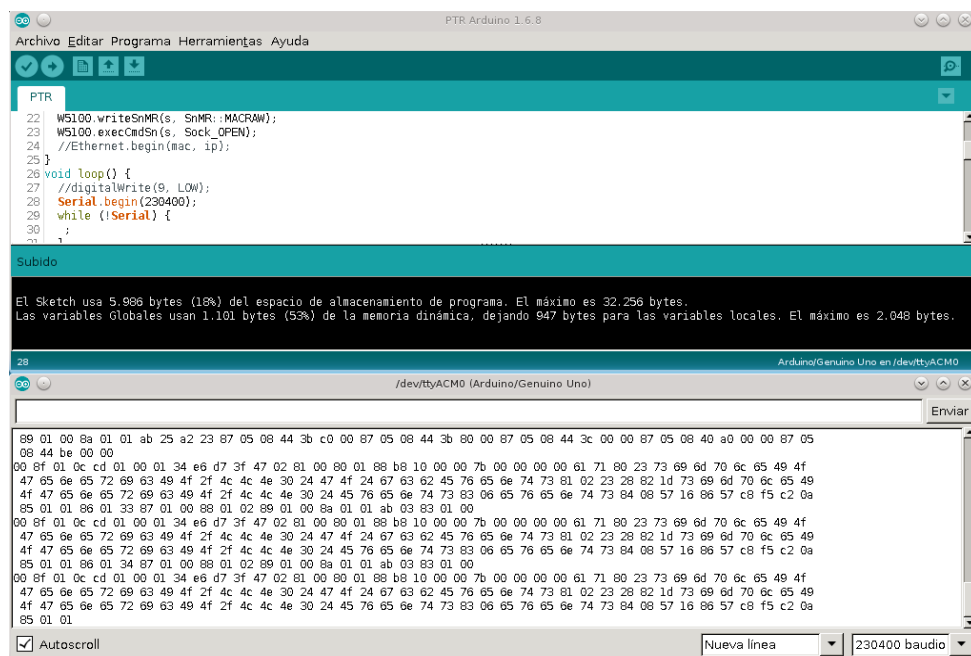


Figura 32: Monitor Serie

La segunda librería, ethernet.h, es la que nos permite acceder algunas funciones básicas de red, pero fundamentalmente otorga el acceso a las funciones reservadas de la tercera librería: w5100.h ubicada en utility. Esta última es clave para acceder a directamente a la capa de enlace del modelo OSI, y capturar todas las tramas presentes en la capa física.

```
W5100.writeSnMR(s, SnMR::MACRAW);
W5100.execCmdSn(s, Sock_OPEN);
```

Los métodos permiten abrir un Socket y acceder a los datos “Crudos”. Void loop realizará un bucle permanente en el que se inspeccionará la presencia de tramas en el medio. Mediante el método `getRXReceivedSize` se devuelve el número de bytes capturados almacenándolos en la variable `rbuf`, si es mayor a cero, entonces se captura la trama y se almacena en el bufer: `rbuf`.

```
void loop() {
  rbuflen=W5100.getRXReceivedSize(s);
  if (rbuflen>0) {
    W5100.recv_data_processing(s, rbuf, rbuflen);
    W5100.execCmdSn(s, Sock_RECV);
  }
}
```

Los datos almacenados estarán disponibles para realizar la validación. Solo se necesita procesar tramas de tipo GOOSE, por lo tanto todas las otras tramas serán descartadas.

Otro criterio de validación es el contenido de las variables del Dataset. En este caso se validará la trama que contiene la variable booleana de tipo Events, ya que al prototipo TPR le interesa saber el estado de servicio de la línea SGR-ARE a 765KV. Los valores analogValues, estarán disponibles para la utilización del IED de monitoreo.

De esta forma la siguiente instrucción validará, en primer lugar, que el EtherType sea de tipo GOOSE.

```
if (rbuf[18]==0x88 && rbuf[19]==0xB8 &&
    rbuf[rbuf[23]-4+21]==0x83 && rbuf[rbuf[23]-4+22]==0x01 && rbuf[rbuf[23]-
    4+23]==0x00)
```

En la Figura 33, se visualizan la cantidad de bits requeridos en cada sección de la trama.

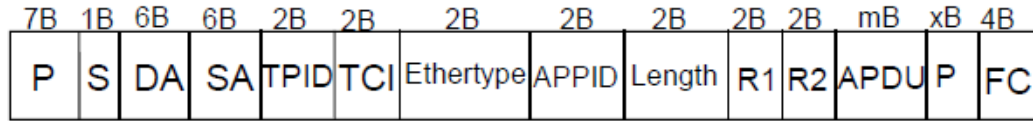


Figura 33: Estructura de una trama GOOSE

[30]

De la Tabla 4, se observa que, descartando los 7 octetos correspondientes al preámbulo, el primer octeto del Ethertype se ubica en la posición 18 y el segundo en la posición 19 del bufer.

Use	Ethertype value (hexadecimal)	APPID type
IEC 61850-8-1 GOOSE	88-B8	0 0
IEC 61850-8-1 GSE Management	88-B9	0 0
IEC 61850-9-2 Sampled Values	88-BA	0 1

Tabla 4: Valores para Ethertype

En la Figura 34 se observa que al Ethertype de protocolo GOOSE le corresponde el valor de 0x88B8

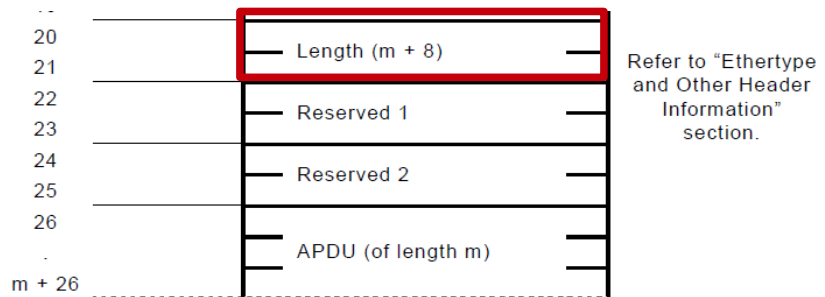


Figura 34: Detalle de Ethertype

[31]

Por lo tanto el primer criterio de validación será:

```
rbuf[18]==0x88 && rbuf[19]==0xB8
```

Para detectar el valor correspondiente al comando de tipo booleano enviado en el mensaje GOOSE, se debe explorar el contenido del APDU. Los dos octetos que contienen la información de la longitud del APDU

En el buffer, las posiciones correspondientes a los dos octetos de la longitud son la 22 y 23.

La longitud $m+8$ será un número entero representado por dos bytes. Esto significa que estos dos octetos tendrían un rango entero positivo de $2^{16} = 65.536$. Según la norma la longitud debe ser menor o igual 1.492 octetos. Para este trabajo, a modo de optimización, se tomó el valor del segundo octeto, ya que se comprobó en la práctica que el APDU es menor a $2^8=256$. Por lo tanto se toma el segundo byte, correspondiente a la longitud, para conocer el valor de m .

En resumen, los octetos correspondientes a la longitud:

```
rbuf[22] y rbuf[23]
```

y utilizando solamente el segundo octeto

```
rbuf[23]=m+8
```

donde m comienza en la posición 28 del bufer, esto significa que si se le suma el contenido de `rbuf[23] - 8`, se obtiene la posición final del APDU.

Posicion final APDU= 28 - rbuf[23] - 8

El APDU contiene la secuencia de datos indicado en la Figura 35

```
IECGoosePdu ::= SEQUENCE {  
    gocbRef          [0]    IMPLICIT VISIBLE-STRING,  
    timeAllowedtoLive [1]    IMPLICIT INTEGER,  
    datSet           [2]    IMPLICIT VISIBLE-STRING,  
    goID             [3]    IMPLICIT VISIBLE-STRING OPTIONAL,  
    t                [4]    IMPLICIT UtcTime,  
    stNum            [5]    IMPLICIT INTEGER,  
    sqNum            [6]    IMPLICIT INTEGER,  
    test             [7]    IMPLICIT BOOLEAN DEFAULT FALSE,  
    confRev          [8]    IMPLICIT INTEGER,  
    ndsCom           [9]    IMPLICIT BOOLEAN DEFAULT FALSE,  
    numDatSetEntries [10]   IMPLICIT INTEGER,  
    allData          [11]   IMPLICIT SEQUENCE OF Data,  
    security         [12]   ANY OPTIONAL,  
                        -- reserved for digital signature  
}
```

Figura 35: Secuencia de datos del APDU

[32]

El campo allData es el de interés, ya que en el se encuentra la variable booleana requerida para conocer el estado de la línea. El campo allData utiliza los últimos tres octetos de m y el último octeto corresponde al valor booleano 0x00 o bien 0xFF.

Por lo tanto, de esta forma, quedan definidas las posiciones de validación de la siguiente forma:

Posición final APDU= 28 - rbuf[23] - 8

Posición allData primer byte = 28 - rbuf[23] - 8 - 3

Posición de allData segundo byte = 28 - rbuf[23] - 8 - 2

Posición del contenido booleano= 28 - rbuf[23] - 8 - 1

En consecuencia, las reglas de validación serán las siguientes:

```
if (rbuf[28-rbuf[23]-8-1]==0x00) → Línea fuera de servicio  
if (rbuf[28-rbuf[23]-8-1]==0xFF) → Línea en servicio
```

Estas validaciones permitirán accionar los relés en el prototipo TPR para realizar la desconexión de carga.

Como se vio en el capítulo I, las placas Arduino disponen de unos pines que pueden asumir valores analógicos y digitales. En este trabajo se utilizarán los pines en modo digital. Los pines se pueden definir de entrada o salida (INPUT, OUTPUT)

Para definir un PIN en modo salida se debe ejecutar la siguiente instrucción:

```
pinMode(9, OUTPUT);
```

También es necesario especificar si un PIN es analógico o digital. Los pines digitales aceptan solamente dos valores definidos por las constantes LOW y HIGH.

Mediante la instrucción digitalWrite se puede asignar valores a los pines. Ejemplo:

```
digitalWrite(9, LOW);  
digitalWrite(9, HIGH);
```

La primera instrucción pondrá un voltaje igual a cero en el pin número nueve de la placa Arduino UNO R3 y la segunda un valor de +5V.

En el esquema de la Figura 36 se indican las conexiones de los pines de la

placa Arduino con el módulo de relés.

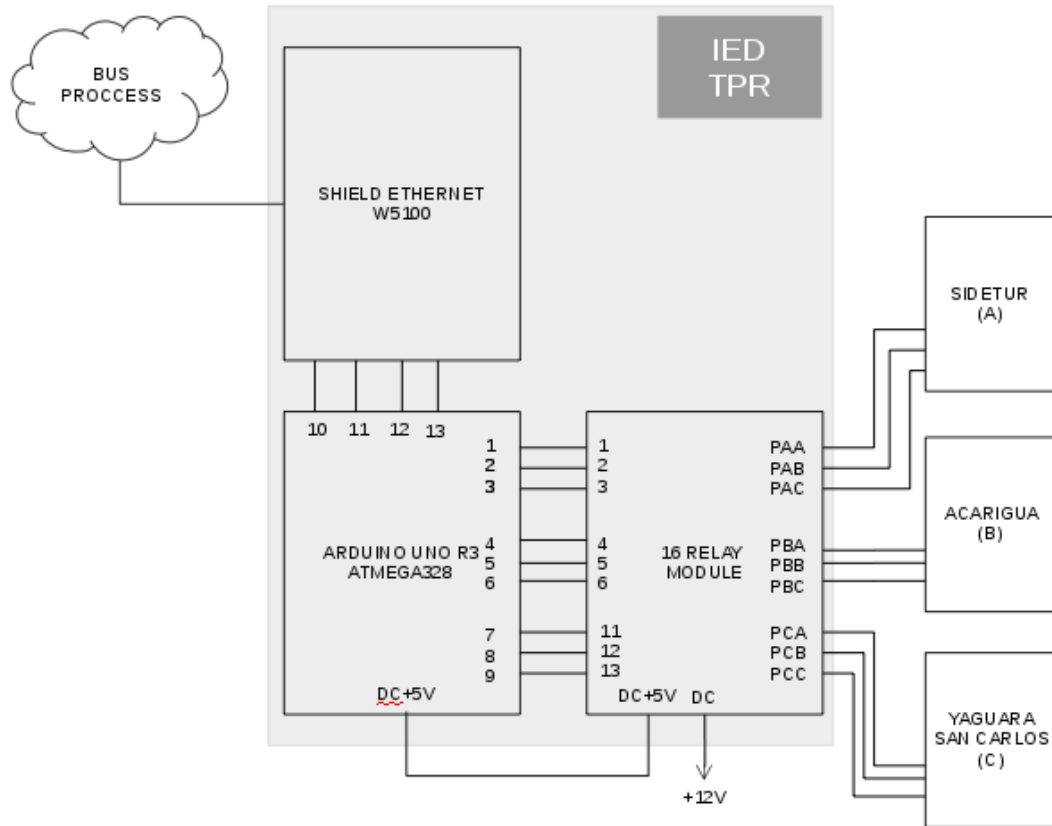


Figura 36: Esquema de conexión del prototipo TPR

Al capturar un mensaje GOOSE indicando un evento de desconexión de línea se coloca en valor alto el PIN número 9, para accionar el conjunto de relés que desconectan las cargas.

```
if (rbuf[28-rbuf[23]-8-1]==0x00) → Línea fuera de servicio
```

```
digitalWrite(9, HIGH);
```

Los relés pueden manejar tensiones de hasta 30 voltios en corriente continua con una circulación máxima de 10 Ampere. Estos voltajes, en un escenario real, pueden funcionar como señales para controlar la actuación de equipos de protección que no soporten el estándar IEC 61850.

El algoritmo que procesa el prototipo para la captura de tramas GOOSE se muestra en la Figura 37.

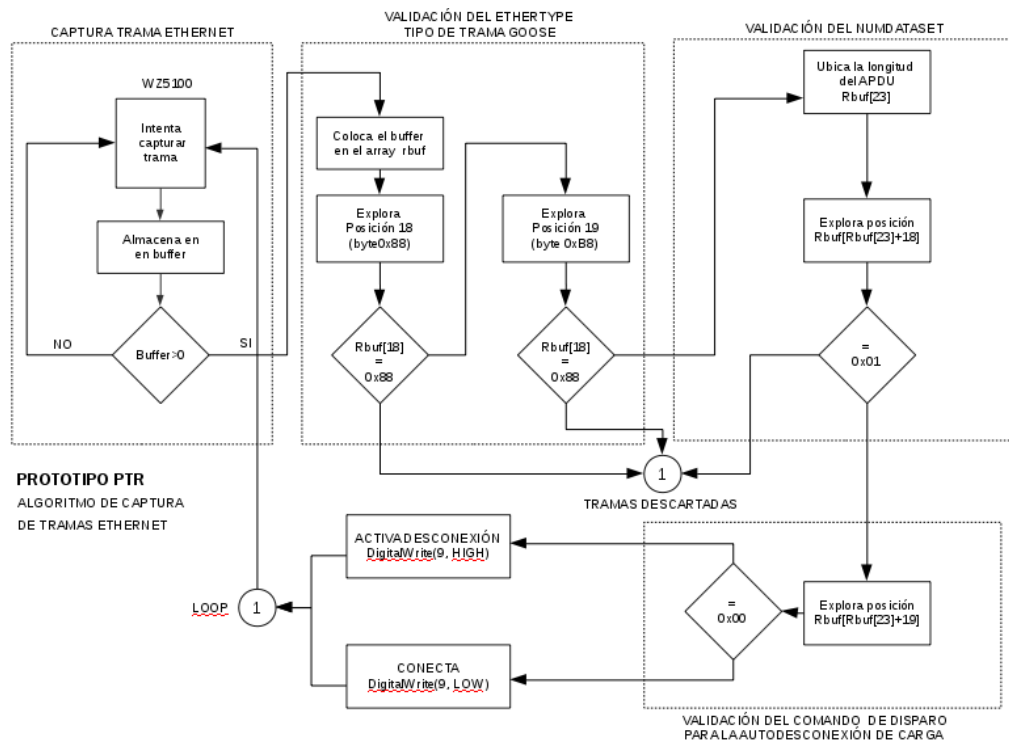


Figura 37: Algoritmo de captura de tramas GOOSE

3.4.3. Herramienta de monitoreo

En la Figura 38 se muestra la herramienta de monitoreo de la línea SGR-ARE a 765 kV.

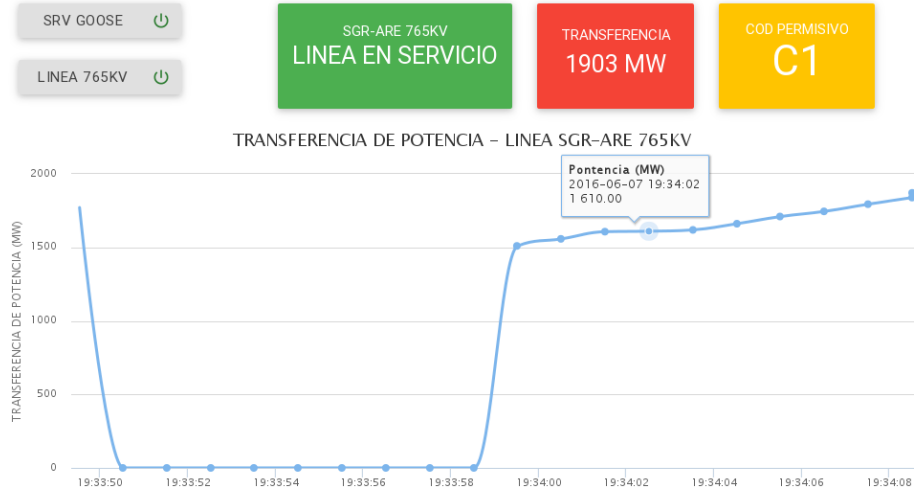


Figura 38: Interfaz de monitoreo para la simulación

Al comienzo de la simulación la línea se encuentra fuera de servicio, posteriormente se coloca en servicio y es posible visualizar la potencia transferida en el tiempo, al alcanzar el límite de transferencia de potencia, actúan las protecciones dejándola fuera de servicio y al mismo tiempo el servidor publica el mensaje GOOSE con el disparo transferido a la S/E ARE a 765 kV. Con el analizador de protocolos se pueden encontrar los dataset y conocer su contenido, tal como se muestra en la Figura 39.

[coloring Rule Name: 0x000001] [coloring Rule String: eth[0] & 1] - Ethernet II, Src: 201-242-160-223.local (34:e6:d7:3f:47:02), Dst: Iec-Tc57_01:00:01 (01:0c:cd:01:00:01) > Destination: Iec-Tc57_01:00:01 (01:0c:cd:01:00:01) > Source: 201-242-160-223.local (34:e6:d7:3f:47:02) - Type: 802.1Q Virtual LAN (0x8100) 802.1Q Virtual LAN, PRI: 4, CFI: 0, ID: 1 100 = Priority: controlled load (4) ...0 = CFI: canonical (0) 0000 0000 0001 = ID: 1 - Type: IEC 61850/GOOSE (0x88b8) - GOOSE - APPID: 0x1000 (4096) - Length: 122 - Reserved 1: 0x0000 (0) - Reserved 2: 0x0000 (0) - goosePdu - gocbRef: simpleIOGenericIO/LLN0\$G0\$gcbEvents - timeAllowedToLive: 30 - datSet: simpleIOGenericIO/LLN0\$Events - goID: events - t: Apr 18, 2016 03:52:06.640999972 UTC - stNum: 12 - sqNum: 1 - test: False - confRev: 2 - ndsCom: False - numDatSetEntries: 1 - allData: 1 item - data: boolean (3) - boolean: True	[coloring Rule Name: 0x000001] [coloring Rule String: eth[0] & 1] - Ethernet II, Src: 201-242-160-223.local (34:e6:d7:3f:47:02), Dst: Iec-Tc57_01:00:01 (01:0c:cd:01:00:01) > Destination: Iec-Tc57_01:00:01 (01:0c:cd:01:00:01) > Source: 201-242-160-223.local (34:e6:d7:3f:47:02) - Type: 802.1Q Virtual LAN (0x8100) 802.1Q Virtual LAN, PRI: 4, CFI: 0, ID: 1 100 = Priority: controlled load (4) ...0 = CFI: canonical (0) 0000 0000 0001 = ID: 1 - Type: IEC 61850/GOOSE (0x88b8) - GOOSE - APPID: 0x1000 (4096) - Length: 122 - Reserved 1: 0x0000 (0) - Reserved 2: 0x0000 (0) - goosePdu - gocbRef: simpleIOGenericIO/LLN0\$G0\$gcbEvents - timeAllowedToLive: 30 - datSet: simpleIOGenericIO/LLN0\$Events - goID: events - t: Apr 18, 2016 03:52:06.640999972 UTC - stNum: 11 - sqNum: 0 - test: False - confRev: 2 - ndsCom: False - numDatSetEntries: 1 - allData: 1 item - data: boolean (3) - boolean: False
---	---

Figura 39: Comparación de los dataset en tramas GOOSE

CAPÍTULO IV

SIMULACIÓN Y ANÁLISIS RESULTADOS

Se procedió a realizar el montaje según se muestra en el esquema de la Figura 36 del capítulo anterior.

El IED SRV, ubicado en la S/E San Gerónimo, es una computadora con sistema operativo Slackware/Linux de arquitectura x86 de 64bits, que cuenta con un procesador Core i7 y una versión de kernel 4.4.6 compilado para multiproceso.

El servidor (SRV) es una versión modificada del servidor GOOSE provisto en las librerías libIEC61850, escrito en lenguaje C, que consume un archivo ICD (IED Capability Description) escrito bajo el lenguaje SCL (Substation Configuration Language) el cual describe las capacidades de la subestación en la que se encuentra el servidor GOOSE.

El Prototipo TPR, se ubica en la S/E La Arenosa, y cuenta con un algoritmo básico de detección de bytes que explora las tramas en la capa de enlace del modelo OSI.

Se utiliza un Switch de 8 puertos que establece una red de comunicación LAN entre ambos IED simulando el BUS de procesos de la subestación.

En la simulación se desprecia el tiempo de conmutación de los switches en el bus de procesos. Se toma en cuenta el tiempo de procesamiento del prototipo TPR

y el tiempo aproximado de propagación de la señal en el tramo de 270 km correspondiente a la línea SGR-ARE a 765KV sobre fibra óptica.

La velocidad de transmisión de una trama en un medio óptico es aproximadamente un 30% inferior a la velocidad de la luz, es decir: 209.854 km/s, por lo tanto se estima que el tiempo de propagación en un medio de fibra óptica con una longitud de 270 km es aproximadamente 1,29 ms.

Las mediciones reales no pudieron ser efectuadas por tratarse de un medio de operación crítico, que no admite trabajar en ambiente de pruebas. Por lo tanto se asumen las consideraciones anteriormente expuestas relativas al tiempo.

4.1. Equipos utilizados

Para efectuar la simulación se simula un bus de procesos en la subestación interconectando los siguientes equipos en una red privada de área local:

1. Switch TP-LINK, 8 Puertos, 10/100Mbps, Modelo TL-SF1008D
2. Arduino UNO R3, Microprocesador ATmega 328
3. ShieldEthernet, Microchip ZW5100
4. Laptop DELL Latitud E7250, Core I7
5. Relay Module
6. Cables UTP Categoría 5e.

4.2. Esquema de montaje

El esquema de montaje se muestra en la Figura 40.

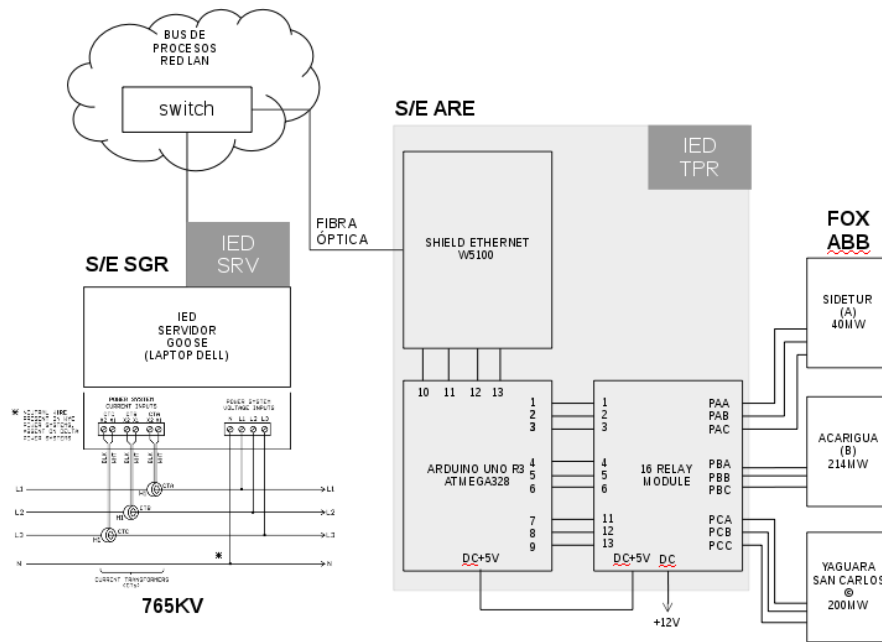


Figura 40: Esquema general de montaje

4.3. Procedimiento

Se coloca en servicio la interfaz de monitoreo y el analizador de protocolos en modo promiscuo. Posteriormente se inicia el servidor GOOSE (SRV) , el cual es simulado en la S/E SGR, realizando un muestreo del flujo de potencia existente en la línea mediante los transformadores de instrumentación (TC y TP) de la S/E SGR. Al detectar un incremento de la potencia transmitida cercano al límite establecido, se activa un código permisivo (C1) que habilita un selector y deja disponibles tres cargas: Sidetur (40 MW), Anillo La Yaguara-San Carlos (200 MW) y Acarigua (214 MW.) el servidor GOOSE SRV simula un incremento creciente de la transferencia de potencia de la línea hasta alcanzar el límite permitido, actúan las protecciones dejando fuera de servicio la línea y se emite un mensaje GOOSE de alta prioridad, que viaja por la fibra óptica en el bus de procesos, hasta ser recibido por el prototipo TPR en la S/E ARE 765 kV. El prototipo realiza la lectura y validación de la trama

para obtener su contenido. Al detectar una función de disparo, el prototipo realiza el accionamiento de los relés que controlan cada fase de las cargas. Los relés inyectan una corriente continua en los multiplexores de señales de comunicación y estos, a su vez, realizan un disparo teletransferido a las S/E que atienden cada una de las cargas, para finalmente desconectarlas del sistema y proteger equipos y personas.

La línea, al principio de la simulación, se encuentra fuera de servicio, posteriormente es energizada y su transferencia aumenta sostenidamente hasta alcanzar el límite de transferencia de potencia

En la Figura 41 se muestran los resultados de la simulación en una gráfica I/O del analizador de protocolos WireShark. El eje de las ordenadas representa la cantidad de tramas o paquetes y el de las abscisas, indica el tiempo en que transcurren.

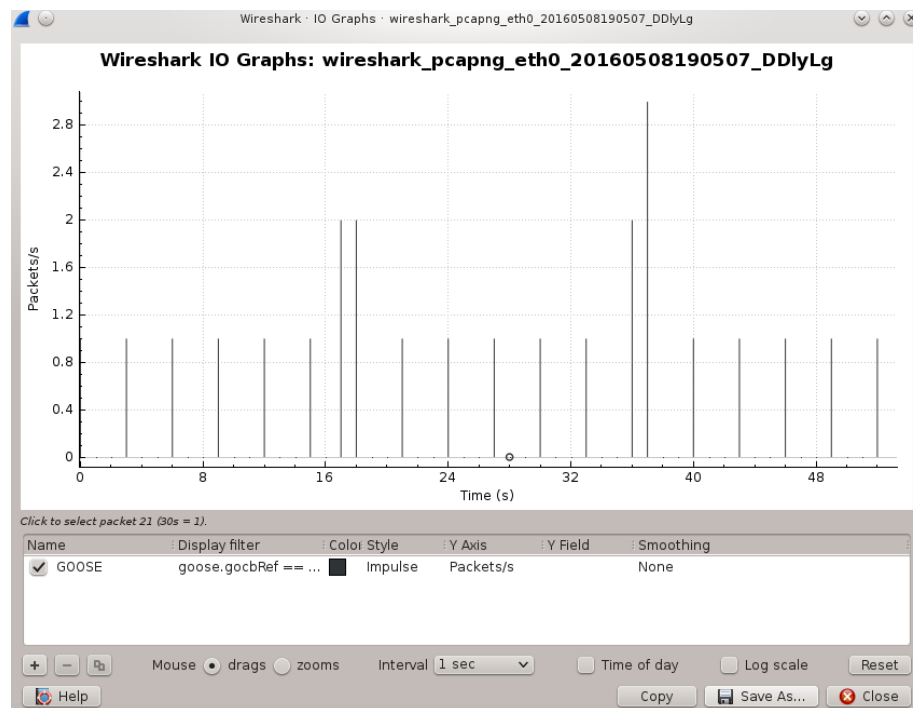


Figura 41: Ráfagas de tramas GOOSE

La Figura 42 amplia la anterior mostrando la secuencia de tramas periódicas emitidas por el servidor GOOSE.

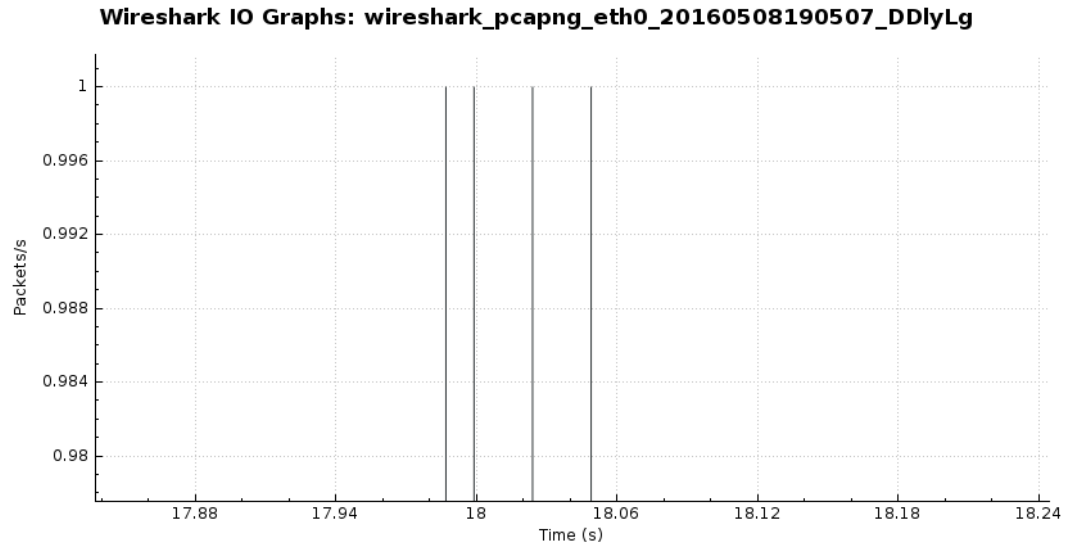


Figura 42: Activación de la Línea

Después del segundo 16, aparece la ráfaga de tramas GOOSE evidenciando la presencia de evento generado por la puesta en servicio de la línea SGR-ARE 765 kV.

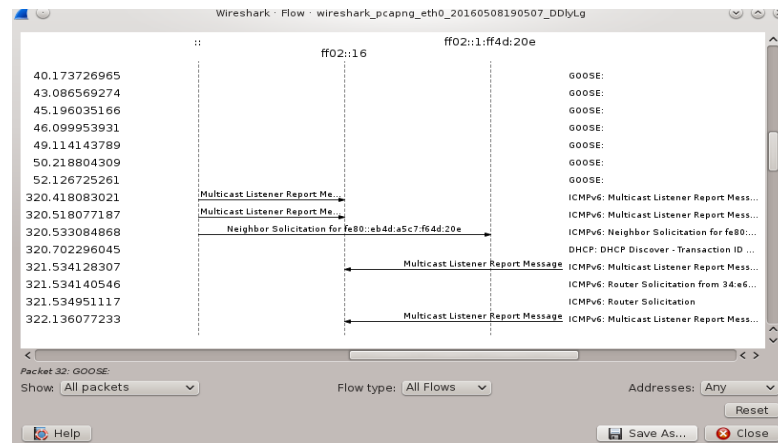


Figura 43: Flujo de Tramas

Después de este evento las tramas regularizan su frecuencia hasta el segundo

36, en el que dispara la línea por violación de límite de transferencia. La Figura 43 muestra el diálogo entre el servidor GOOSE y el prototipo TPR.

Un detalle del contenido de las tramas GOOSE se puede distinguir en la Figura 44. El contenido de la función de disparo es un valor booleano alojado en la sección jerárquica allData.

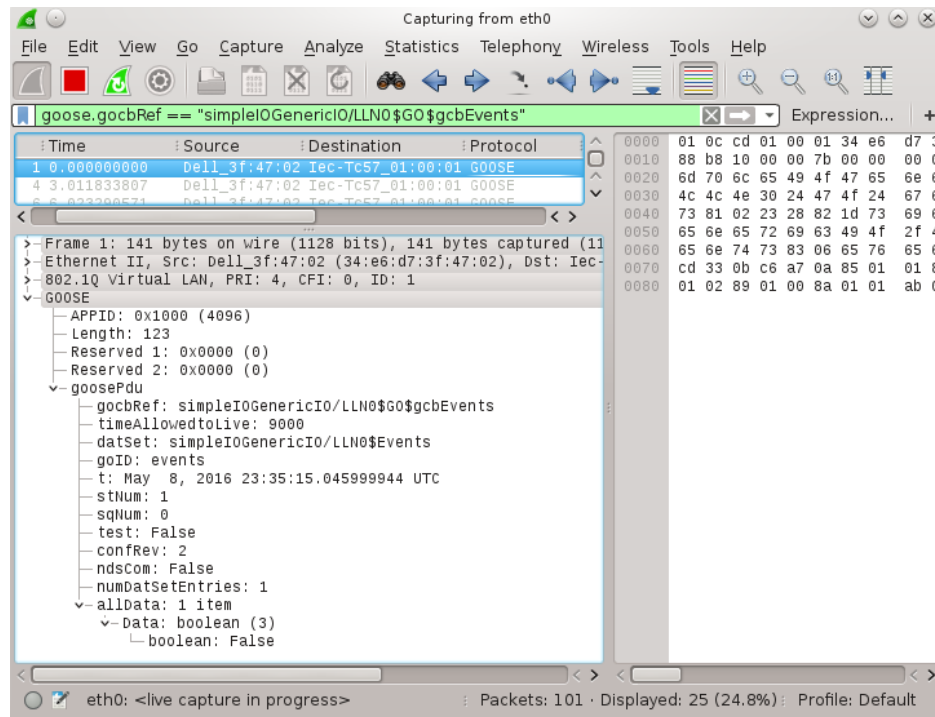


Figura 44: Detalle de la Trama

La sección 3.4.2. de este capítulo: “Desarrollo de algoritmos para la interpretación de tramas en el Prototipo”, muestra donde se aloja el contenido de la variable booleana que activa la función de disparo requerida:

Función de disparo en el APDU = 28 - rbuf[23] - 8

Ese valor, de tan solo un byte, activa la función de disparo.

En la Tabla 5, se muestran los resultados de la simulación sin confirmación de tramas. En estas mediciones de tiempo es posible observar que ante la ocurrencia de un evento que cambie el estado de la línea, la frecuencia de envío se incrementa notablemente llegando a alcanzar una separación en tiempo de las tramas por el orden de los 3 ms, pero al efectuarse un cambio en el contenido de la trama se produce un aumento en la frecuencia de envío de mensajes GOOSE.

Se realiza una corrida de la simulación observando solamente el comportamiento del servidor GOOSE, practicando el procedimiento indicado anteriormente:

1. Línea fuera de servicio.
2. Línea en servicio.
3. Línea fuera de servicio (por excedencia del límite).

Se obtienen los resultados obtenidos en la Tabla 5 que se muestra a continuación:

SGR-ARE 765KV		LENGTH	TIME (s)	TIMEATOLIVE	STNUM	SQNUM	DATA	TDIF (ms)
FUERADE SERVICIO	1	123	0,0000000000	9000	1	0	0X00	-
	2		3,0118338070			1		3011,83
	3		6,0232905710			2		3011,46
	4		9,0362328970			3		3012,94
	5		12,0491172230			4		3012,88
	6		15,0624033360			5		3013,29
EN SERVICIO	7	122	17,9872163970	30	2	0	0XFF	2924,81
	8	17,9996273070	1			12,41		
	9	18,0247540850	2			25,13		
	10	123	18,0498474430	3		25,09		
	11		21,0636714310	4		3013,82		
	12		24,0763257920	5		3012,65		
	13		27,0895839960	6		3013,26		
	14		30,1030271570	7		3013,44		
	15		33,1168441650	8		3013,82		
	16		36,1305344240	9		3013,69		
FUERADE SERVICIO	17	122	36,9921250040	30	3	2	0X00	861,59
	18	37,0096544180	3			17,53		
	19	37,0347640820	1			25,11		
	20	123	37,0598718520	2		25,11		
	21		40,0733972390	3		3013,53		
	22		43,0865692740	4		3013,17		
	23		46,0999539310	5		3013,38		
	24		49,1141437890	6		3014,19		
	25		52,1267252610	7		3012,58		

Tabla 5: Tiempos de las ráfagas de las tramas

En los anexos se encuentra el código fuente del archivo ICD, escrito en SCL, en el cual se puede observar el tiempo máximo y mínimo configurado para la secuencia de tramas GOOSE.

```
<GSEControl appID="events" name="gcbEvents" type="GOOSE" dataSet="Events" confRev="2"
minTime="10" maxTime="3000" />
```

En el IEC 61850-5 se define el tiempo de transmisión y los tiempos internos que lo componen. La Figura 45 muestra esta definición.

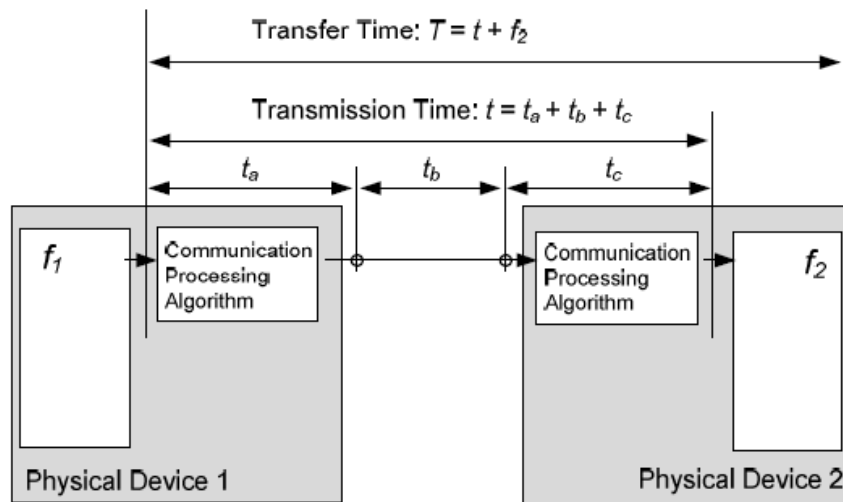


Figura 45: Definición del tiempo de transmisión (tomado de IEC 61850-2)

Es preciso notar que el tiempo de transmisión no contempla la función correspondiente al tratado del contenido del mensaje, en cambio el tiempo de transferencia si lo hace.

El estándar establece el tiempo óptimo que debería tener cada tipo de mensaje. En la Tabla 6 se observa el detalle de los tiempos de transmisión según su aplicación.

IEC 61850 MESSAGE TYPES AND PERFORMANCES

Type	Applications	Performance Class	Requirements (Transmission Time)
1A	Fast Messages (Trip)	P1	10 ms
		P2/P3	3 ms
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)

Tabla 6: Tipo de mensaje y de las tramas

La simulación se realiza para un mensaje GOOSE tipo A1 con desempeño P1.

Daqing Hou y Dave Dolezilek, escriben en la publicación “IEC 61850 – What It Can and Cannot Offer to Traditional Protection Schemes” que el tiempo t_b lo consideran en el orden de los 0,1ms. En el presente trabajo se asumirá ese valor para el tiempo establecido entre los IED y los switches de fibra óptica. Entre los switches de fibra óptica el tiempo será el calculado al principio del capítulo III, de 1,29 ms. para una distancia de 270 km.

En la Figura 46 se visualizan 43 muestras producto de la simulación de dos eventos de alta prioridad GOOSE, en color amarillo se resalta la diferencia de tiempo capturada en el analizador de protocolos y que representa el tiempo de transferencia

(Transfer Time) de la Figura 45 mostrada anteriormente.

No.	Time	Source	Destination	Protocol	Length	DT
1	0.000000	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
2	0.000009	Del_3t4792	Iec-T57_01:00:01	GOOSE	180	
3	0.009780	Selektro_e6d73f	008f013cxd01	0x4702	325	9,7212
4	3.011975	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
5	3.016571	Selektro_e6d73f	008f013cxd01	0x4702	143	4,5958
6	5.019834	Del_3t4792	Iec-T57_01:00:01	GOOSE	180	
7	6.023969	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
8	6.028570	Selektro_e6d73f	008f013cxd01	0x4702	143	4,6004
9	9.035788	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
10	9.040331	Selektro_e6d73f	008f013cxd01	0x4702	143	4,5426
11	10.039367	Del_3t4792	Iec-T57_01:00:01	GOOSE	180	
12	12.046455	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
13	12.051054	Selektro_e6d73f	008f013cxd01	0x4702	143	4,5998
14	14.079765	Del_3t4792	Iec-T57_01:00:01	GOOSE	140	
15	15.006969	Del_3t4792	Iec-T57_01:00:01	GOOSE	140	
16	15.032044	Del_3t4792	Iec-T57_01:00:01	GOOSE	140	
17	15.057124	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
18	15.057134	Del_3t4792	Iec-T57_01:00:01	GOOSE	180	
19	18.068755	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
20	20.077529	Del_3t4792	Iec-T57_01:00:01	GOOSE	180	
21	21.081741	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
22	24.092984	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
23	25.096747	Del_3t4792	Iec-T57_01:00:01	GOOSE	180	
24	27.104703	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
25	30.116773	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
26	30.116782	Del_3t4792	Iec-T57_01:00:01	GOOSE	180	
27	33.128544	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
28	34.988054	Del_3t4792	Iec-T57_01:00:01	GOOSE	40	
29	34.993526	Selektro_e6d73f	008e013cxd01	0x4702	42	4,5728
30	35.010914	Del_3t4792	Iec-T57_01:00:01	GOOSE	40	
31	35.015406	Selektro_e6d73f	008e013cxd01	0x4702	42	4,4920
32	35.036004	Del_3t4792	Iec-T57_01:00:01	GOOSE	40	
33	35.040480	Selektro_e6d73f	008e013cxd01	0x4702	42	4,4757
34	35.061107	Del_3t4792	Iec-T57_01:00:01	GOOSE	41	
35	35.065754	Selektro_e6d73f	008f013cxd01	0x4702	43	4,6478
36	35.136455	Del_3t4792	Iec-T57_01:00:01	GOOSE	80	
37	38.073445	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
38	38.077979	Selektro_e6d73f	008f013cxd01	0x4702	143	4,5340
39	40.157233	Del_3t4792	Iec-T57_01:00:01	GOOSE	180	
40	41.086305	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
41	41.090797	Selektro_e6d73f	008f013cxd01	0x4702	143	4,4929
42	44.097001	Del_3t4792	Iec-T57_01:00:01	GOOSE	141	
43	44.101579	Selektro_e6d73f	008f013cxd01	0x4702	143	4,5778

Tiempo promedio
de procesamiento TPR
4,5ms

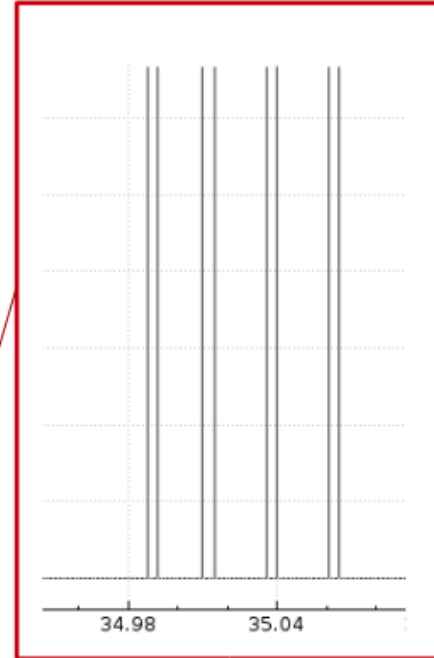


Figura 46: Resultados de la Simulación

Se promediaron las respuestas de las tramas verificando que estén dentro del tiempo establecido en la IEC 61850-5.

Se utilizarán las siguientes suposiciones, de acuerdo a la Figura 45, el tiempo entre switches y el tiempo de transmisión por fibra óptica:

El tiempo del algoritmo de procesamiento para a la comunicación es a nivel de hardware en la interfaz de red, por lo tanto se asume que ambas interfaces manejan aproximadamente los datos a la misma velocidad y tiempo.

$$t_a \approx t_c \quad (1)$$

Por otro lado, según las consideraciones expuestas por Daqing Hou y Dave Dolezilek para t_b

$$t_b \approx 0,1 \text{ ms} \quad (2)$$

Del tiempo de la fibra óptica, según las apreciaciones de propagación en 270 km expuestas con anterioridad, resulta:

$$t_{\text{fibra optica}} \approx 1,29 \text{ ms} \quad (3)$$

De la definición de los tiempos según la norma IEC 61850 de la Figura 45 se obtiene:

$$\text{Transfer Time} : T = t + f_2 \quad (4)$$

$$\text{Transmission Time} : t = t_a + t_b + t_c \quad (5)$$

Siendo

$$t_b = t_{b'} + t_{\text{fibra optica}} \quad (6)$$

El tiempo promedio de procesamiento obtenido en la Figura 46, como resultado de la simulación, puede escribirse como:

$$4,5 = 2(t_a + t_b + t_c + f_2) \quad (7)$$

Por lo tanto

$$4,5 = 2(t_{b'} + 2t_c + f_2) \quad (8)$$

Despejando:

$$t_c = 1,075 - f_2/2 \quad (9)$$

Como $t_c > 0 \Rightarrow f_2 < 1,075 \text{ ms}$

No se conoce el valor t_c ni de f_2 pero se sabe que la suma de ambos no puede ser mayor a 1,075 ms.

Para tomar el caso más extremo del tiempo de transmisión se escoge $f_2 = 0$. Esto equivale a decir que la velocidad de procesamiento que utiliza el prototipo en el algoritmo de lectura de la trama es prácticamente nulo y el algoritmo de procesamiento en la interfaz de red es máximo.

$$t_c = 1,075 \text{ ms}$$

De la Tabla 6, se observa que el tiempo de transmisión requerido para un mensaje GOOSE tipo 1A, clase P1, es de 10ms.

Sustituyendo en la ecuación (5) el valor obtenido de t_c e incorporando el tiempo de propagación de la señal en el tramo de fibra óptica de 270 km, resulta:

$$\text{Transmission Time} : t = t_a + t_b + t_{\text{fibra óptica}} + t_c$$

$$\text{Transmission Time} : t = 2 \times 1,075 + 0,1 + 1,29$$

$$\text{Transmission Time} = 3,54 \text{ ms}$$

El resultado obtenido cumple con las exigencias del estándar IEC 61850 especificado en la Tabla 6, ya que el tiempo de transmisión entre el servidor GOOSE y el prototipo, en su caso más desfavorable se encuentra 64,6% por debajo del máximo permitido.

4.4. Fiabilidad del Prototipo

4.4.1. Muestreo en diferentes condiciones de espacio y tiempo

Para analizar la fiabilidad del prototipo, se realizaron diez muestras del tiempo de respuesta en diferentes horas del día y espacios físicos, buscando generar condiciones adversas de humedad y temperatura en el ambiente que puedan influir en el dispositivo y alterar su comportamiento. Los resultados obtenidos se pueden consultar en la tabla 7 y la representación gráfica de la dispersión de la muestra en la figura 47.

Muestra	Tiempo (s)	Transmission time (s)
1	4,55	3,57
2	4,61	3,60
3	4,53	3,56
4	4,58	3,58
5	4,53	3,56
6	4,61	3,60
7	4,60	3,59
8	4,48	3,53
9	4,52	3,55
10	4,50	3,54

Margen de tolerancia	6,41
Promedio	3,57
Máximo	3,60
Mínimo	3,53
Desviación estándar	0,05
Desviación Máxima	0,02
Desviación Mínima	0,04

Tabla 7: Análisis de dispersión de la muestra

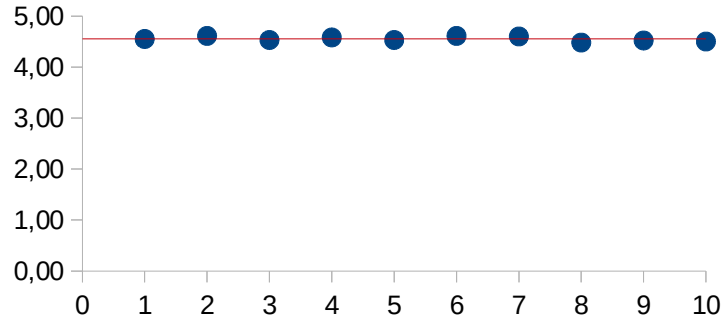


Figura 47: Gráfica de dispersión de la muestra

Se obtiene un margen de tolerancia en el tiempo de transmisión aproximadamente de 162 veces el valor de la desviación estándar, esto significa que la dispersión es baja y las variaciones que presenta el prototipo ante diferentes condiciones de temperatura y humedad no son significativas.

4.4.2. Muestreo con presencia de tráfico en la red

Con la herramienta de software PackETH se procedió a realizar una contaminación del medio con tramas Ethernet enviadas a distintas velocidades para estudiar la variación en el tiempo de respuesta.

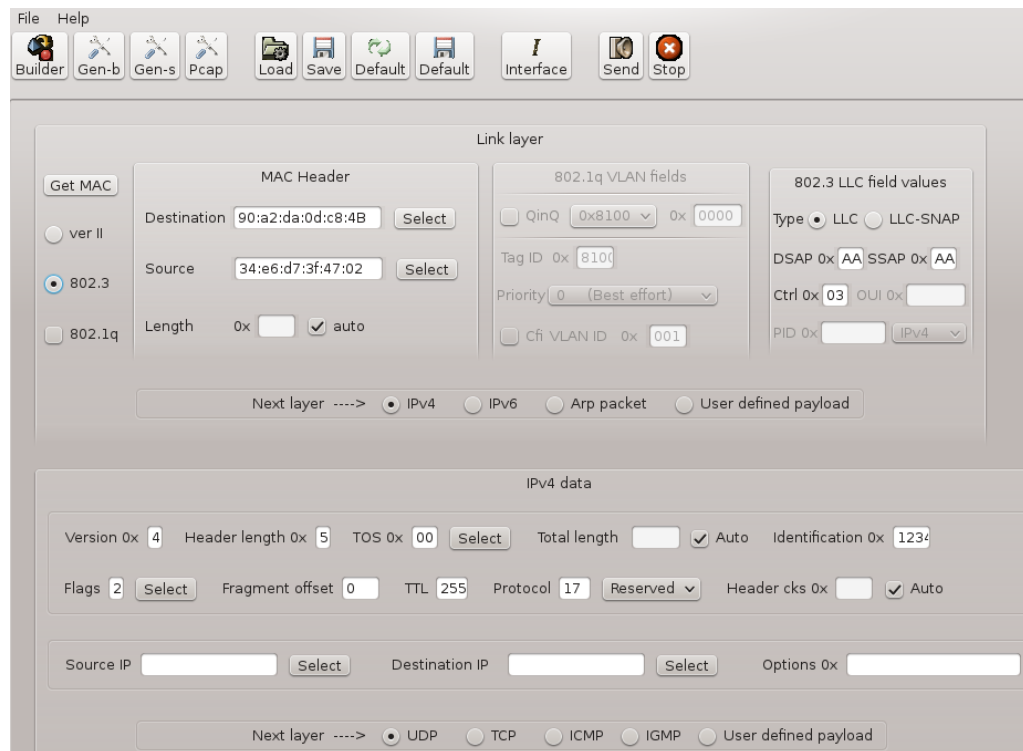


Figura 48: GUI herramienta de software PackETH

Los resultados se pueden apreciar en la figura 49.

Al contaminar el medio con tramas desde 10 kbps a 430 kbps se observó que a partir de las ráfagas a 200 kbps el tiempo de respuesta comenzó a incrementarse.

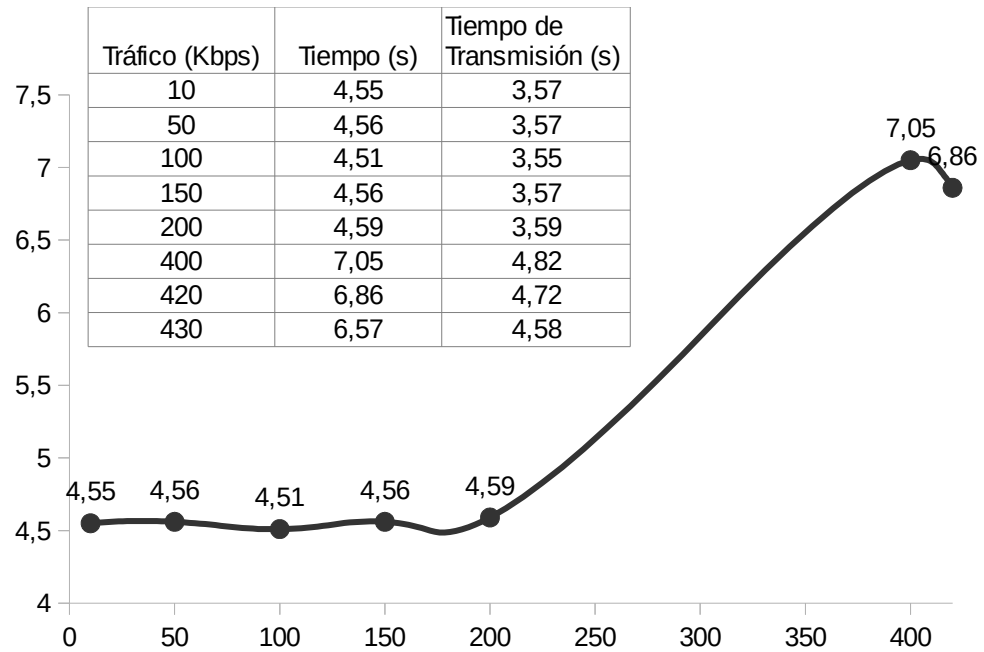


Figura 49: GUI herramienta de software PackETH

Es importante observar que a pesar de que se evidencia un incremento en el tiempo de transmisión (4,82 ms para el tiempo máximo) este no es representativo frente al valor recomendado en el estándar (10 ms). Por lo cual las pruebas de contaminación del medio arrojan resultados favorables.

CONCLUSIONES

El diseño del prototipo de protección y teleprotección, en el contexto de este trabajo, muestra un caso de uso real en el Sistema Eléctrico Nacional Venezolano, evidenciando la potencialidad y posibilidad de interoperabilidad que puede ofrecer el estándar IEC 61850 en procesos de actuación crítica.

La creación de archivos que describen los equipos que trabajan dentro y fuera de una subestación posibilitaron la creación de un modelo jerárquico en una capa abstracta que facilita la integración e interoperabilidad entre los dispositivos. En este trabajo se configuró un archivo ICD como parte de la descripción de la configuración de subestaciones (SCD) la cual se logró con la limitación propia del alcance del proyecto.

No fue posible realizar las pruebas en los laboratorios de CORPOELEC a causa de la emergencia eléctrica presentada por los bajos niveles de los embalses afectados por el cambio climático, ni se pudo simular sobre el medio de fibra óptica por razones de seguridad. Sin embargo, fue posible determinar en la simulación parte de uno de los objetivos específicos más importantes: El tiempo de procesamiento del TPR, concluyendo que se ubica dentro de los rangos de tiempo de transmisión aceptables para la operación.

Se comprobó que el prototipo diseñado cumple con las exigencias del estándar IEC 61850 en cuanto al tiempo de transmisión

RECOMENDACIONES

Si bien el tiempo de procesamiento cumple con las exigencias del estándar, el prototipo no está completamente parametrizado para cumplir con todas las demás exigencias que se requieren para cumplir con la certificación de la norma. Por lo tanto es importante cumplir con las demás exigencias y aumentar las capacidades y funcionalidades de acuerdo a las necesidades planteadas.

Se recomienda ampliar la investigación con la utilización de otros microprocesadores programables y diseñar la electrónica necesaria para impulsar la fabricación de dispositivos que atiendan las necesidades futuras del Sistema Eléctrico Nacional, tanto a nivel académico como industrial.

Para mensajes GOOSE rápidos (Trip) de tipo A1, clase P2/P3 el prototipo podría cumplir con los requisitos si los IED se ubican dentro de la subestación.

REFERENCIAS BIBLIOGRÁFICAS

- [1] S. Blair, Universidad de Strachclyde, Glasgow, Escocia. Rapid-prototyping protection schemes with IEC 61850.
- [2] T.Perez, D. Montoya, P. Tremante y R. Diaz, Caracas, Memorias del II Seminario de Modelos y Modelado : Conceptos, técnicas y aplicaciones, 2014
- [3] Siemens [En línea] <<http://www.energy.siemens.com>> [Consulta 2016]
- [4] Hirschmann [En línea <www.hirschmann.com> [Consulta 2016]
- [5] International Journal of Computer Applications (0975 – 8887), Vol 93 – No 16, Mayo 2014
- [6] ABB Group - Automation and Power Technologies [En línea] <<http://www.abb.com/media/454>> [Consulta 2016]
- [7] ABB Group - Automation and Power Technologies [En línea] <<http://www.abb.com/media/123>> [Consulta 2016]
- [8] IEC 61850. International Electrotechnical Commission. Communications networks and systems in substations.IEC 61850-8-1:2004 p.p 19
- [9] IEC 61850. International Electrotechnical Commission. Communications networks and systems in substations.IEC 61850-8-1:2004 p.p 19
- [10] Arduino [En línea] <www.arduino.cc/en/Reference/56> [Consulta 2016]
- [11] Arduino [En línea] <www.arduino.cc/en/Reference/45> [Consulta 2016]
- [12] Arduino [En línea] <www.arduino.cc/Reference/789> [Consulta 2016]
- [13] Arduino [En línea] <www.arduino.cc/Reference/654> [Consulta 2016]
- [14] Computer Science & Engineering University of Washington [En línea], <<http://www.cs.washington.edu>>
- [15] Arduino [En línea] <www.arduino.cc/Reference/12> [Consulta 2016]

- [16] Arduino [En línea] <www.arduino.cc> [Consulta 2016]
- [17] IEEE [En línea] <<http://www.ieee802.org>> [Consulta 2016]
- [18] WIZNET [En línea] <<http://www.wiznet.co.kr> [En línea]>
- [19] Arduino [En línea] <www.arduino.cc/Reference/789> [Consulta 2016]
- [20] W. Vivas Implementación del Esquema de Desconexión de Carga del SEN con selectividad. Opsis. Caracas. 2008
- [21] Arduino [En línea] <www.arduino.cc/reference/34> [Consulta 2016]
- [22] Arduino [En línea] <www.arduino.cc/Reference/13> [Consulta 2016]
- [23] Arduino [En línea] <www.arduino.cc/Redeference/89> [Consulta 2016]
- [24] Arduino [En línea] <www.arduino.cc/Reference/90> [Consulta 2016]
- [25] IEC 61850. International Electrotechnical Commision. Communications networks and systems in substations.IEC 61850-8-1:2004 p.p 19
- [26] IEC 61850. International Electrotechnical Commision. Communications networks and systems in substations.IEC 61850-8-1:2004 p.p 19
- [27] IEC 61850. International Electrotechnical Commision. Communications networks and systems in substations.IEC 61850-8-1:2004 p.p 19
- [28] IEEE Standard for Floating-Point Arithmetic. USA. 2008
- [29] Pila de protocolos sobre el modelo de capas OSI [En línea] <<http://www.xelasenergy.com>> [Consulta 2016]
- [30] IEC 61850. International Electrotechnical Commision. Communications networks and systems in substations.IEC 61850-8-1:2004 p.p 19
- [31] IEC 61850. International Electrotechnical Commision. Communications networks and systems in substations.IEC 61850-8-1:2004 p.p 20
- [32] IEC 61850. International Electrotechnical Commision. Communications networks and systems in substations.IEC 61850-8-1:2004 p.p 25

BIBLIOGRAFÍAS

- J.Zamora, M.Ricucci, D.Flores, E.Cárdenas, V.Depablos, F.Aldana, Et al. El Bote de Carga en Venezuela con Transporte Multitrayectoria de Comandos deTeleprotecciones por sistemas de Telecomunicaciones. II Congreso Venezolano de Redes y Energía Eléctrica. Caracas. Junio 2009
- P. Klaus, W.Wolfgang. Modeling Interoperable Protection and Control Devices for Substation Automation According to IEC 61850. Switzerland. 2006.
- H. Daqing, D.Dolezilek. IEC 61850 – What It Can and Cannot Offer to Traditional Protection Schemes. Schweitzer Engineering Laboratories, Inc. Switzerland. 2008.
- IEC 61850. International Electrotechnical Commision. Communications networks and systems in substations.IEC 61850-8-1. Geneva. 2002
- IEC 61850. International Electrotechnical Commision. Communications networks and systems in substations.IEC 61850-7-2. Geneva. Agosto 2002
- M. Serra, F. Castro. Using IEC 61850 for Teleprotection, 19th International Conference on Electricity Distribution, Vienna, Mayo 2007.
- C.Fernandes, S.Borkar. J. Gohil. Testing of Goose Protocol of IEC61850 Standard in Protection IED. International Journal of Computer Applications

(0975 – 8887). Volume 93 – No 16, May 2014.

- T.Perez, D. Montoya, P. Tremante y R. Diaz, Simulación de Sistemas de Potencia con ATP. Memorias del II Seminario de Modelos y Modelado : Conceptos, técnicas y aplicaciones. Caracas. 2014
- S. Blair, F. Coffele, C. Booth, G. Burt. An Open Platform for Rapid-Prototyping Protection
- and Control Schemes with IEC 61850. IEEE Transactions on power delivery. UK, 2011
- N. Patel. iEC 61850 Horizontal Goose Communication and Overview. LAP Lambert Acad. Publ., 2011
- Advances in Computing, Communication and Control: International Conference, ICAC3 2011 India. January 2011
- M. Margolis. Arduino Cookbook. O'Reilly Media, Inc. Marzo 2011
- E. Padilla. Substation Automation Systems: Design and Implementation. Venezuela. 2015
- P. Verma. Wireshark Network Security. Julio. 2015

ANEXOS