



Universidad
Carlos III de Madrid

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

PROYECTO FIN DE CARRERA

INGENIERÍA DE TELECOMUNICACIÓN

**MEJORA Y VALIDACIÓN DE UN
SIMULADOR DE REDES PRIME
BASADA EN ESCENARIOS
REALES**

AUTOR: ÓSCAR PALOMO MIÑAMBRES
TUTOR: DR. GREGORIO LÓPEZ LÓPEZ
CO-DIRECTOR: DR. JAVIER MATANZA DOMINGO

LEGANÉS, DICIEMBRE DE 2016

TÍTULO: MEJORA Y VALIDACIÓN DE UN SIMULADOR DE REDES
PRIME BASADA EN ESCENARIOS REALES
AUTOR: ÓSCAR PALOMO MIÑAMBRES
TUTOR: DR. GREGORIO LÓPEZ LÓPEZ
CO-DIRECTOR: DR. JAVIER MATANZA DOMINGO

EL TRIBUNAL

PRESIDENTE: DR. JOSÉ IGNACIO MORENO NOVELLA

VOCAL: DRA. HORTENSIA AMARÍS DUARTE

SECRETARIO: DRA. ALICIA RODRÍGUEZ CARRIÓN

REALIZADO EL ACTO DE DEFENSA Y LECTURA DEL PROYECTO FIN DE CARRERA EL DÍA 14 DE DICIEMBRE DE 2016 EN LEGANÉS, EN LA ESCUELA POLITÉCNICA SUPERIOR DE LA UNIVERSIDAD CARLOS III DE MADRID, ACUERDA OTORGARLE LA CALIFICACIÓN DE

VOCAL

SECRETARIO

PRESIDENTE

A mi familia

Agradecimientos

He de expresar mi profundo agradecimiento a todas aquellas personas que me han dado la oportunidad de desarrollarme, tanto intelectual como personalmente, en la realización de esta meta, de este sueño que es tan importante para mí. Agradecer todas sus ayudas, sus palabras motivadoras, sus conocimientos, sus consejos y su dedicación. Su influencia es de una importancia capital para poder llegar al punto en el que me encuentro.

Muestro mis más sinceros agradecimientos a mi tutor de proyecto, Gregorio, quien con su conocimiento y su guía fue una pieza clave para conseguir esta meta. Su apoyo, dedicación y ánimos fueron imprescindibles para cada etapa de desarrollo del trabajo.

En la misma medida me ha llenado de satisfacción y alivio poder contar con los conocimientos y apoyo de Miguel Seijo y Javier Matanza durante la realización de todo el proyecto, su dedicación y tiempo me han servido para solventar algunos de los problemas que han ido apareciendo en la realización del mismo.

A mi madre, la verdadera responsable de intentar ser Ingeniero, por su apoyo incondicional, por su cariño, por su comprensión, por darme fuerzas en momentos difíciles que a pesar de no poder estar hoy aquí presente sé que se alegraría de verme conseguir mis metas.

Por último, quiero agradecer a la base de todo, a mi familia, en especial a mi mujer y mis hijos, que son la base de mi existencia y por los que día a día intento superarme para poder darles lo mejor, han sido y siguen siendo el motor de arranque y mi

constante motivación, les doy gracias por estar a mi lado, por su paciencia, y sobre todo por su amor.

Gracias.

*“Nuestra recompensa se encuentra en el esfuerzo y no en el resultado.
Un esfuerzo total es una victoria completa”
Mahatma Gandhi*

Resumen

PRIME es un estándar de comunicaciones PLC de Banda Estrecha ampliamente utilizado en la última milla de los despliegues de infraestructuras avanzadas de medición en España (Iberdrola, Unión Fenosa) y con proyección internacional. Las herramientas de simulación son especialmente importantes para la planificación y evaluación de este tipo de redes, permitiendo minimizar riesgos, tiempo y costes. Sin embargo, para conseguir este objetivo sus resultados deben ajustarse a resultados reales.

El objetivo de este Proyecto Fin de Carrera es precisamente validar si los resultados obtenidos en el simulador SimPrime se ajustan a los resultados medidos en campo mediante el análisis de trazas reales. La métrica utilizada para conseguir tal objetivo es el TTRi, que es el tiempo que tarda el concentrador de datos en obtener medidas de consumo de un contador después de haber realizado una solicitud a tal efecto.

El presente Proyecto Fin de Carrera ha sido desarrollado dentro del ámbito del proyecto de investigación nacional OSIRIS (Optimización de la Supervisión Inteligente de la Red de Distribución), financiado por el Ministerio de Economía y Competitividad y liderado por Unión Fenosa Distribución (tercera distribuidora eléctrica a nivel nacional).

Palabras clave

Comunicaciones PLC de Banda Estrecha; Infraestructuras Avanzadas de Medición; PRIME; Python; Red Inteligente; Simulador de Redes de Comunicaciones

Abstract

Narrow Band PLC communications are gaining importance as last mile technologies in Advanced Metering Infrastructures. PRIME is a Narrow Band PLC communications standard widely used in Spain (e.g., by Unión Fenosa or Iberdrola), and with international projection. Simulation tools are particularly important in this area, as they allow planning, evaluating and making decisions about PRIME networks, minimizing risks, time and costs. However, the value that this kind of tools actually bring depends on whether they fit reality or not.

The aim of this thesis is to validate if the results obtained in the network simulator SimPrime fit the results measured in the field by the analysis of actual traffic traces. The metric used for this purpose is the TTRi, which is the time it takes the data concentrator to obtain measures of consumption from a meter after having made a request to that effect.

This Thesis has been developed within the scope of the national research project OSIRIS (Optimization of the Distribution Network Intelligent Monitoring), funded by the Spanish Ministry of Economy and Competiveness and led by Unión Fenosa Distribución (third company in the Spanish electricity distribution market).

Keywords

Advanced Metering Infrastructure (AMI); Communications Network Simulator; PowerLine Intelligent Metering Evolution (PRIME); Python; Smart Grids

Índice general

Agradecimientos	5
Resumen	9
Abstract	11
Índice de figuras	15
Índice de tablas.....	18
Acrónimos	19
1. Introducción.....	21
1.1 Motivación	22
1.2 Objetivos	24
1.3 Estructura de la memoria.....	25
2. Estado del arte	27
2.1 Arquitectura de comunicación para AMI	28
2.2 Tecnologías PLC de Banda Estrecha	31
2.2.1 Visión global	31
2.2.2 PRIME.....	33
2.2.3 DLMS/COSEM	35
2.3 Formato de trazas	37
2.3.1 ZIV	45
2.3.2 CIRCUTOR.....	46
2.4 Simuladores de redes de comunicaciones	48
2.4.1 Riverbed Modeler.....	48
2.4.2 NS3.....	48
2.4.3 OMNeT++	49

2.5	Simulador SimPRIME.....	50
2.5.1	Visión Global	50
2.5.2	SimPrime	50
2.6	Tecnologías utilizadas para el desarrollo	53
2.6.1	Python.....	53
2.6.2	Django	54
2.6.3	Celery	54
2.6.4	Highcharts	55
2.6.5	Google Chart	55
2.6.1	RabbitMQ.....	56
2.6.2	PostgreSQL	56
2.6.3	Docker	56
3.	Diseño	58
3.1	Metodología.....	59
3.2	Requisitos	60
3.3	Planificación.....	63
3.4	Diseño.....	64
4.	Desarrollo	69
4.1	Análisis de Trazas	70
4.1.1	Módulo standalone para el cálculo del <i>TTRi</i>	70
4.1.2	Integración del módulo standalone para el cálculo del <i>TTRi</i> en la herramienta PRIME Analytics.....	72
4.2	Mejora de la aplicación web de SimPRIME	74
5.	Validación	77
5.1	Validación del módulo <i>standalone</i> para el cálculo del <i>TTRi</i>	78
5.2	Validación del simulador frente a un escenario real.....	102
6.	Conclusiones y Trabajos Futuros	105
6.1	Conclusiones	106
6.2	Trabajos futuros.....	108
7.	Referencias.....	109
8.	Presupuesto	113
I.1	Planificación.....	114
I.2	Recursos y Costes.....	117
I.2.1	Recursos tangibles e intangibles.....	117
I.2.2	Recursos humanos	117
I.2.3	Presupuesto total.....	118

Índice de figuras

Figura 1 Principales componentes de un sistema AMI	28
Figura 2 Visión Global de la arquitectura de red del proyecto PRICE-GEN.....	30
Figura 3: Mapa del despliegue de tecnologías NB-PLC en Europa (18)	31
Figura 4 Torre de Protocolos Red PRIME	34
Figura 5 Relación Cliente/servidor	35
Figura 6. Comunicación Cliente/Servidor a nivel de aplicación.....	36
Figura 7 Fragmentación de mensajes DLMS/COSEM que superan la MTU establecida.....	37
Figura 8 Estructura de datos de una traza PRIME	38
Figura 9 Estructura paquetes de datos PRIME.....	38
Figura 10 Cabecera correspondiente a la capa MAC en tráfico PRIME.....	40
Figura 11 Identificadores de entidad en redes PRIME.....	41
Figura 12. Estructura Lógica de la Red PRIME.....	41
Figura 13 Estructura Lógica red PRIME.....	41
Figura 14 Intercambio de mensajes a nivel de aplicación.....	42
Figura 15 Panel de Telemedida.....	44
Figura 16 Estructura del laboratorio de integración de redes (LINTER)	44
Figura 17 Ejemplo de traza PRIME asociada a ZIV	45
Figura 18 Parámetros Capa Física ZIV	45
Figura 19 Parámetros Capa MAC ZIV	45
Figura 20 Parámetros Capa de aplicación ZIV	46
Figura 21 Tiempo de registro de la recepción/envío de trazas ZIV	46
Figura 22 Ejemplo de traza PRIME asociada a CIRCUTOR.....	46
Figura 23 Parámetros Capa Física CIRCUTOR.....	47
Figura 24 Parámetros Capa MAC CIRCUTOR.....	47

Figura 25	Parámetros Capa de aplicación CIRCUTOR.....	47
Figura 26	Tiempo de registro de la recepción/envío de trazas CIRCUTOR.....	47
Figura 27	Esquema arquitectura de simulación utilizada.....	51
Figura 28	Resumen gráfico de entradas y salidas del simulador (5).....	52
Figura 29	Aplicación Web, simPRIME y módulo para procesar Shapefiles (6).....	52
Figura 30	Gestión de proyectos usando metodología PMI	59
Figura 31:	Entradas y salidas del sistema.....	62
Figura 32	Extracción Datos ZIV en formato estándar	66
Figura 33	Extracción Datos CIRCUTOR en formato estándar	66
Figura 34	Ejemplo de trazas PRIME correspondiente a un bloque procedente de la capa de aplicación	67
Figura 35	Explicación del proceso de extracción.....	70
Figura 36	Salida Shell del módulo standalone	71
Figura 37	Obtención de estadísticos relativos al TTRi	71
Figura 38	Diagrama de caja y bigote obtenido mediante el módulo standalone.....	71
Figura 39	Diagrama de flujo PRIME ANALYTICS.....	72
Figura 40	Salida Herramienta PRIME ANALYTICS para la obtención del TTRi.....	73
Figura 41	Formato Datos TTRi exportados de PRIME ANALYTICS	73
Figura 42	Formato de Salida TTRi Simulador.....	74
Figura 43	Obtención del TTRi global en el Simulador	75
Figura 44	Obtención del TTRi por nodo en el simulador	75
Figura 45.	Flujo de trabajo para extender la funcionalidad del Simulador	76
Figura 46	Ejecución Módulo Standalone para el análisis de trazas reales	79
Figura 47	Elección del archivo de trazas reales en el módulo standalone desarrollado.....	79
Figura 48	Salida Módulo standalone para ZIV	80
Figura 49	Intercambio de mensajes entre Concentrador y contador (ZIV)	81
Figura 50	Formato trazas ZIV	81
Figura 51	Solicitud informe S02 ZIV	82
Figura 52	Transmisión de un bloque completo a nivel de aplicación ZIV	83
Figura 53	Fin transmisión Informe S02 ZIV	83
Figura 54	Solicitud S02 al contador 09459	84
Figura 55	Evidencia recepción último bloque informe S02 del contador 09459	84
Figura 56	Solicitud S02 al contador 09970	84
Figura 57	Evidencia recepción último bloque informe S02 del contador 09970	85
Figura 58	Solicitud S02 al contador 09567	85
Figura 59	Evidencia recepción último bloque informe S02 del contador 09567	85
Figura 60	Solicitud S02 al contador 08717	86
Figura 61	Evidencia recepción último bloque informe S02 del contador 08717	86
Figura 62.	Ejemplo de tráfico anómalo en ZIV.....	87
Figura 63	Salida Módulo standalone para CIRCUTOR.....	90
Figura 64.	TTRi obtenido en el análisis de trazas reales para CIRCUTOR.....	91
Figura 65	Intercambio de mensajes entre CNTR y contador (CIRCUTOR)	92
Figura 66	Intercambio de mensajes completo para una solicitud de informe S02	93
Figura 67	Solicitud informe S02 CIRCUTOR	93
Figura 68	Transmisión de un bloque completo a nivel de aplicación CIRCUTOR.....	93
Figura 69	Fin transmisión Informe S02 CIRCUTOR	94
Figura 70	Solicitud S02 al contador 109196.....	94
Figura 71	Evidencia recepción último bloque informe S02 del contador 109196	94
Figura 72	Validación de resultados CIRCUTOR para el contador con NID 109196	95
Figura 73	Solicitud S02 al contador 59010.....	95
Figura 74	Evidencia recepción último bloque informe S02 del contador 59010	95

Figura 75 Validación de resultados CIRCUTOR para el contador con NID 59010	96
Figura 76 Solicitud S02 al contador 68222	96
Figura 77 Evidencia recepción último bloque informe S02 del contador 68222	96
Figura 78 Validación de resultados CIRCUTOR para el contador con NID 68222	97
Figura 79 Solicitud S02 al contador 178900	97
Figura 80 Evidencia recepción último bloque informe S02 del contador 178900	97
Figura 81 Validación de resultados CIRCUTOR para el contador con NID 178900	98
Figura 82 Intercambio Anómalo para una solicitud de informe de consumo S02	99
Figura 83 Resumen Intercambio completo de solicitud de informe S02	99
Figura 84 Salida del módulo standalone para CIRCUTOR con intercambio anómalo.....	100
Figura 85 Escenario real utilizado para la validación de los resultados del simulador	102
Figura 86 Resumen de tareas	115
Figura 87 Diagrama de Gantt	116

Índice de tablas

Tabla 1 Campos de cabecera asociados a paquetes PRIME	39
Tabla 2 TTRi ZIV	86
Tabla 3 Formato de traza ZIV en bytes.....	88
Tabla 4 Cálculo tiempo transmisión por tipo de traza (ZIV)	89
Tabla 5 TTRi CIRCUTOR.....	98
Tabla 6 Formato de traza CIRCUTOR en bytes	100
Tabla 7 Cálculo tiempo transmisión por tipo de traza (CIRCUTOR).....	101
Tabla 8 Parámetros PRIME.....	101
Tabla 9 Estadísticos Relevantes Obtenidos en el análisis de los resultados de TTRi	103
Tabla 10 Tabla comparativa estadísticos TTRi con WS=1	103
Tabla 11 Tabla comparativa estadísticos TTRi con WS=4.....	103
Tabla 12 Coste de los recursos tangibles e intangibles.	117
Tabla 13 Coste de recursos humanos.	118

Acrónimos

AA	<i>Application Association</i>
AE	<i>Application Entity</i>
AMI	<i>Advanced Metering Infrastructure</i>
AMQP	<i>Advanced Message Queuing Protocol</i>
ARQ	<i>Automatic Repeat Request</i>
A-XDR	<i>eXternal Data Representation</i>
BBDD	<i>Bases de Datos</i>
BER	<i>Bit Error Rate</i>
CFP	<i>Contention Free Period</i>
CNTR	<i>Concentrador de datos</i>
COSEM	<i>COmpanion Specification for Energy Metering</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CT	<i>Centro de transformación</i>
DLMS	<i>Device Language Message Specification</i>
DPSK	<i>Differential Phase Shift Keying</i>
FEC	<i>Forward Error Correction</i>
GML	<i>Geography Markup Language</i>
GNU	<i>General Public License</i>
IEC	<i>Comisión Electrotécnica Internacional</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
ITU-T	<i>Telecommunication Standardization Sector of the International Telecommunications Union</i>

LLC	<i>Logical Link Control</i>
MAC	<i>Medium Access Control</i>
MDMS	<i>Metering Data Management System</i>
MTU	<i>Maximum Transmission Unit</i>
MTV	<i>Model-Template-View</i>
MVC	<i>Modelo-Vista-Controlador</i>
M2M	<i>Machine-to-Machine</i>
NB-PLC	<i>Narrowband Power Line Communications</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
OSGP	<i>Open Smart Grid Protocol</i>
OSIRIS	Optimización de la Supervisión Inteligente de la Red de Distribución
PER	<i>Packet Error Rate</i>
PFC	Proyecto Fin de Carrera
PHY	<i>PHYSical</i>
PLC	<i>Power Line Communications</i>
PMBOK	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
PRIME	<i>PowerLine Intelligent Metering Evolution</i>
SCP	<i>Shared Contention Period</i>
SGBD	Sistema de gestión de bases de datos
QoS	<i>Quality of Service</i>
SIG	Sistemas de Información Geográfica
SNR	<i>Signal-to-Noise-Ratio</i>
TIC	Tecnologías de la Información y la Comunicación
VMDS	<i>Version Manager Data Store</i>
WS	<i>Windows Size</i>

Capítulo 1

Introducción

El primer capítulo de este Proyecto Fin de Carrera está dedicado a proporcionar una visión global del mismo. Comenzaremos por presentar los diferentes aspectos que han motivado su desarrollo y los objetivos que se pretenden alcanzar en él. Finalmente se describe como han sido estructurados los diferentes contenidos del presente documento.

1.1 Motivación

Las denominadas Redes Eléctricas Inteligentes (en inglés, *Smart Grids*) son redes capaces de integrar de un modo automatizado y eficiente el comportamiento y acciones de todos los agentes conectados a ellas con el fin de proporcionar un suministro eléctrico económicamente eficiente, sostenible, con bajas pérdidas y elevados niveles de calidad y seguridad (1).

Las redes de comunicaciones *Machine-to-Machine* (M2M) son de especial importancia para las *Smart Grids* ya que engloban la conexión y la comunicación entre todas las máquinas y dispositivos de un sistema sin intervención humana, haciendo posible controlar, automatizar, mejorar los procesos y expandir los negocios con seguridad y costes reducidos. Las comunicaciones *PowerLine Communications* (PLC) son especialmente relevantes en este tipo de entornos porque se ajustan a los requisitos que deben cumplir este tipo de redes, tales como baja latencia, alta disponibilidad, bajos coste de despliegue y operación, etc. (2)

Así, las tecnologías PLC de banda estrecha (del inglés *Narrowband PLC* (NB-PLC)) están siendo ampliamente desplegadas en infraestructuras de medición avanzada (AMI) debido a los bajos coste de integración que conlleva utilizar las instalaciones de baja tensión convencionales para conectar los contadores inteligentes (del inglés *Smart Meters*) con los concentradores (CNTR) de datos situados en los centros de transformación (CTs) (2).

Entre las distintas opciones NB-PLC disponibles en el mercado destaca la tecnología *Powerline Intelligent Metering Evolution* (PRIME), por ser hoy en día un estándar PLC maduro, consolidado y abierto con proyección internacional. Cuenta con un gran número de equipos y sistemas certificados de diferentes fabricantes que ya ha permitido el despliegue de más de 12 millones aproximadamente de *Smart Meters* (3).

Si bien es cierto que las comunicaciones basadas en la propia infraestructura eléctrica presentan multitud de ventajas para las distribuidoras eléctricas, también es cierto que se trata de un medio de comunicaciones hostil, por lo que es necesario desarrollar herramientas de monitorización y simulación que permitan planificar, depurar, evaluar y diagnosticar redes de comunicaciones PRIME minimizando riesgos, tiempo y costes.

SimPRIME es un simulador de redes PRIME, basado en Matlab y OMNeT++, que persigue dichos objetivos. El núcleo del simulador fue desarrollado como parte de la tesis doctoral de Javier Matanza Domingo (4) despertando gran interés por parte de distribuidoras eléctricas como Unión Fenosa o Iberdrola.

Sin embargo, la larga curva de aprendizaje que requiere la utilización de este tipo de simuladores junto con la imposibilidad de poder permitir simular escenarios reales supuso un obstáculo para ser usado en entornos de producción. Estas dificultades de manejo y de adaptación a un entorno real motivó el desarrollo de una aplicación Web, que permitió simplificar la interacción con el simulador, y de un módulo software que permite procesar información geográfica de redes de distribución eléctrica desplegadas en campo permitiendo la simulación de redes PRIME en operación. Ambos desarrollos fueron

llevados a cabo en sendos Proyectos Fin de Carrera previos con el objetivo de intentar aumentar el impacto de SimPRIME en la industria. (5), (6).

Sin embargo, para que los resultados del simulador aporten realmente valor es necesario comprobar si se ajustan a la realidad o no.

Cabe destacar que este trabajo ha sido desarrollado dentro del ámbito del proyecto de investigación nacional OSIRIS (7) (Optimización de la Supervisión Inteligente de la Red de Distribución), financiado por el Ministerio de Economía y Competitividad y liderado por Unión Fenosa Distribución (tercera distribuidora eléctrica a nivel nacional).

1.2 Objetivos

El objetivo principal de este PFC es validar los resultados obtenidos por el simulador SimPRIME con resultados obtenidos en redes reales mediante el análisis de trazas de tráfico obtenidas en las mismas. La validación se llevará a cabo utilizando el valor TTR_i como métrica. El TTR_i es el tiempo que tarda el concentrador (CNTR) en obtener medidas de consumo de un contador después de haber realizado una solicitud a tal efecto.

Para conseguir dicho objetivo global, se han establecido una serie de sub-objetivos o hitos descritos a continuación:

- Estudio de la torre de protocolos utilizada en la última milla de AMI (PRIME+DLMS/COSEM).
- Estudio de los formatos de las trazas de tráfico utilizados por distintos fabricantes.
- Desarrollo de un módulo *standalone* para el cálculo del TTR_i que permita calcular el TTR_i a partir del análisis de trazas reales.
- Análisis de la arquitectura y tecnologías utilizadas en la herramienta *PRIME Analytics* (8), (9) para integrar el módulo *standalone* mencionado en el punto anterior y poder calcular y representar los valores TTR_i para cada contador.
- Análisis de la arquitectura y tecnologías utilizadas en la aplicación Web del simulador SimPRIME para extender las funcionalidades del mismo pudiendo obtener y representar los datos asociados al TTR_i para cada contador.
- Validación de los resultados obtenidos.
- Documentación del desarrollo realizado.

1.3 Estructura de la memoria

La organización de los contenidos del presente Proyecto Fin de Carrera se detalla a continuación:

- **Capítulo 1: Introducción**

El primer capítulo de este Proyecto Fin de Carrera está dedicado a proporcionar una visión global del mismo. En este capítulo se presentan los diferentes aspectos que han motivado el desarrollo de este PFC, los objetivos que se pretenden alcanzar en él y la estructura del presente documento.

- **Capítulo 2: Estado del Arte**

El propósito de este capítulo es obtener una visión global de todas aquellas tecnologías y herramientas relacionadas con el presente Proyecto Fin de Carrera. Así, se revisan las tecnologías NB-PLC para presentar más detalladamente las redes PRIME y así comprender la importancia de la existencia de simuladores de redes de comunicaciones en este ámbito. Seguidamente, se presenta el funcionamiento del simulador simPRIME. A continuación se presenta un análisis de todas las tecnologías utilizadas para llevar a cabo el desarrollo *software* propuesto.

- **Capítulo 3: Diseño**

En este capítulo se describe el proceso de diseño y la metodología de trabajo seguida en el presente proyecto. Para ello, se presentan en primer lugar los requisitos funcionales proporcionados y, a continuación, las principales decisiones técnicas y de diseño que se han tomado de cara al desarrollo de los módulos *software* descritos en el capítulo siguiente.

- **Capítulo 4: Desarrollo**

El objetivo de este capítulo es presentar los aspectos más técnicos del desarrollo realizado en el proyecto, tomando como referencia el diseño presentado en el capítulo anterior. Así, en este capítulo se detalla la implementación de cada uno de los módulos en el orden en que fueron desarrollados para permitir comprender el proceso de forma clara y sencilla.

- **Capítulo 5: Validación**

Este capítulo detalla las diferentes validaciones realizadas para comprobar el correcto funcionamiento de los módulos *software* desarrollados, así como una comparativa de los valores obtenidos por el módulo de análisis de trazas y por el simulador, con el fin de determinar la validez de este último. En primer lugar, se realiza la validación del módulo *software* desarrollado para calcular el TTR_i mediante el procesamiento de trazas de tráfico de redes PRIME reales. Posteriormente se comparan los valores obtenidos mediante simulaciones con los valores medidos en un caso real, comprobándose que encajan, lo

que supone la principal conclusión de este trabajo. Por último, se analiza cómo varía la salida del simulador al cambiar ciertos parámetros de comunicaciones.

- **Capítulo 6 Conclusiones y Líneas de Trabajo Futuras**

En este capítulo final se exponen las principales conclusiones obtenidas tras la ejecución del presente Proyecto Fin de Carrera, así como posibles trabajos futuros.

- **Anexo I: Presupuesto**

Este anexo presenta la planificación y fases de desarrollo del Proyecto Fin de Carrera así como el presupuesto del mismo, desglosándolo en los correspondientes costes de material y costes de recursos humanos.

Capítulo 2

Estado del arte

El propósito de este capítulo es obtener una visión global de todas aquellas tecnologías y herramientas relacionadas con el presente Proyecto Fin de Carrera. Así, se revisa la arquitectura de comunicaciones para las infraestructuras de medición avanzada (AMI) donde las tecnologías NB-PLC tienen especial importancia. Posteriormente se presentan más detalladamente las redes PRIME para comprender la importancia de la existencia de simuladores de redes de comunicaciones en este ámbito. Seguidamente, se presenta el funcionamiento del simulador simPRIME. Por último, se da una visión global de las principales tecnologías utilizadas en este trabajo.

2.1 Arquitectura de comunicación para AMI

Un sistema de medición inteligente es un sistema de medición que registra el consumo de los clientes (y posiblemente otros parámetros) horariamente o con mayor frecuencia entregando de manera diaria o en menor tiempo las mediciones hacia una central de recolección a través de una red de comunicación (10). En definitiva, se considera un sistema de medición inteligente cuando se dispone de un contador que es capaz de transmitir la información de sus mediciones a la empresa de servicios. Éste normalmente mide, registra y almacena el consumo y otros parámetros que son necesarios por las empresas proveedoras para sus aplicativos, con la finalidad de realizar análisis e implementar nuevas prestaciones que optimicen sus procesos y la gestión de sus activos; si el modelo de comunicación es bidireccional se conoce como un sistema de Infraestructura de Medición Avanzada (AMI).

Las AMI tienen como objetivo la automatización del ciclo comercial de lectura, suspensión y reconexión del suministro de energía eléctrica e inclusive de otros servicios (como gas y agua) a través de un sistema de comunicación bidireccional que asocia diversos elementos *hardware* (contadores inteligentes, integradores, concentradores de datos, entre otros) con el *software* para el manejo de los datos de medida, así como con los sistemas de gestión de usuarios.

En la Figura 1 se identifican claramente tres componentes fundamentales en un sistema de medición avanzada.

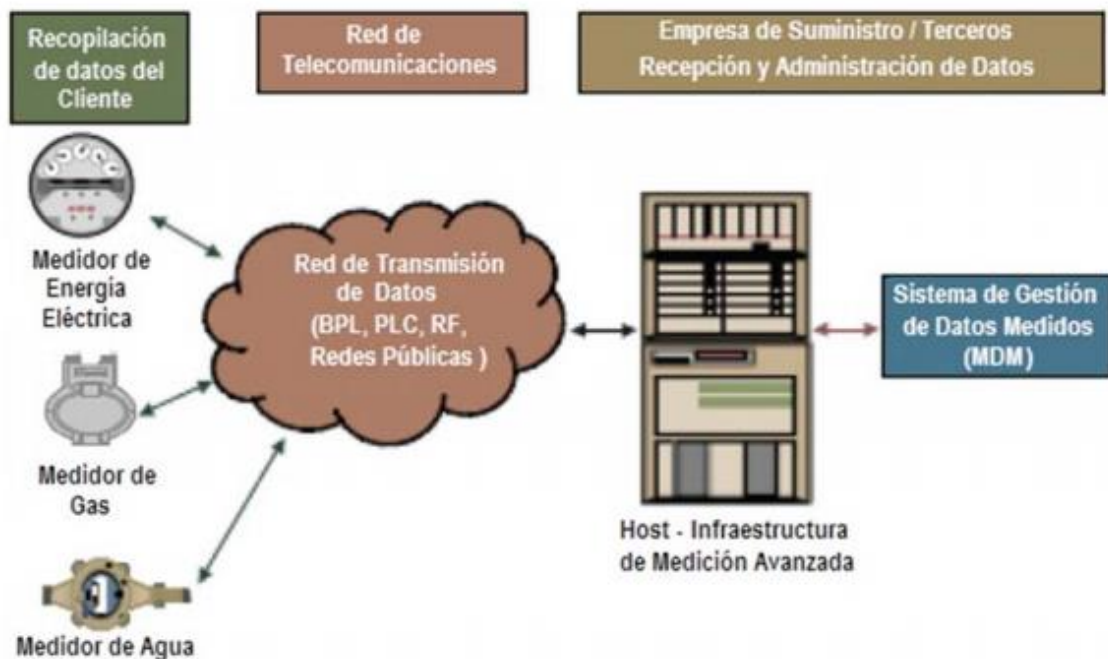


Figura 1 Principales componentes de un sistema AMI

Uno de los principales componentes del sistema AMI es el equipo de medición inteligente (*Smart Meter*), ya que es éste el elemento que actúa como interfaz entre el cliente y los sistemas de gestión de información del operador de red. Además de garantizar la robustez, precisión y fiabilidad de los medidores tradicionales, el medidor inteligente debe incorporar la capacidad de comunicación bidireccional fiable hacia otros equipos y sistemas de gestión, por lo que deja de ser un mero sensor y pasa a ser parte del núcleo de la infraestructura.

Un segundo componente que constituye el medio por el cual los medidores transfieren los datos a las bases de datos remotas, es decir, las redes de telecomunicaciones. Para la comunicación es posible utilizar diversas tecnologías: RF, PLC, GPRS, banda ancha (Ethernet, WiFi, WiMAX, ADSL), Satelital y Fibra Óptica, e inclusive combinaciones de ellas

Por último, un sistema de *hardware* y *software* que controla la comunicación, recibe los datos enviados de los medidores y los almacena. Esta información puede ser administrada por un sistema de gestión de Datos de Medición (MDMS – *Metering Data Management System*).

Uno de los problemas tecnológicos fundamentales de AMI es el de comunicación. Cada contador es capaz de comunicar de forma fiable y segura la información recogida a un receptor central, pero, teniendo en cuenta los diferentes ambientes y lugares en los que se encuentran los contadores, se evidencian diferentes problemas.

No existe una solución única para determinar el medio utilizado de forma óptima, pues existe una dependencia directa con las características particulares de los sitios donde se presta el servicio.

La Figura 2 muestra una visión global de la arquitectura de comunicaciones para AMI típicamente desplegada en España, diseñada en el proyecto PRICE-GEN (11) y utilizada en el proyecto OSIRIS. Dicha arquitectura está diseñada para ajustarse a los requisitos de calidad de servicio (del inglés *Quality of Service* (QoS)), interoperabilidad, escalabilidad y seguridad que exigen las denominadas redes inteligentes (12) así como a las diferentes tipologías de red presentes en este tipo de despliegues (urbana, semi-urbana y rural).

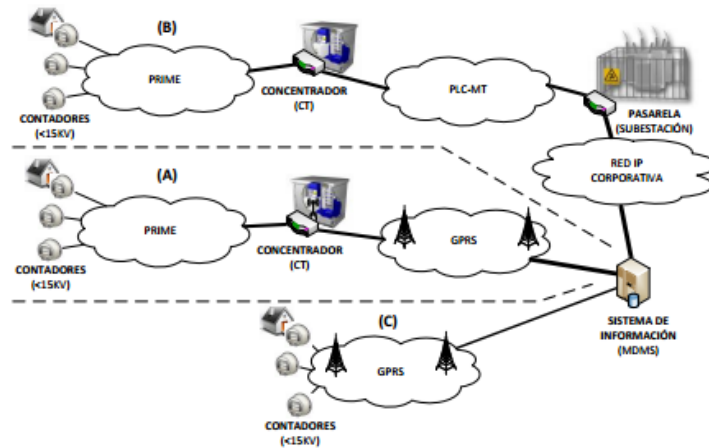


Figura 2 Visión Global de la arquitectura de red del proyecto PRICE-GEN

Se trata de una arquitectura de red heterogénea en tanto en cuanto comprende a su vez tres arquitecturas de red (identificadas como (A), (B) y (C) en la Figura 2), cada una de las cuales está compuesta por diferentes segmentos de red, en los que se emplean diferentes tecnologías de comunicación.

La tecnología PLC utiliza las líneas de potencia como medio para la transmisión de datos a lo largo de los sistemas de transmisión y distribución eléctrica. Desde el punto de vista económico, son proyectos rentables para las empresas de electricidad, debido a que no existen inversiones adicionales asociadas a su implementación, comparado con otros tipos de tecnologías (p.ej., infraestructuras de red RF). Al usar PLC se aprovecha la infraestructura eléctrica existente otorgando mayor agilidad y escalabilidad al despliegue.

En la siguiente sección damos una visión global de las tecnologías PLC.

2.2 Tecnologías PLC de Banda Estrecha

2.2.1 Visión global

Múltiples estudios (13) (14) (15) (16) y proyectos piloto desarrollados por grandes distribuidoras eléctricas (17) apuntan a que NB-PLC es la tecnología más apropiada especialmente para la última milla de despliegues AMI, como interfaz entre los concentradores de datos y los contadores inteligentes.

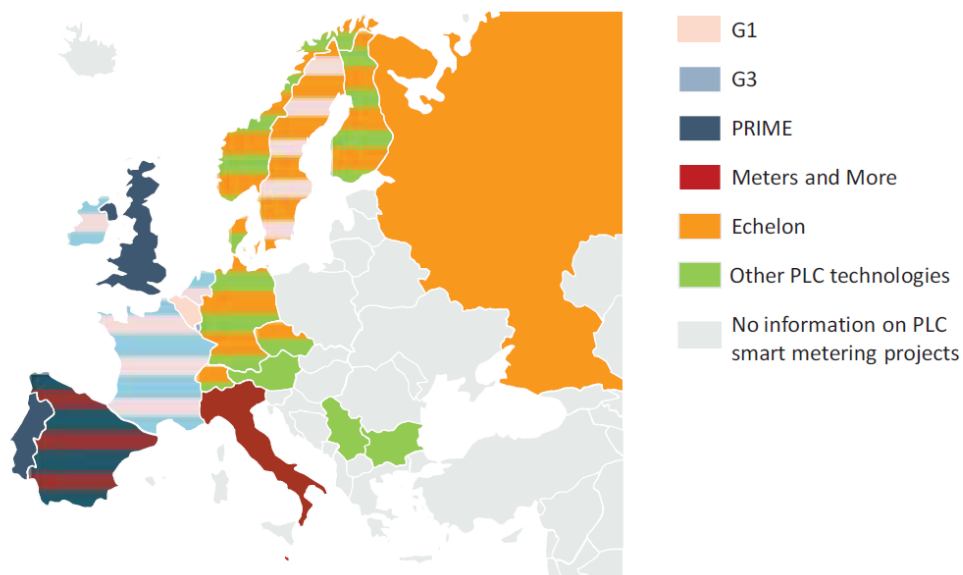


Figura 3: Mapa del despliegue de tecnologías NB-PLC en Europa (18)

Existen de hecho diferentes estándares NB-PLC en el mercado, con distintas características técnicas y niveles de penetración, como ilustra la Figura 3:

- *Meters and More* es una tecnología cuya especificación lidera el grupo Enel, aunque sus capas inferiores también están siendo estandarizadas por IEC/CENELEC (CLC TS 50568-4). Presenta sus mayores niveles de penetración en mercados donde opera el grupo ENEL, tales como Italia (el 100% de los contadores italianos usan esta tecnología actualmente) o España (aproximadamente la mitad del parque de contadores inteligentes españoles utilizarán esta tecnología en 2018). Funciona en la banda CENELEC-A. Utiliza una única portadora, destacando por su robustez, aunque alcanza tasas de transmisión de datos bajas (9,6 kbps).
- *Open Smart Grid Protocol (OSGP)* es una tecnología promovida por el fabricante norteamericano Echelon, aunque sus capas inferiores están también estandarizadas por el IEC (IEC 14908.1). Presenta sus mayores tasas de despliegue en los países nórdicos y Rusia. Funciona en la banda CENELEC-A. Se trata de una tecnología monoportadora que alcanza tasas de transmisión de datos de hasta 3,6 kbps.

- *CXI* es una tecnología promovida por Siemens y cuyas capas inferiores están siendo estandarizadas también por IEC/CENELEC (CLC TS 50590). Actualmente se está desplegando en Austria. Funciona en la banda CENELEC-A. Utiliza una modulación multiportadora adaptativa que le permite alcanzar tasas de hasta 64 kbps.
- *G3* es una tecnología cuya especificación lidera la distribuidora EDF y el fabricante de chipsets Maxim, aunque sus capas PHY y de enlace han sido publicadas como estándar por la ITU-T recientemente (ITU-T G9903). Presenta sus mayores niveles de penetración en mercados donde opera EDF, como Francia. Inicialmente definida para que funcionase en la banda CENELEC-A, ha sido extendida recientemente para poder ser utilizada en el mercado americano (FCC) y asiático (ARIB). Se trata de una tecnología multiportadora que alcanza tasas de hasta 34 kbps y que ha destacado desde sus inicios por ser especialmente robusta.
- Como ya se ha comentado, *PowerLine Intelligent Metering Evolution* (PRIME) es una tecnología promovida por la PRIME Alliance, liderada por distribuidoras eléctricas españolas como Iberdrola y Unión Fenosa Distribución y por fabricantes de chipsets como Texas Instruments. Sus capas PHY y de enlace también han sido publicadas como estándar por la ITU-T (ITU-T G.9904) (19)). Al igual que G3, fue inicialmente definida para que funcionase en la banda CENELEC-A y ha sido extendida recientemente para poder ser utilizada en el mercado americano (FCC) y asiático (ARIB). Es una tecnología multiportadora que puede llegar a alcanzar tasas de hasta 128,6 kbps (en su versión para la banda CENELEC-A).
- *G.hnem* (20) se trata de un esfuerzo de la ITU-T por homogeneizar G3 y PRIME, aunque su implementación es computacionalmente más compleja que las anteriores. Su utilización actualmente es baja.
- *IEEE 1901.2* es la especificación NB-PLC del IEEE. Su utilización actualmente es muy baja.

PRIME destaca como una tecnología NB-PLC prometedora dentro de las disponibles. La especificación PRIME es abierta y gratuita, lo que permite que cualquier fabricante pueda desarrollar productos que se ajusten a ella, favoreciendo notablemente la interoperabilidad. Está siendo implantada en países como España, Portugal, Reino Unido, Polonia, Brasil o Australia, existiendo de manera oficial más de 12 M de contadores con tecnología PRIME instalados actualmente en campo (3). Además, en 2018 habrá alrededor de 15 M de contadores con tecnología PRIME desplegados únicamente en España (debido a la legislación española IET/290/2012) y la nueva versión del estándar (v1.4) amplía su dominio al resto del mundo, incluyendo bandas de frecuencia para los mercados americano y asiático.

Como ya se ha comentado, PRIME es el estándar que implementa la herramienta de simulación SimPRIME, en la que se basa este PFC. Por ello, el siguiente apartado profundiza en esta tecnología, resumiendo brevemente sus principales detalles técnicos.

2.2.2 PRIME

PRIME es una tecnología NB-PLC de segunda generación cuyo desarrollo fue inicialmente liderado por la PRIME Alliance, estando considerada la versión 1.3.6 de la especificación de las capas físicas, MAC y de Convergencia, como un estándar por la ITU-T desde 2012. La versión 1.4 expande el espectro de frecuencia utilizado para poder operar en los mercados americano y asiático e incluye algunas funciones para incrementar la robustez de la comunicación a nivel de la capa física y MAC.

La arquitectura de protocolos definida en PRIME se compone de tres capas: capa física (*PHYsical layer*), capa de enlace (*Data Link Layer – MAC*) y capa de convergencia (*Convergence layer –CS*)

Desde una perspectiva de la capa física, PRIME opera en la banda de 41-89 kHz (v1.3.6) o de 3-500 kHz (v1.4), usando la modulación *Orthogonal Frequency Division Multiplexing* (OFDM). PRIME no fija la modulación utilizada, la elección queda de la mano de los fabricantes, los cuales pueden usar las modulaciones DPSK, DQPSK o D8PSK. Además, permite la utilización de FEC para recuperarse ante posibles errores introducidos por el canal. A nivel de capa física, la velocidad de transmisión en redes PRIME puede ir desde los 5.4 kbps a 1028.8 kbps, dependiendo de la combinación de modulación digital y FEC utilizadas (también conocido como modo de comunicación).

A nivel MAC, se definen dos tipos de nodos: Base y de Servicio. Sólo se permite un Nodo Base por cada red PRIME, será el gestor principal de las comunicaciones. En terminología AMI, el Nodo Base es conocido como concentrador de datos (CNTR), mientras que la funcionalidad de nodo de servicio la desempeñan los contadores. PRIME contempla que un nodo de servicio pueda funcionar como Repetidor (*Switch*), para mitigar los efectos del ruido y de la atenuación. Así, la topología física de este tipo de redes es en bus, mientras que la topología lógica tiene típicamente forma de árbol. Un aspecto a considerar es que la posición de estos *switches* puede influir en el comportamiento de la red, como se discute en (21). A pesar de ello, el estándar no da indicaciones sobre qué criterios seguir para introducir *switches*.

En relación a los mecanismos MAC, aunque el estándar define un periodo de contienda (SCP) y un periodo libre de contienda (CFP), actualmente sólo el SCP está implementado por los fabricantes. A pesar de ello, se está investigando sobre los beneficios potenciales de usar CFP para nuevos servicios de Redes Eléctricas Inteligentes (22). En el SCP se utiliza CSMA/CA como técnica de acceso al medio.

La capa de convergencia clasifica el tráfico intercambiado entre los diferentes equipos asociándola con la conexión de la capa de enlace correspondiente. Se divide en dos capas:

- CPCS (*Common Part Convergence Sublayer*): Responsable de la fragmentación y el ensamblado de PDU del nivel superior.
- SSSCS (*Service Specific Convergence Sublayer*): Permite gestionar diferentes tipos de tráfico de nivel superior, IPv4, IPv6, IEC 61334-4-32 y una versión genérica para cualquier otro protocolo de nivel superior.

El control de flujo está implementado en PRIME mediante el establecimiento de una Unidad Máxima de Transmisión (MTU) y usando un procedimiento de **ventana deslizante** (WS). PRIME no fija dichos parámetros, la elección queda de la mano de los fabricantes. La MTU define la longitud en bytes del mayor paquete de datos que puede encapsular el nivel MAC. En caso de que la aplicación intente enviar un mensaje mayor, la capa CPCS fragmenta el mensaje en varios paquetes, ninguno de los cuales será mayor que la MTU. Cada uno de estos paquetes (también conocidos como fragmentos) está etiquetado con un identificador para que la parte receptora pueda reensamblarlos.

PRIME establece diferentes valores permitidos para el Tamaño de Ventana (WS). El valor del WS puede jugar un rol importante en el rendimiento de la red en términos de latencia. Además, los dispositivos PRIME pueden implementar o no capacidades de Repetición de Solicitudes (ARQ) para asegurar la correcta recepción de todos los mensajes. Dado que los parámetros ARQ son negociados en la fase de conexión, este proceso funciona de punto a punto entre el transmisor final y el receptor, siendo por tanto los *switches* transparentes al mismo.

En la capa de aplicación, DLMS/COSEM es el estándar utilizado sobre todas las tecnologías NB-PLC disponibles en el mercado. En concreto, COSEM (IEC 62056-61/62) es un perfil del protocolo de aplicación DLMS (IEC 62056-53) especialmente diseñado para la medición de energía (23) (24). Como tal, DLMS/COSEM incluye modelos de datos para representar parámetros comunes relacionados con la energía junto con un protocolo de comunicación diseñado para transmitir este tipo de información. Para el análisis de las trazas de mensajes será necesario conocer las características fundamentales de dicho protocolo. Por ello, el siguiente apartado profundiza en este protocolo, resumiendo brevemente sus principales detalles técnicos.

La Figura 4 muestra las tecnologías de comunicación y la torre de protocolos completa utilizada en la arquitectura básica del Proyecto PRICE-GEN para comunicaciones a través de concentrador, que será la arquitectura relevante en este PFC.

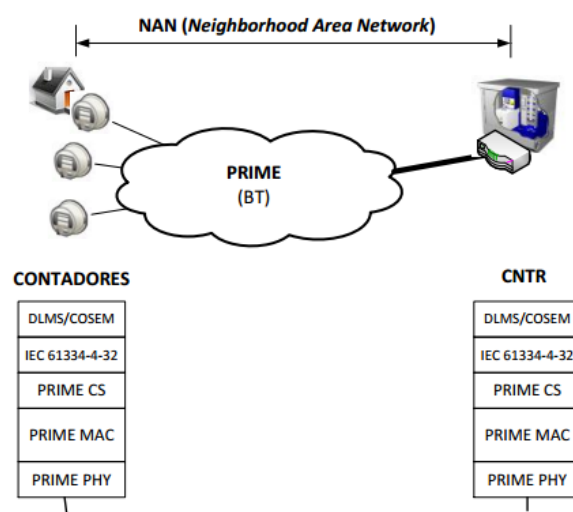


Figura 4 Torre de Protocolos Red PRIME

2.2.3 DLMS/COSEM

DLMS/COSEM (IEC 62056-53) (23) especifica un modelo de interfaz y protocolos de comunicación para el intercambio de datos con equipos de medida. COSEM es el modelo de datos y DLMS es el protocolo de capa de aplicación de PRIME que define el intercambio de datos entre equipos de medida.

El intercambio de datos entre sistemas de almacenamiento de datos y dispositivos de medida está basado en el modelo cliente/servidor, donde los sistemas de almacenamiento de datos juegan el papel del cliente y los dispositivos de medida hacen de servidor. El cliente envía solicitudes de servicio (*service requests*) al servidor, que envía respuestas de servicio (*service responses*). Además, el servidor puede iniciar solicitudes de servicio de manera no solicitada para informar al cliente de eventos o para enviar datos en condiciones pre-configuradas.

La Figura 5 ilustra el intercambio de mensajes entre cliente y servidor.

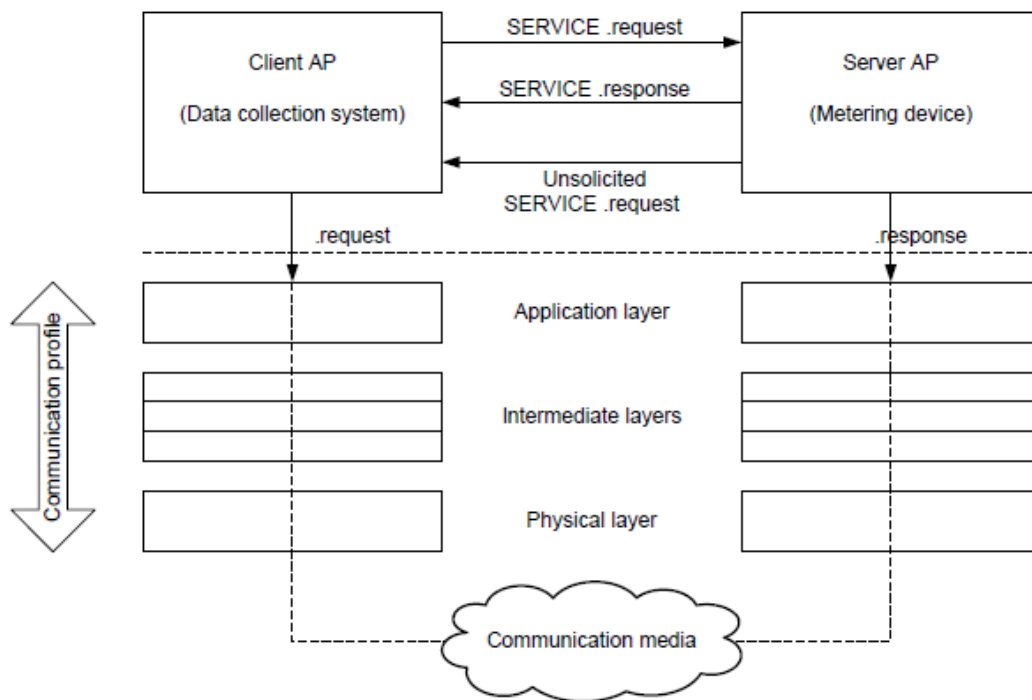


Figura 5 Relación Cliente/servidor

A las entidades DLMS/COSEM (clientes, servidores y terceros) se les debe asignar un “*system title*” de manera permanente. El “*system title*” puede identificar tanto a un cliente o un servidor como a terceros que puedan acceder al servidor por medio de un cliente y, además debe ser único.

Las direcciones de los dispositivos físicos pueden ser pre-configuradas o pueden ser asignadas durante un proceso de registro que también incluya la unión entre las direcciones y los “*system titles*”.

Una sesión de comunicación consta de tres fases, como se muestra en la Figura 6.

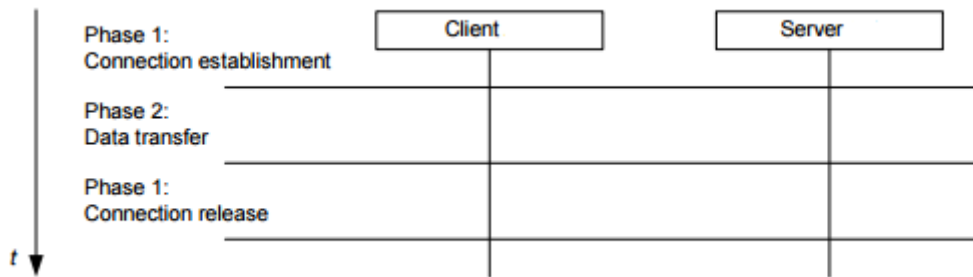


Figura 6. Comunicación Cliente/Servidor a nivel de aplicación

- Primero, se establece una conexión a nivel de aplicación, llamada Asociación de Aplicación (AA), entre un cliente y un servidor *Application Entity* (AE).
- Una vez que la AA se ha establecido, puede comenzar el intercambio de mensajes.
- Cuando el intercambio de mensajes termina, la AA se libera.

Una iteración típica en DLMS/COSEM consiste en una consulta del valor que toma un determinado atributo específico en el contador (por ejemplo, informe S02) y una respuesta de este con el valor codificado según las reglas de codificación *eXternal Data Representation* (A-XDR)

Para el caso que nos atañe es necesario determinar la consulta asociada a la solicitud de un informe de consumo (S02) desde del cliente/concentrador al servidor/contador, así como el flujo de comunicación establecido entre ambas entidades para su transmisión.

En la siguiente sección se detalla el formato de mensajes que existe en una comunicación en redes PRIME, así como los parámetros más importantes asociados a cada uno de los protocolos/capas intervinientes en la comunicación y, en especial, el formato que presenta cualquier comunicación existente entre dos entidades (concentrador, contador).

2.3 Formato de trazas

Como se ha mencionado anteriormente, el objetivo del presente PFC es validar los resultados obtenidos en el simulador con los obtenidos mediante el análisis de trazas de tráfico capturadas en campo. En este apartado, y una vez vistos los apartados 2.2.2 y 2.2.3, es importante recalcar el formato de traza de mensajes y la relación existente entre los diferentes protocolos que intervienen en el proceso de comunicación en este tipo de redes.

PRIME define una Unidad Máxima de Transmisión (MTU), sin embargo, no fija la MTU utilizada, la elección queda de la mano de los fabricantes. La MTU define la longitud en bytes del mayor paquete de datos que puede encapsular el nivel MAC. En caso de que la aplicación intente enviar un mensaje mayor, la capa CPCS fragmenta el mensaje en varios paquetes, ninguno de los cuales será mayor que la MTU. Cada uno de estos paquetes (también conocidos como fragmentos) está etiquetado con un identificador para que la parte receptora pueda re-ensamblarlos. La Figura 7 ilustra la fragmentación de un mensaje DLMS/COSEM en varias trazas PRIME. Dicha información es fundamental para poder entender los archivos de trazas proporcionados.

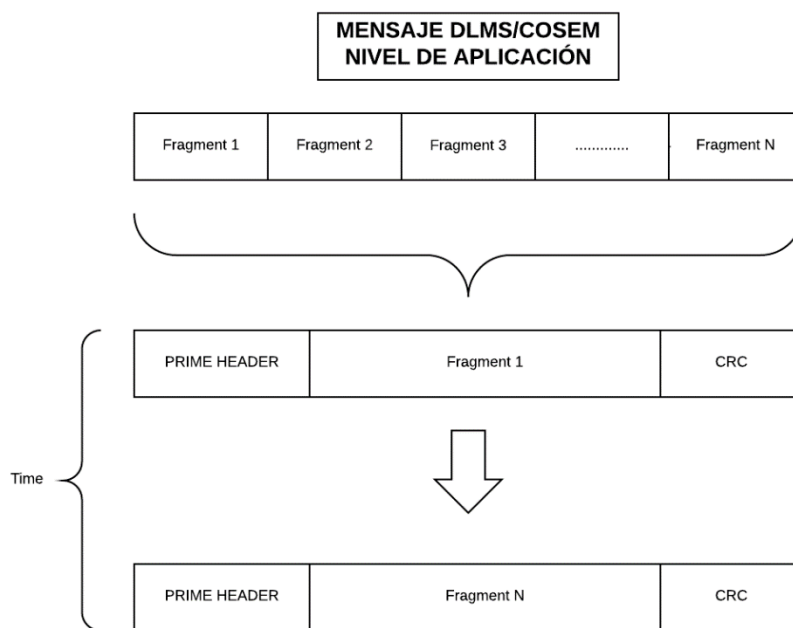


Figura 7 Fragmentación de mensajes DLMS/COSEM que superan la MTU establecida

El formato de las trazas de mensajes es imprescindible debido a la importancia que suponen para el presente PFC. Los fabricantes de dispositivos tienen libertad a la hora de registrar los mensajes en sus equipos, por lo que el análisis de las trazas va ligado al fabricante del equipo de medida, es decir, no es estándar, por lo que decidimos soportar dos de los principales fabricantes del mercado (ZIV y CIRCUTOR).

Para entender el formato de las trazas es necesario dar una visión global de los datos que se esperan analizar. La Figura 8 ilustra la estructura de una traza PRIME.

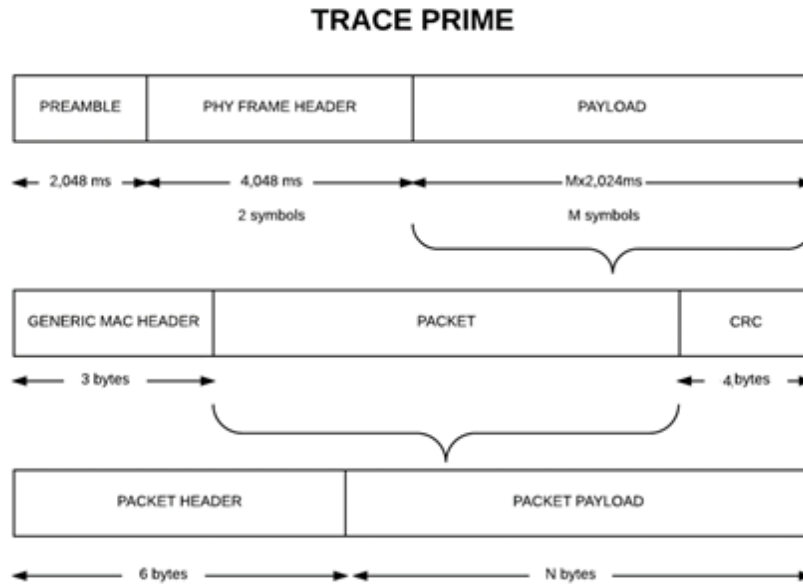


Figura 8 Estructura de datos de una traza PRIME

El estándar PRIME define dos tipos de mensajes, **DATA** y **CONTROL**. El tipo de mensaje queda identificado mediante el campo PKT.C, englobado en la cabecera de paquete descrita en la Figura 8. Dicha cabecera es especialmente relevante para el desarrollo de este PFC, por lo que la Figura 9 muestra su estructura.

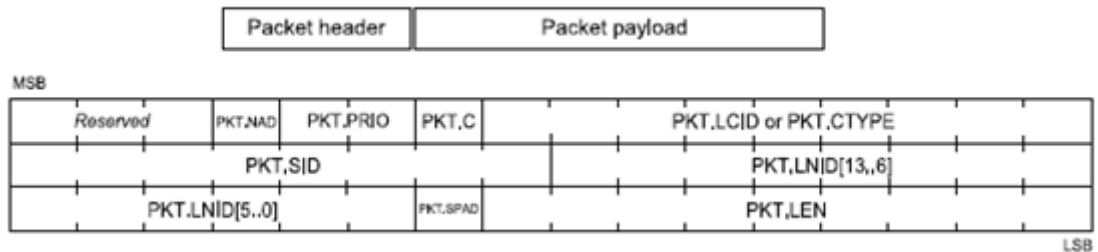


Figura 9 Estructura paquetes de datos PRIME

En la Tabla 1 se describe cada uno de los campos con más detalle, ya que muchos de ellos son necesarios para nuestro análisis.

Name	Length	Descripción
Reserved	3 bits	Always 0 for this version of the Recommendation. Reserved by ITU-T.
PKT.NAD	1 bit	No aggregation at destination <ul style="list-style-type: none"> • If PKT.NAD=0 the packet may be aggregated with other packets at destination. • If PKT.NAD=1 the packet may not be aggregated with other packets at destination.
PKT.PRIO	2 bits	Indicates a packet priority between 0 and 3
PKT.C	1 bit	Control <ul style="list-style-type: none"> • If PKT.C=0 it is a data packet. • If PKT.C=1 it is a control packet.
PKT.LCID / PKT.CTYPE	9 bits	Local connection identifier or control type <ul style="list-style-type: none"> • If PKT.C=0, PKT.LCID represents the local connection identifier of a data packet. • If PKT.C=1, PKT.CTYPE represents the type of the control packet.
PKT.SID	8 bits	Switch identifier <ul style="list-style-type: none"> • If HDR.DO=0, PKT.SID represents the SID of the packet source. • If HDR.DO=1, PKT.SID represents the SID of the packet destination.
PKT.LNID	14 bits	Local node identifier <ul style="list-style-type: none"> • If HDR.DO=0, PKT.LNID represents the LNID of the packet source. • If HDR.DO=1, PKT.LNID represents the LNID of the packet destination.
PKT.SPAD	1 bit	Indicates if padding is inserted while encrypting payload. Note that this bit is only of relevance when security profile is used.
PKT.LEN	9 bits	Length of the packet payload in bytes

Tabla 1 Campos de cabecera asociados a paquetes PRIME

Para conseguir el objetivo global de este PFC, es necesario analizar los diferentes parámetros asociados a una red PRIME en las trazas de mensajes registradas por cada fabricante. Del análisis de las trazas se van a obtener indicadores de calidad de las comunicaciones (en concreto el tiempo de lectura TTR_i de cada contador), así como parámetros de implementación PRIME como la MTU y WS.

Los parámetros necesarios para el procesado de cada traza se detallan a continuación:

NIVEL FÍSICO

Scheme: El parámetro Scheme especifica el esquema de transmisión que se utilizará para la *MAC Protocol Data Unit* (MPDU). Puede tener cualquiera de los siguientes valores:

- 0: DBPSK
- 1: DQPSK
- 2: D8PSK
- 3: not used
- 4: DBPSK + convolutional code
- 5: DQPSK + convolutional code
- 6: D8PSK + convolutional code
- 7: not used.

El parámetro **Level** especifica el nivel de señal en el que la capa PHY recibió la PSDU. Puede tomar uno de dieciséis valores:

- 0: ≤ 70 dBuV
- 1: ≤ 72 dBuV
- 2: ≤ 74 dBuV
- ...
- 15: > 98 dBuV

El parámetro **SNR** se refiere a la relación señal-ruido, definida como la relación entre el nivel de la señal recibida medida y el nivel de ruido de la última PDU recibida. Puede tomar uno de ocho valores:

- 0: ≤ 0 dB
- 1: ≤ 3 dB
- 2: ≤ 6 dB
- ...
- 7: > 18 dB

NIVEL MAC

Link: Recepción o envío de la traza según el sentido de la comunicación: *Downlink/uplink*. Dicho campo se especifica en la cabecera MAC. En la Figura 10 el campo HDR.DO especifica el sentido de la comunicación.

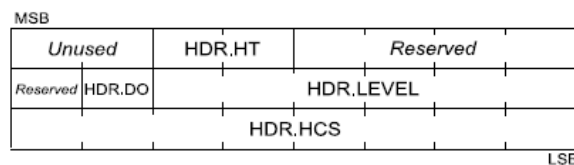


Figura 10 Cabecera correspondiente a la capa MAC en tráfico PRIME

Downlink/uplink

HDR.DO=1 if the MAC PDU is downlink

HDR.DO=0 if the MAC PDU is uplink

SNA: MAC address del nodo base. Identifica la red.

LSID (8 bits) = identificadores de *switch* local que posee cada nodo en la red. El nodo base asigna el identificador local de red (LNID) durante el “*promotion process*”. Cada *Switch node* en la red es identificado por el SNA del nodo base y el LSID. LSID=0x00 está reservado al nodo base.

LNID (14 bits): Durante el *Registration Process*, cada nodo recibe un identificador local LNID. El LNID identifica directamente un nodo entre todos los nodos que cuelgan de un *switch*. La combinación del LNID y del SID (LSID del

nodo *switch* del que cuelga) forma el NID (22 bits). Este identificador identifica un nodo de forma única en la red. LNID = 0x0000 está reservado para el nodo base. Para todas las solicitudes de registro aceptadas con éxito, el nodo base (i.e., el concentrador) debe adjudicar un LNID único dentro del dominio del *switch* a través del cual se realiza el *attach*. La combinación del LNID con el SID del nodo a través del cual se registró formaría el NID para el *service node* registrado.

LCID: Identificador para identificar una conexión única entre dos nodos. La combinación del NID y LCID forman un identificador de conexión de 31 bits. El LCID se asigna durante el establecimiento de conexión.

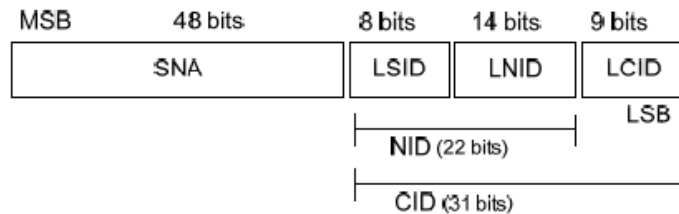


Figura 11 Identificadores de entidad en redes PRIME

Conociendo los datos descritos anteriormente y procesando los mismo podemos obtener la estructura lógica en árbol de la red que estamos analizando, tal y como se muestra en las Figuras 12 y 13.

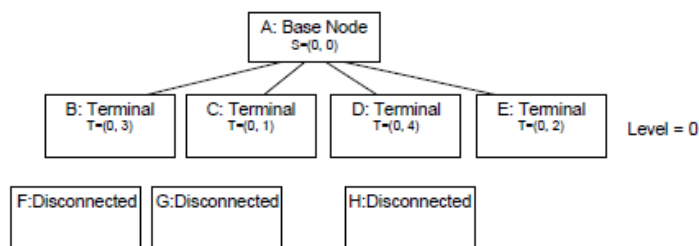


Figura 12. Estructura Lógica de la Red PRIME

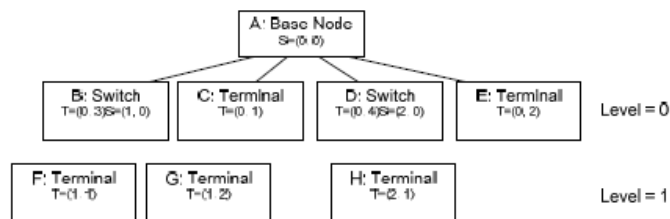


Figura 13 Estructura Lógica red PRIME

EUI-48: Dirección MAC de cada nodo, definida en IEEE 802-2001. Si queremos identificar cada NID en la red PRIME con su correspondiente dirección física debemos analizar los tipos de paquetes identificados como **REG_RSP**. Este tipo de mensajes no interactúa con la capa de aplicación, son mensajes de tipo CONTROL

PNA: *MAC address* de un nodo que realiza las funciones de *switch*. Cuando un nodo no puede alcanzar a cualquier otro *switch*, puede enviar tramas de promoción

(*promotion-needed*) de modo que un terminal puede ser ascendido y comenzar a funcionar como *switch*.

Si queremos identificar que nodos son promovidos a *Switch*, la relación directa entre NID y MAC se encuentra en los mensajes **PRO_REQ_S**. Este tipo de mensajes no interactúa con las capa de aplicación, son mensajes de tipo CONTROL

NIVEL APLICACIÓN

Time: Tiempo de registro.

PACKET PAYLOAD: Registra la iteración a nivel de aplicación. Una iteración típica en DLMS/COSEM consiste en una consulta del valor que toma un determinado atributo específico en el contador y una respuesta de este con el valor codificado en según las reglas de codificación A-XDR (*eXternal Data Representation*) nuestro objetivo es determinar el formato de consulta utilizado para solicitar el informe de consumo S02 y el intercambio de mensajes entre ambas entidades hasta transferir todos los datos necesarios.

Una vez identificadas las trazas de tipo data, el cálculo del *TTRi* comienza con la solicitud del informe S02 desde el CNTR, que en términos de DLMS/COSEM se indica mediante el mensaje hexadecimal **c001c100070100630100ff**. Dicho mensaje se debe encontrar dentro del *Packet PAYLOAD* representado en la Figura 8. El cálculo del *TTRi* finaliza con el envío por parte del contador del último bloque correspondiente al informe S02, que en términos de DLMS/COSEM se indica mediante el mensaje hexadecimal **c402c101**. Dicho mensaje se debe encontrar dentro del *Packet PAYLOAD*.

Como hemos explicado con anterioridad, en el caso que el nivel de aplicación intente enviar información con un tamaño en bytes superior al MTU establecido en el establecimiento de la conexión, dicho mensaje será fragmentado, por lo que es necesario analizar el intercambio de mensajes que se produce al solicitar un informe de consumo S02. La Figura 14 muestra el intercambio de mensajes originado ante la solicitud de un informe de consumo (S02) por parte del cliente/concentrador.

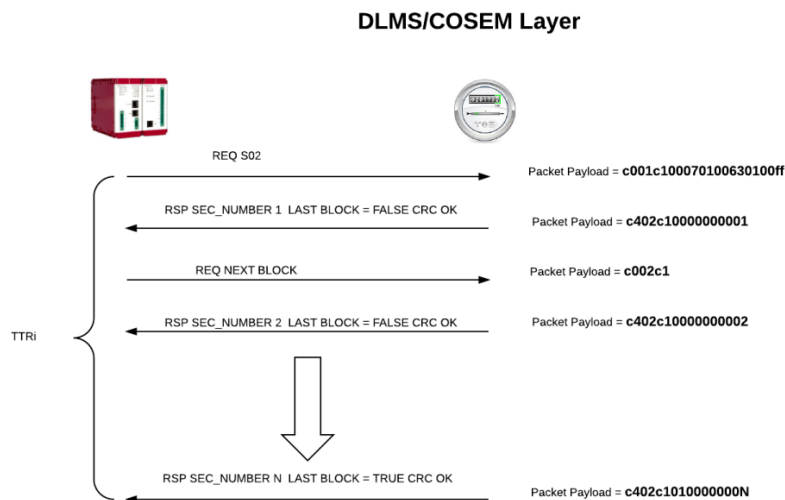


Figura 14 Intercambio de mensajes a nivel de aplicación

El contenido del *Packet Payload* de las trazas PRIME a nivel DLMS/COSEM ilustrado en la Figura 14 queda explicado a continuación:

- Request (Concentrador)
 - Service.request → **c001c1**
 - Service.requestS02 → **c001c100070100630100ff**
 - Service.request.NextBlock → **c002c1**
- Response (Contador)
 - Service.response.lastblock_false → **c402c100** (los cuatro bytes siguientes indican el número de secuencia correspondiente)
 - Service.response.lastblock_true → **c402c101** (los cuatro bytes siguientes indican el número de secuencia correspondiente)

En el presente PFC vamos a considerar trazas de ZIV y de CIRCUTOR, ambos fabricantes de dispositivos para redes AMI. En los siguientes apartados realizaremos el desglose de las diferentes trazas aportadas. Es necesario indicar que el procesado inicial de la traza será diferente para cada tipo de traza, ya que los formatos de los mismos son diferentes, pero todas ellas engloban los parámetros descritos en esta sección.

Las infraestructuras de las que se dispone para obtener estas trazas son:

Panel de Telemidia.

El panel de telemidida instalado en la UC3M, se ha montado principalmente como soporte a la creación y validación de las herramientas PRIME. El panel se compone de:

- Cuatro contadores monofásicos de la marca ZIV, modelo 5CTME2C47536AVF, los cuales disponen de interfaz de comunicaciones PLC bajo el estándar PRIME
- Un concentrador de comunicaciones PRIME PLC trifásico de la marca ZIV, modelo 4CCT-EA8-032126BJ

La Figura 15 muestra el esquema de conexionado del panel. El concentrador va además conectado por medio de un cable Ethernet a una red IP para su gestión. Es importante recalcar que el panel está diseñado con fines de investigación en el área de las comunicaciones, y por tanto dispone del inter-conexionado mínimo necesario para la comunicación entre los contadores y el concentrador.

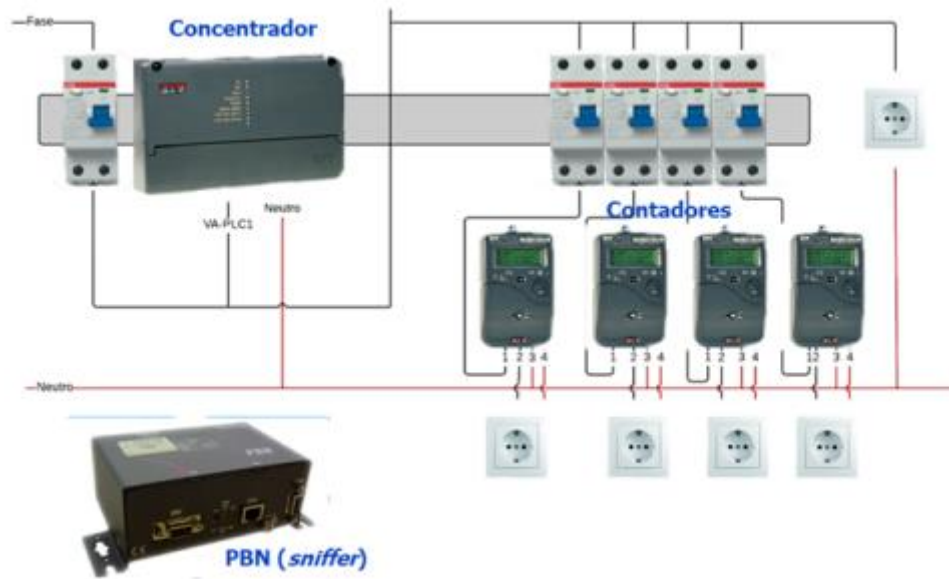


Figura 15 Panel de Telemedida

LINTER

Laboratorio de Integración de Redes (LINTER), está equipado con contadores inteligentes y concentradores de los principales fabricantes del mercado (25). En la Figura 16 se muestra un esquema del equipamiento desplegado en el laboratorio.

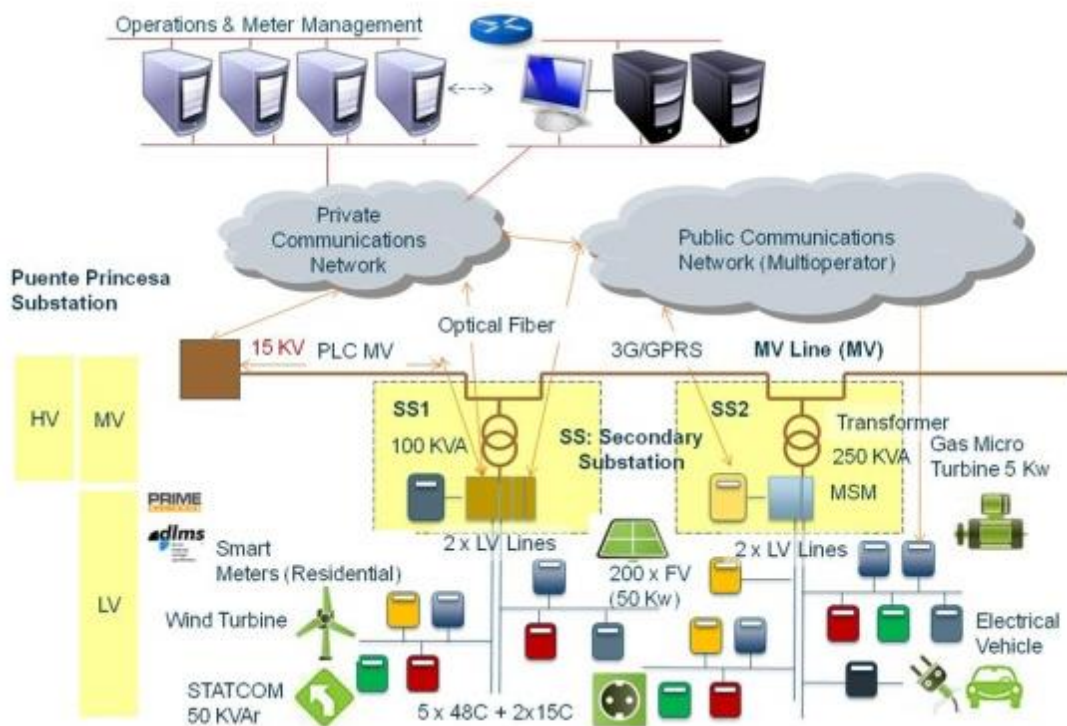


Figura 16 Estructura del laboratorio de integración de redes (LINTER)

Además, también se dispondrá de trazas capturadas en redes en explotación.

2.3.1 ZIV

Una de las trazas a analizar está tomada de los equipos de medida del fabricante ZIV. Tomando como referencia una traza real capturada en el panel, plasmada en el siguiente recuadro, vemos el formato utilizado por el fabricante para registrar los parámetros descritos con anterioridad. Para este caso disponemos de la traza completa PRIME codificada en hexadecimal. La Figura 17 muestra el formato utilizado por ZIV.

```
[RX] 2016-09-20 18:08:34 5758.542472(0.026688) dbpsk_f rxpow:6.23016(Vrms) evm:10.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.172148 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[a2 90 03 40 00 06 00
10 | 00 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00
20 | 00 00 02 08 09 0c 07 e0 09 13 01 14 00 00 00 80
30 | 00 80 11 00 06 00 00 00]dd f3 59 00
```

Figura 17 Ejemplo de traza PRIME asociada a ZIV

La Figura 18 muestra todos los parámetros asociados a la capa **PRIME PHY**. En este caso no es necesario analizar las cabeceras de la traza debido a que la información necesaria ya se encuentra reflejada en texto claro facilitando la labor de procesado de la misma.

```
[RX] 2016-09-20 18:08:34 5758.542472(0.026688) dbpsk_f rxpow:6.23016(Vrms) evm:10.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.172148 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[a2 90 03 40 00 06 00
10 | 00 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00
20 | 00 00 02 08 09 0c 07 e0 09 13 01 14 00 00 00 80
30 | 00 80 11 00 06 00 00 00]dd f3 59 00
```

Figura 18 Parámetros Capa Física ZIV

La Figura 19 ilustra todos los parámetros asociados a la capa **PRIME MAC**, tanto en texto claro como su correspondiente codificación en hexadecimal. En este caso no es necesario analizar las cabeceras de la traza debido a que la información necesaria ya se encuentra reflejada en texto claro facilitando la labor de procesado de la misma.

```
[RX] 2016-09-20 18:08:34 5758.542472(0.026688) dbpsk_f rxpow:6.23016(Vrms) evm:10.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.172148 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[a2 90 03 40 00 06 00
10 | 00 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00
20 | 00 00 02 08 09 0c 07 e0 09 13 01 14 00 00 00 80
30 | 00 80 11 00 06 00 00 00]dd f3 59 00
```

Figura 19 Parámetros Capa MAC ZIV

La Figura 20 ilustra todos los parámetros asociados a la capa de aplicación, **DLMS/COSEM, packet payload**, que se observa van entre corchetes. Todos los datos correspondientes a esta capa se encuentran en formato hexadecimal por lo que será necesario utilizar las codificaciones de mensajes utilizadas por dicho protocolo para

buscar los patrones necesarios para el cálculo del *TTRi*. Como hemos visto en apartados previos tenemos que tener en cuenta que una consulta a nivel de aplicación comprende varias trazas a nivel PRIME.

```
[RX] 2016-09-20 18:08:34 5758.542472(0.026688) dbpsk_f rxpow:6.23016(Vrms) evm:10.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.172148 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[a2 90 03 40 00 06 00
10 | 00 00 00 06 00 00 00 00 06 00 00 00 06 00 00
20 | 00 00 02 08 09 0c 07 e0 09 13 01 14 00 00 00 80
30 | 00 80 11 00 06 00 00 00]dd f3 59 00
```

Figura 20 Parámetros Capa de aplicación ZIV

Uno de los parámetros con mayor interés para nuestro caso es el tiempo. La Figura 21 ilustra su formato. Disponemos de dos parámetros para calcular el eje de tiempos, cada uno de los cuales presenta un formato característico.

```
[RX] 2016-09-20 18:08:34 5758.542472(0.026688) dbpsk_f rxpow:6.23016(Vrms) evm:10.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.172148 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[a2 90 03 40 00 06 00
10 | 00 00 00 06 00 00 00 00 06 00 00 00 06 00 00
20 | 00 00 02 08 09 0c 07 e0 09 13 01 14 00 00 00 80
30 | 00 80 11 00 06 00 00 00]dd f3 59 00
```

Figura 21 Tiempo de registro de la recepción/envío de trazas ZIV

2.3.2 CIRCUTOR

Otra de las trazas a analizar está tomada de los equipos de medida del fabricante CIRCUTOR.

Como se puede apreciar en la Figura 22 las trazas de mensajes extraídas de un concentrador de CIRCUTOR presentan valores separados por comas para los parámetros de la capa PRIME y representación hexadecimal para el protocolo DLMS/COSEM. Este formato de trazas presenta diferencias frente al formato ZIV visto en el apartado previo, por lo que el procesamiento de las trazas será diferente, aunque los campos a buscar sean los mismos.

```
PCTime,Time,DeltaTime,HType,DO,LEVEL,HCS,SID,LNID,PRI0,Len,CRC,Txt,Scheme,PHYLevel,PHYSNR,PHYRQ
00:30:28.0,369009472,791,Data,UP,000,OK,07,20ae,4,64,OK,LCID: 0x0100 DATA:
ac812503900101c402c10000000040081d5060000008c02080,BPSK-FEC,14,7,-,
```

Figura 22 Ejemplo de traza PRIME asociada a CIRCUTOR

La Figura 23 ilustra todos los parámetros asociados a la capa **PRIME PHY**.

```
PCTime,Time,DeltaTime,HType,DO,LEVEL,HCS,SID,LNID,PRIO,Len,CRC,Txt,Scheme,PHYLevel,PHYSNR,PHYRQ
00:30:28.0,369009472,791,Data,UP,000,OK,07,20ae,4,64,OK,LCID: 0x0100 DATA:
ac812503900101c402c10000000040081d5060000008c02080,BPSK-FEC,14,7,-,
```

Figura 23 Parámetros Capa Física CIRCUTOR

La Figura 24 ilustra todos los parámetros asociados a la capa **PRIME MAC**. En este caso no es necesario analizar las cabeceras de la traza debido a que la información necesaria ya se encuentra reflejada en texto claro facilitando la labor de procesado de la misma.

```
PCTime,Time,DeltaTime,HType,DO,LEVEL,HCS,SID,LNID,PRIO,Len,CRC,Txt,Scheme,PHYLevel,PHYSNR,PHYRQ
00:30:28.0,369009472,791,Data,UP,000,OK,07,20ae,4,64,OK,LCID: 0x0100 DATA:
ac812503900101c402c10000000040081d5060000008c02080,BPSK-FEC,14,7,-,
```

Figura 24 Parámetros Capa MAC CIRCUTOR

La Figura 25 ilustra todos los datos asociados a la capa **DLMS/COSEM**. Todos los datos correspondientes a esta capa se encuentran en formato hexadecimal.

```
PCTime,Time,DeltaTime,HType,DO,LEVEL,HCS,SID,LNID,PRIO,Len,CRC,Txt,Scheme,PHYLevel,PHYSNR,PHYRQ
00:30:28.0,369009472,791,Data,UP,000,OK,07,20ae,4,64,OK,LCID: 0x0100 DATA:
ac812503900101c402c10000000040081d5060000008c02080BPSK-FEC,14,7,-,
```

Figura 25 Parámetros Capa de aplicación CIRCUTOR

Uno de los parámetros con mayor interés para nuestro caso es el tiempo. La Figura 26 muestra su formato. Disponemos de dos parámetros para calcular el eje de tiempos, cada uno de los cuales presenta un formato característico.

```
PCTime,Time,DeltaTime,HType,DO,LEVEL,HCS,SID,LNID,PRIO,Len,CRC,Txt,Scheme,PHYLevel,PHYSNR,PHYRQ
00:30:28.0,369009472,791,Data,UP,000,OK,07,20ae,4,64,OK,LCID: 0x0100 DATA:
ac812503900101c402c10000000040081d5060000008c02080,BPSK-FEC,14,7,-,
```

Figura 26 Tiempo de registro de la recepción/envío de trazas CIRCUTOR

2.4 Simuladores de redes de comunicaciones

Esta sección analiza algunas de las soluciones disponibles en el mercado para desarrollar herramientas de simulación de redes de comunicaciones. Existen multitud de alternativas. Dentro del ámbito de los simuladores de redes de comunicaciones propiamente dichos, esta sección se centra en los siguientes, por tratarse de los tres con más aceptación en el mercado:

- Riverbed Modeler (anteriormente, OPNET) (26).
- NS3 (27).
- OMNeT++ (28).

Sin embargo, también pueden desarrollarse herramientas de simulación en base a *software* de cálculo (p.ej., Microsoft Excel o Matlab (29)) o en lenguajes de *scripting* (p. ej., Python (30)), dependiendo de las características de las tecnologías de simulaciones a simular.

2.4.1 Riverbed Modeler

Riverbed Modeler (antes conocido como OPNET Modeler), producto actualmente perteneciente a Riverbed, es un potente simulador de redes de comunicaciones basado en eventos discretos que incluye modelos muy precisos de tecnologías de comunicaciones como, por ejemplo, las celulares (2G, 3G). Sin embargo, no existe modelo aun para redes PLC, aunque podría desarrollarse partiendo de los módulos ya implementados (p. ej., Ethernet). El lenguaje de programación que se utiliza en este simulador es C++.

La versión comercial de Riverbed Modeler requiere del pago de una licencia que incluye la utilización del simulador así como soporte técnico y formación. En cualquier caso, las universidades y centros de investigación actualmente tienen la posibilidad de obtener una licencia de pruebas de 6 meses, que puede renovarse siempre y cuando se hagan públicos los resultados de investigación obtenidos.

2.4.2 NS3

NS3 es la última versión del popular simulador de redes de comunicaciones NS2. NS2 y NS3 son bastante comunes en entornos académicos, llegando a ser casi de uso obligatorio para que los resultados obtenidos al simular determinados protocolos tengan validez dentro de la comunidad investigadora (p.ej., para protocolos del IETF – *Internet Engineering Task Force*).

Existe una implementación de PLC para NS3 (31) que podría utilizarse como referencia y adaptarse a PRIME.

NS es software libre que se ofrece bajo la versión 2 de la GNU (*General Public License*).

2.4.3 OMNeT++

OMNeT++ es un simulador de redes de comunicaciones modular basado en eventos discretos. En OMNeT++ existen dos tipos de módulos:

- Simples: que representan elementos atómicos;
- Compuestos: compuestos – valga la redundancia – por una combinación de módulos simples.

En OMNeT++ se manejan tres tipos de ficheros:

- .cpp: definen funcionalidad, utilizando C++ como lenguaje de programación;
- .ned: definen la topología de red;
- .ini: configuran las simulaciones a realizar (parámetros de entrada, parámetros de salida, duración, etc.).

No se ha encontrado implementación oficial de PRIME para OMNeT++. Sin embargo, como ya se ha comentado, en (4) se presenta la versión inicial de SimPRIME, una implementación de PRIME para OMNeT++ desarrollada a partir de la implementación de Ethernet disponible en el INET Framework.

OMNeT++ se distribuye bajo *Academic Public License*, que no permite su uso comercial. La versión comercial de OMNeT++ es OMNEST.

2.5 Simulador SimPRIME

2.5.1 Visión Global

Dado el aumento de la popularidad de las redes PRIME, una amplia cantidad de herramientas de simulación están siendo desarrolladas para evitar el alto coste y la complejidad de los laboratorios de pruebas o de los despliegues en campo. Prueba de ello son los numerosos trabajos de investigación que se han llevado a cabo recientemente en este sentido:

- (32) presenta, por ejemplo, un análisis interesante sobre el rendimiento disponible en redes PLC multi-salto. Sin embargo, no se considera ruido en el canal, por lo que la comunicación no presenta errores.
- (33) propone un método para abstraer la capa PHY de las simulaciones a través de curvas de PER vs SNR. Sin embargo, esas curvas se calculan para un tamaño de paquete fijo, lo que hace que los resultados no sean realistas.
- (34) presenta un simulador de redes PRIME basado en OMNeT++. Sin embargo, este simulador no considera la capa de aplicación, sino que se fija exclusivamente en probabilidades de error. Este tipo de estudios pueden tener interés, pero los DSO se fijan en otro tipo de parámetros representativos del rendimiento de la red, tales como la latencia.
- (35) presenta también un simulador de redes PRIME basado en OMNeT++. Dicho simulador se utiliza concretamente para simular la telecarga de firmware en redes PRIME. Al igual que (36), abstrae la capa PHY de las simulaciones a través de curvas de PER vs SNR. En este caso también se toma un tamaño de paquete fijo para calcular la PER pero además es el máximo posible, lo que penaliza al tráfico de control, que se suele caracterizar por el reducido tamaño de sus paquetes.

El simulador de redes PRIME en el que se centra este PFC (simPRIME) utiliza MATLAB para abstraerle el nivel PHY a OMNeT++, pero trabaja a nivel de BER y cubre no sólo hasta el nivel de enlace sino hasta el de aplicación. El siguiente apartado resume brevemente sus principales detalles de implementación.

2.5.2 SimPrime

Como ya se ha comentado, el núcleo del simulador SimPRIME fue desarrollado como parte de la tesis doctoral de Javier Matanza (4). Este simulador combina MATLAB y OMNeT++ para modelar los efectos de la capa PHY y de las capas superiores (MAC, LLC y aplicación) respectivamente, como ilustra la Figura 27.

Respecto a la torre de protocolos, y en consonancia con lo comentado en la sección 2.2.2, la implementación de las capas PHY, MAC y LLC se ajusta a la especificación PRIME, mientras que la capa de aplicación se modela en base a DLMS/COSEM.

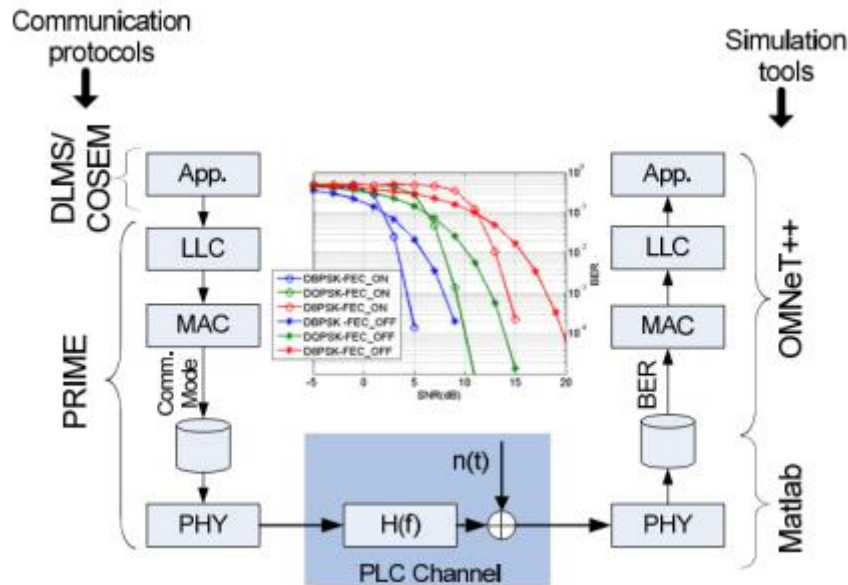


Figura 27 Esquema arquitectura de simulación utilizada

La interacción entre MATLAB y OMNeT++ se lleva a cabo de la siguiente manera: conocidos la potencia de transmisión y el nivel de ruido base, cuando un nodo envía un mensaje a otro, se calcula la potencia recibida en base a una matriz de atenuaciones que contiene la atenuación entre cada par de nodos. Con la potencia recibida y la potencia de ruido base se obtiene la SNR, tal y como muestra la siguiente ecuación.

$$SNR[dB] = P_{RX}[dBW] - P_N[dBW] = P_{TX}[dBW] - Att[dBW] - P_N[dBW]$$

Conocidos la SNR y la constelación utilizada, se obtiene la BER mediante las curvas SNR frente a BER mostradas en la Figura 27. Dichas curvas se calculan *a priori* en base al estándar usando para ello MATLAB. OMNeT++ utiliza el valor de BER resultante para decidir si el mensaje recibido contiene errores (y por tanto debe ser descartado) o por el contrario fue recibido correctamente (y puede ser procesado por las capas superiores).

La versión original del simulador ha sido extendida para poder simular topologías lógicas reales obtenidas a partir del procesamiento del fichero de topología estándar que proporcionan los concentradores (S11.xml). En esta misma línea, el simulador original también ha sido extendido para que las atenuaciones entre cada par de nodos no sólo puedan obtenerse en base a teoría de líneas de transmisión (37), sino que también puedan obtenerse a partir de determinadas BER fijadas para cada par de nodos de la red dependiendo del nivel lógico en el que se encuentren.

Para poder realizar un uso más amigable del simulador de forma que sea más atractivo y fácil de usar se ha desarrollado una aplicación Web que permite ejecutar simulaciones y visualizar resultados de forma amigable (5). Una de las finalidades del presente PFC es añadir un nuevo módulo a dicha aplicación web para poder procesar,

almacenar y representar los valores de $TTRi$ necesarios para poder validar los resultados. En la Figura 28 se muestra las entradas y salidas actuales de la aplicación web a las que habrá que añadir la salida $TTRi$.

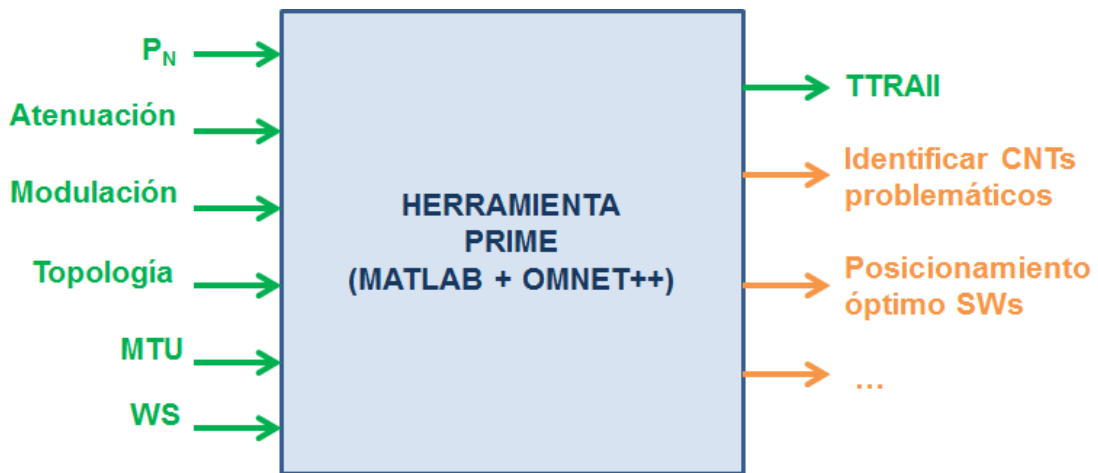


Figura 28 Resumen gráfico de entradas y salidas del simulador (5)

Las funcionalidades del simulador fueron extendidas para permitir simular escenarios reales desplegados en campo mediante el desarrollo de un nuevo módulo *software* que procesa mapas de redes de distribución eléctrica en formato *Shapefile* (6). La Figura 29 presenta un diagrama de la aplicación Web junto con el módulo Shapefile incorporado. Dicho módulo será utilizado para simular un entorno real en el presente PFC con el fin de poder comparar los resultados obtenidos con los procedentes del análisis de trazas de tráfico reales.

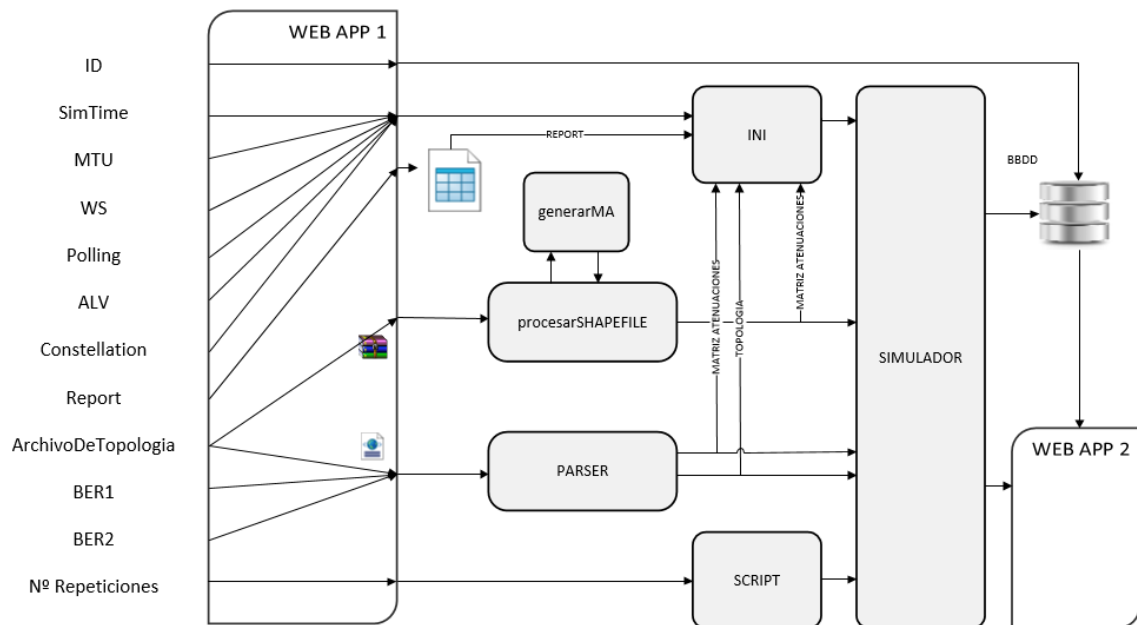


Figura 29 Aplicación Web, simPRIME y módulo para procesar Shapefiles (6)

2.6 Tecnologías utilizadas para el desarrollo

Un lenguaje de programación no es más que un sistema estructurado y diseñado principalmente para que las máquinas y computadoras se entiendan entre sí y con nosotros, los humanos. Contiene un conjunto de acciones consecutivas que el ordenador debe ejecutar.

En esta sección damos algunos detalles de las tecnologías utilizadas para desarrollar la aplicación web, presentada justo al final de la sección anterior. Es relevante tener una visión global de ellas en tanto en cuanto habrá que extender su funcionalidad. Además el grueso del PFC será desarrollado en Python, por lo que dicha sección es especialmente importante.

2.6.1 Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad (38).

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
- Además, Python es gratuito, incluso para propósitos empresariales.

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

2.6.2 Django

Django es un *framework* para Python mantenido por la *Django Software Foundation* (39), orientado al desarrollo de aplicaciones web utilizando el patrón MVC. Está diseñado para facilitar una implementación y un despliegue rápidos de aplicaciones, facilitando la interacción del sistema implementado con la base de datos. Es fácilmente escalable y ofrece por defecto características de seguridad como protección contra inyecciones SQL o un sistema de gestión de cuentas de usuario (40).

Este *framework* induce a un diseño modular de aplicaciones para fomentar la reutilización del código.

Además, Django incluye un sistema de gestión (`django-admin` o `manage.py`). Una herramienta de línea de comandos para realizar tareas administrativas.

Incluye también otros componentes útiles como:

- Serialización de formularios (genera formularios a partir de modelos de datos).
- Validación de formularios según los tipos de datos admitidos y la obligatoriedad de los campos.
- Sistema de plantillas con herencia entre las mismas.

Aunque Django sigue un patrón de diseño MVC, según los desarrolladores del framework, esa terminología es debatible y prefieren referirse a un patrón de diseño Modelo-Plantilla-Vista (MTV o *Model-Template-View*), correspondiéndose los componentes vista y controlador del patrón MVC a los componentes plantilla y vista respectivamente (el componente modelo no varía de nombre) (41).

2.6.3 Celery

Celery es un sistema de colas de tareas distribuidas. Está implementado en Python y, aunque su protocolo puede implementarse con otros lenguajes de programación, ofrece un alto grado de compatibilidad por defecto con aplicaciones escritas en este lenguaje además de con aplicaciones implementadas con Django (42).

Las colas de tareas son un mecanismo para distribuir el trabajo entre distintos hilos o máquinas. Celery se comunica mediante mensajes, normalmente mediante el uso de un servidor de colas, para mediar entre clientes (generadores de tareas) y ejecutores (43).

2.6.4 Highcharts

Highcharts es una biblioteca escrita en JavaScript pensada para generar gráficas y diagramas de una forma sencilla (44).

Soporta gran cantidad de tipos de gráficas, incluyendo:

- Diagramas de puntos y líneas.
- Líneas de tiempo.
- Diagramas de dispersión.
- Diagrama de áreas.
- Gráficos circulares o de tarta.
- Gráficos circulares de donut.
- Gráficos de radar.
- Gráficos de embudo.
- Diagramas de barras (incluyendo diagramas de cajas y bigotes).

Para que esta biblioteca dibuje la gráfica correspondiente, debe llamarse a la función de JavaScript provista con los datos de configuración del gráfico y una matriz con los datos que deben representarse.

2.6.5 Google Chart

Google Chart es una aplicación de Google (45) para realizar estadísticas web, de fácil uso para desarrolladores de software web, usado en muchos campos como *Google Analytics*, se puede usar con diferentes formatos, *Json*, *Javascript* y *plugins* que se pueden integrar con varios lenguajes de programación.

Soporta gran cantidad de tipos de gráficas (46), incluyendo

- Diagramas de puntos y líneas.
- Líneas de tiempo.
- Diagramas de dispersión.
- Diagrama de áreas.
- Gráficos circulares o de tarta.
- Gráficos circulares de donut.
- Gráficos de radar.
- Gráficos de embudo.

2.6.1 RabbitMQ

Las colas de mensajes (del inglés *Message Queues*) son componentes que proporcionan un protocolo de comunicación asíncrona, es decir, el emisor y el receptor del mensaje no tienen que conectarse simultáneamente a la cola para que la transmisión de los mensajes sea posible.

Estos mensajes son tradicionalmente utilizados en ingeniería de *software* para intercambiar información y lanzar acciones entre distintos procesos. Este es un procedimiento muy utilizado en aplicaciones web que requieren de la ejecución de tareas pesadas en segundo plano. De esta forma, la interfaz web puede comunicarse con un motor o *backend* que ejecute dichos procesos pesados mientras continúa interactuando con el usuario.

RabbitMQ es uno de los servidores de colas de mensajes más usados. Implementa el Protocolo Avanzado de Encolado de Mensajes (AMQP o *Advanced Message Queuing Protocol*), permitiendo el encaminamiento avanzado de mensajes y asegurando la entrega de los mismos (la cola de mensajes es persistente, y se mantiene aunque se reinicie el servicio). (47).

También es posible introducir una instancia de RabbitMQ en un *cluster* de nodos RabbitMQ. Esto puede hacerse bien para asegurar la escalabilidad del sistema, o bien para sistemas de alta disponibilidad con tolerancia a caídas y errores.

2.6.2 PostgreSQL

PostgreSQL es un SGBD relacional de código abierto. Implementa características sofisticadas como control de transaccionalidad y cumple completamente con el estándar ANSI-SQL:2008 (48).

2.6.3 Docker

Docker es una tecnología de código abierto (49) que le permite crear, ejecutar, probar e implementar aplicaciones distribuidas dentro de contenedores de software. Le permite empaquetar una aplicación en una unidad estandarizada para el desarrollo de software, que contiene todo lo que necesita para funcionar: código, tiempo de ejecución, herramientas y bibliotecas del sistema, etc. Docker le permite implementar las aplicaciones de forma rápida, fiable y sistemática, en cualquier entorno.

Los contenedores son un método de virtualización del sistema operativo que le permiten ejecutar una aplicación y sus dependencias en procesos de recursos aislados. Los contenedores también le permiten empaquetar con facilidad el código de una aplicación, sus configuraciones y sus dependencias en bloques de construcción de uso sencillo que aportan uniformidad de entorno, eficacia operativa, productividad para los

desarrolladores y control de versiones. Los contenedores pueden ayudarle a garantizar la implementación rápida, consistente y de confianza de las aplicaciones independientemente del entorno. Los contenedores también aportan un control más minucioso de los recursos, lo que se traduce en una mayor eficacia de la infraestructura.

Docker reduce el tiempo empleado en configurar nuevos entornos o en solucionar los problemas asociados con el uso de entornos diferentes. Las aplicaciones compatibles con Docker pueden transferirse a la perfección de unos equipos a otros. .

Capítulo 3

Diseño

En este capítulo se describe el proceso de diseño y la metodología de trabajo seguida en el presente proyecto. Para ello, se presentan en primer los requisitos funcionales proporcionados y, a continuación, las principales decisiones técnicas y de diseño que se han tomado de cara al desarrollo de los módulos *software* descritos en el capítulo siguiente.

3.1 Metodología

Por definición, un proyecto es una planificación que consiste en el conjunto de actividades interrelacionadas y coordinadas con el fin de alcanzar unos objetivos específicos dentro de unos límites de tiempo, calidad y presupuesto. En función de esos objetivos, hay múltiples caminos para llegar a ellos por lo que puede aumentar el grado de complejidad de su desarrollo. Este hecho nos lleva a seguir los pasos definidos por el *Project Management Institute (PMI)* para un proyecto de desarrollo de *software*.

El PMI define la gestión de proyectos como la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer sus requisitos. Para que la gestión de proyectos sea satisfactoria es imprescindible partir de una planificación coherente y que permita alcanzar los objetivos del proyecto optimizando la asignación y coste de recursos.

El más famoso y reconocido producto del PMI es el *Project Management Body of Knowledge (PMBOK)*. Como su nombre sugiere, describe un conjunto de conocimientos y de prácticas aplicables a cualquier situación que se quiera formular. PMBOK provee un marco de referencia formal para desarrollar proyectos, guiando y orientando sobre la forma de avanzar en los procesos y pasos necesarios para la construcción de resultados y alcanzar los objetivos.

Por lo tanto la metodología a seguir queda descrita en la Figura 30.

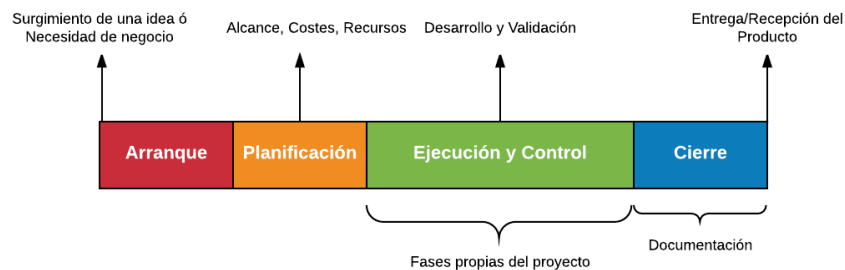


Figura 30 Gestión de proyectos usando metodología PMI

En nuestro caso particular el PFC consta de varias **subfases**, cada una de las cuales tiene **un ciclo completo** de diseño, desarrollo y pruebas de validación del software.

Por lo tanto Requisitos, Diseño, Desarrollo, Pruebas y Despliegue son actividades y no fases, donde cada fase rodea a todas las actividades y siempre debe dar como resultado **software que funciona**.

La estructuración en fases permite la división del proyecto en subconjuntos lógicos para facilitar su dirección, planificación y control.

3.2 Requisitos

Los requisitos que definen este proyecto vienen inicialmente establecidos por los antecedentes que ya fueron presentados en el capítulo 1. El objetivo de este Proyecto Fin de Carrera es validar si los resultados obtenidos en el simulador SimPrime se ajustan a los resultados medidos en campo mediante el análisis de trazas. La métrica utilizada para conseguir tal objetivo es el TTR_i , que es el tiempo que tarda el concentrador en obtener medidas de consumo de un contador después de haber realizado una solicitud a tal efecto.

Será necesario extender la funcionalidad del simulador implementado y mejorado mediante una aplicación Web, por lo que partimos de los requisitos de dicho entorno (5).

Atendiendo a la metodología descrita en el apartado previo podemos estructurar esta sección en tres fases:

- Diseño y Desarrollo Módulo *Standalone*
 - **Requisito I: Cálculo del TTR_i**

El objetivo principal de este PFC es la obtención de los parámetros necesarios para obtener el cálculo del tiempo que tarda un concentrador en obtener un informe completo de consumo (S02) desde cada uno de los contadores. Las aplicaciones webs que tenemos como objetivo mejorar están desarrolladas bajo el entorno basado en Python. Aunque técnicamente no es imprescindible, la utilización de Python para la realización del módulo standalone se toma como requisito debido a que facilita la integración del código desarrollado y su posterior mantenimiento. Esta tarea se podría llevar a cabo en otros lenguajes de programación como C ya que existen módulos que permiten la integración de aplicaciones escritas en C sobre entornos Python.

- **Requisito II: Se deberán soportar trazas de tráfico de concentradores ZIV y concentradores CIRCUTOR.**

Como hemos visto en el apartado 2.3, el formato de las trazas no es estándar, por lo que se ha decidido analizar trazas de dos de los principales fabricantes del mercado (ZIV y CIRCUTOR). El módulo desarrollado debe ser capaz de procesar los dos tipos de trazas analizados

- **Requisito III: Representación gráfica de los datos obtenidos mediante gráficos de caja y bigote.**

Uno de los métodos más representativos a la hora de comparar resultados es el método gráfico. Para el caso que nos atañe, la representación gráfica mediante un diagrama de caja y bigote aporta gran información, ya que nos aporta datos relevantes como los valores extremos calculados y el valor de la mediana.

- **Requisito IV: Obtención de estadísticos**

El módulo desarrollado va a ser utilizado para validar los resultados obtenidos en el simulador. Disponer de datos estadísticos sobre las muestras obtenidas será relevante a la hora de comprar los resultados.

- Integración del módulo *standalone* en Prime Analytics

- **Requisito I: Integración del módulo en la aplicación existente PRIME ANALYTICS.**

El módulo *standalone* desarrollado no tiene la capacidad de mantener los datos obtenidos de forma persistente, por lo tanto cada vez que sea necesario representar los mismos será necesario ejecutar el análisis de la traza completo. Para evitar este tipo casos, se tomó como requisito integrarlo en la herramienta PRIME-ANALYTICS, la cual dispone de una BBDD integrada que dará persistencia a los datos obtenidos mediante el análisis de las trazas de tráfico reales, pudiendo visualizar los resultados en cualquier momento sin necesidad de disponer de la traza analizada.

- **Requisito II: Representación gráfica y exportación de los datos obtenidos**

Los datos obtenidos mediante el análisis de trazas serán representados de forma gráfica, de tal forma que se obtenga una imagen visual de los resultados. En suma estos datos deben poder ser exportados para poder utilizarlos o procesarlos en entornos ajenos al utilizado para su obtención.

- Diseño y Desarrollo Módulo SimPrime

- **Requisito I: La interfaz web del simulador deberá proporcionar como salida no sólo el TTRAll sino también el TTRi**

Este requisito es evidente ya que de cara la validación es necesario disponer de dicha métrica para poder validar los resultados obtenidos por el simulador.

La Figura 31 ilustra el mapeo entre las entradas y salidas de la aplicación (azul), y las entradas (verde) y salidas (rojo) del simulador actual.

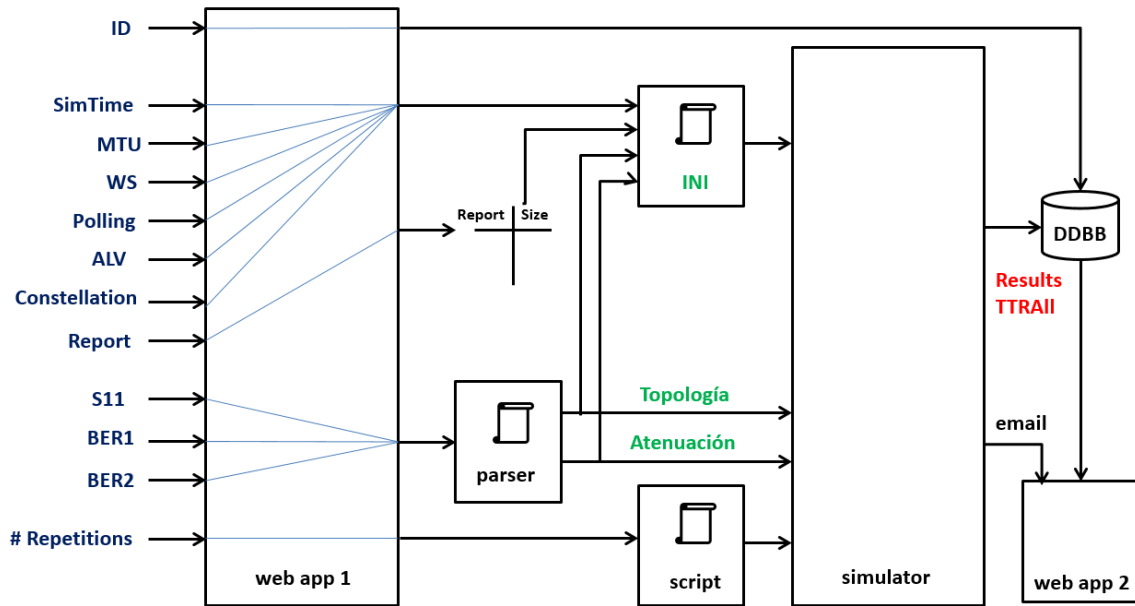


Figura 31: Entradas y salidas del sistema.

- **Requisitos II: Representación gráfica de los datos obtenidos mediante gráficos de caja y bigote.**

Los datos obtenidos mediante el análisis de trazas serán representados de forma gráfica, de tal forma que se obtenga una imagen visual de los resultados. En suma estos datos deben poder ser exportados para poder utilizarlos o procesarlos en entornos ajenos al utilizado para su obtención.

- **Requisito III: Habilitar la exportación de los datos obtenidos**

Los datos referentes al TTR_i deben poder ser exportados para poder utilizarlos o procesarlos en entornos ajenos al utilizado para su obtención.

3.3 Planificación

Como se ha visto en la sección 3.1, la metodología utilizada para el desarrollo de este proyecto se ha basado en el cumplimiento de unos requisitos y de una planificación. Esta última fue acordada con coherencia como punto de partida, definiéndose y secuenciándose las principales fases del proyecto con sus correspondientes tareas y realizándose una estimación inicial de la duración de cada una de estas fases. En el Anexo I se detalla dicha planificación indicándose la duración real de dichas fases.

3.4 Diseño

Como hemos comentado en varias ocasiones, el objetivo de este Proyecto Fin de Carrera es validar si los resultados obtenidos en el simulador SimPrime se ajustan a los resultados medidos en campo mediante el análisis de trazas. La métrica utilizada para realizar tal fin es el TTR_i , que es el tiempo que tarda el concentrador en obtener medidas de consumo de un contador después de haber realizado una solicitud a tal efecto.

Para ello, el primer paso es la generación de un módulo standalone, independiente del simulador que sea capaz de adaptarse a las trazas proporcionadas por los distintos fabricantes y calcular el TTR_i a partir de las mismas. Teniendo en cuenta el apartado 2.3, cada fabricante registra los mensajes intercambiados en un red PRIME según su propio formato. Este hecho es de vital importancia para el desarrollo de este PFC debido a que nuestro módulo debe ser capaz de procesar los diferentes formatos proporcionados y obtener los datos objetivo de forma uniforme.

Decisión I. Procesado del formato de traza mediante expresiones regulares.

Debido a que el registro de trazas de tráfico de cada fabricante no es estandar es necesario procesar las trazas de forma independiente. Para tal fin hacemos uso de expresiones regulares que soporten cada uno de los formatos por separado pero que sean capaces de extraer los mismos datos. El uso de expresiones regulares permite machear cada traza de un archivo de trazas de forma independiente para su posterior procesado.

Decisión II. Se procesan trazas de tipo control, REG y PRO

Para el cálculo del TTR_i , únicamente se necesitan trazas de tipo *DATA*. Se ha extendido la funcionalidad de procesado a trazas de tipo *CONTROL* para poder procesar trazas de *REG* y *PRO* específicas que nos aportan información sobre la red *PRIME*.

PRO control packet (PKT.CTYPE = 3) → PRO_REQ_S

REG control packet (PKT.CTYPE=1) → REG_RSP.

El procesamiento de este tipo de trazas nos permite obtener la relación entre la dirección física de los nodos y su identificador lógico *SID*.

- *PNA*: *MAC address* de un nodo que realiza las funciones de switch. Cuando un nodo no puede alcanzar a cualquier otro switch, puede enviar tramas de promoción (promotion-needed) de modo que un terminal puede ser promovido y comenzar a ser switch.

Si queremos identificar que nodos son promovidos a Switch, la relación directa entre *NID* y *MAC* se encuentra en los mensajes

- EUI-48: Dirección *MAC* de cada nodo, definida en IEEE 802-2001. Si queremos identificar cada NID en la red PRIME con su correspondiente dirección física debemos analizar los tipos de paquetes identificados como **REG_RSP**.

Decisión III. Elección del eje de tiempos

Como ya hemos explicado en la la sección 2.3 existen dos parámetros que nos pueden servir para el cálculo del eje de tiempos. Uno de ellos hace referencia al *pc_time* del equipo mientras que el otro es un tiempo interno del concentrador, *time*. Se ha tomado como parámetro relevante el segundo debido a su facilidad de procesamiento.

Decisión IV. El almacenamiento de los datos obtenidos se va a realizar en formato estándar, independientemente del fabricante, mediante el uso de bases de datos.

Debido a que el registro de de trazas de trafico de cada fabricante no es estandar es necesario establecer las unidades de medida que queremos utilizar. Se toma como referencia el modelado de datos utilizado por CIRCUTOR. De esta forma se establece las siguientes características:

- Tiempo de registro: Las unidades de tiempo utilizadas para representar el *TTRi* serán segundos. Por lo tanto será necesario transformar dicho parámetro en CIRCUTOR, ya que dicho fabricante representa el tiempo en milisegundos.
- SID: Se recuperará y se almacenara como entero. Por lo tanto será necesario transformar el valor hexadecimal obtenido del correspondiente campo registrado en CIRCUTOR
- LNID: Se recuperará y se almacenara como entero. Por lo tanto será necesario transformar el valor hexadecimal obtenido del correspondiente campo registrado en CIRCUTOR
- DATOS: Corresponde al valor hexadecimal de cada traza PRIME. Debido a que las trazas de CIRCUTOR no registran la traza completa en formato hexadecimal se decide almacenar unicamente el fragmento de mensaje que corresponde a los datos de la capa de aplicación, *packet payload*, por lo que se omitaran todos las cabeceras PRIME y campo CRC existente en ZIV

Ademas de estos datos, necesarios para dar robusted a nuestro módulo el procesamiento de las trazas reales se extraen y almacenan otros datos importantes para analizar redes PRIME.

- *specific_type*: Tipo de traza PRIME
- *htype*: tipo de paquete
- SCHEME: Este campo se mapea como BPSK-FEC.
- *Phylevel*: El parámetro Level especifica el nivel de señal en el que la capa PHY recibió la PSDU. Puede tomar uno de dieciséis valores: 1-15
CIRCUTOR registra este parámetro mediante los valores 1-8, sin embargo ZIV, lo registra en db. Ante esta situación se decide registrar este valor como

valores comprendidos entre 1-15, por lo que será necesario realizar las transformaciones necesarios en el preprocesado de ZIV.

- Physnr: El parámetro SNR puede tomar valores comprendidos entre 0 y 7. CIRCUTOR registra este parámetro mediante los valores 0-7, sin embargo ZIV, lo registra en dBs. Ante esta situación se decide registrar este valor como valores comprendidos entre 0-7, por lo que será necesario realizar las transformaciones necesarios en el preprocesado de ZIV.

Las Figuras 32 y 33 se ilustran las transformaciones de datos descritas para conseguir un formato de datos estandar.

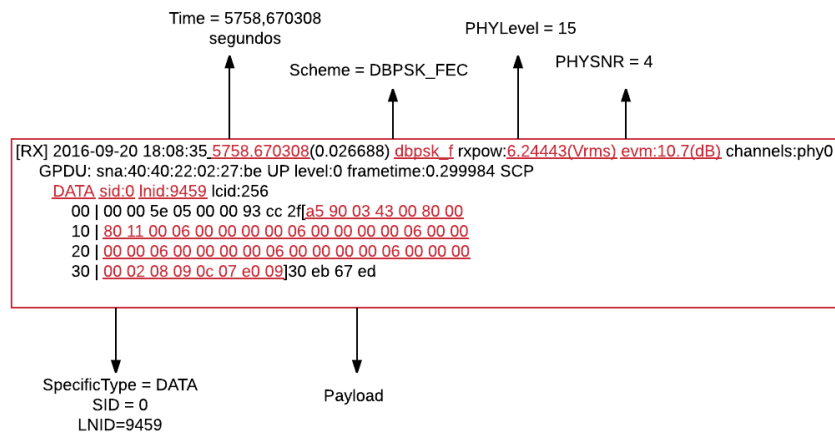


Figura 32 Extracción Datos ZIV en formato estándar

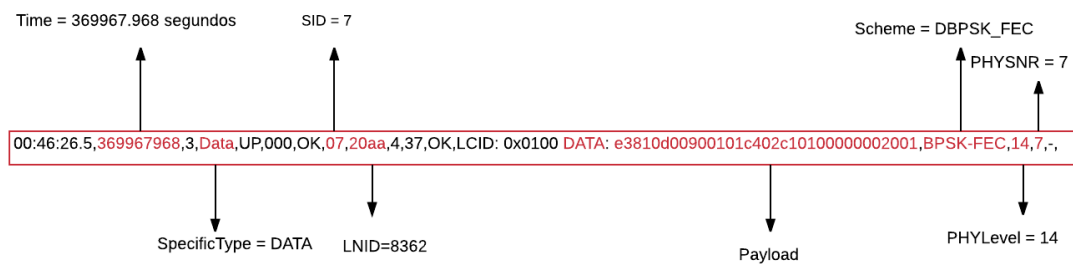


Figura 33 Extracción Datos CIRCUTOR en formato estándar

Decisión V. Inclusión de un mecanismo de control para verificar el correcto procesamiento de las trazas analizadas.

Ha sido necesario crear un módulo adicional de verificación de traza ya que se ha detectado que en ocasiones una traza fuera de secuencia pueda provocar el calculo erróneo del TTRi, por lo tanto es necesario tener el cuenta el número de secuencia de cada fragmento DLMS/COSEM en los que se compone el envío de un informe de consumo solicitado desde el concentrador. De esta forma si llega una traza correspondiente a un bloque fuera de secuencia no será tomada en cuenta.

Decisión VI. Error en el calculo del TTRi (trazas fragmentadas)

Como ya hemos indicado en la sección 2.3, los datos transferidos desde el nivel de aplicación son fragmentados si superan el tamaño de MTU establecido. En la Figura 34 mostramos la recepción del último bloque correspondiente a la solicitud de un informe S02. El tiempo que marcamos como fin de envío del informe es el correspondiente a la traza que tiene el patrón marcado en verde. Sin embargo como ya hemos detallado este no sería el fin del informe completo, ya que el último paquete correspondiente al último bloque se da un tiempo después. El número de trazas que comprenden el bloque depende del WS establecido por el fabricante por lo que a efectos prácticos se toma el tiempo de finalización como el correspondiente a la recepción de la traza que contiene el patrón LAST BLOCK = TRUE.

```
[RX] 2016-09-20 18:14:57 6140.850096(0.026688) dbpsk_f rxpow:6.27722(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.405860 SCP
DATA sid:0 Inid:9970 lcid:256
00 | 00 00 5e 05 00 00 9b c8 2e[99 90 07 04 90 02 01
10 | c4 02 c1 01 00 00 00 05 00 81 c0 02 08 09 0c 07
20 | e0 09 14 02 0f 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00]f5 e5 61 03
[RX] 2016-09-20 18:14:57 6140.900016(0.026688) dbpsk_f rxpow:6.40579(Vrms) evm:11.1(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.455780 SCP
DATA sid:0 Inid:9970 lcid:256
00 | 00 00 5e 05 00 00 9b c8 2e[9a 90 07 40 06 00 00
10 | 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00 00
20 | 00 02 08 09 0c 07 e0 09 14 02 10 00 00 00 80 00
30 | 80 11 00 06 00 00 00]12 69 4c 57
[RX] 2016-09-20 18:14:57 6140.938160(0.026688) dbpsk_f rxpow:6.37579(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.493924 SCP
DATA sid:0 Inid:9970 lcid:256
00 | 00 00 5e 05 00 00 9b c8 2e[9b 90 07 41 00 06 00
10 | 00 00 00 06 00 00 00 06 00 00 00 00 06 00 00 00
20 | 00 00 06 00 00 00 00 02 08 09 0c 07 e0 09 14 02
30 | 11 00 00 00 80 00 80]a1 02 87 ae
[RX] 2016-09-20 18:14:57 6140.971456(0.026688) dbpsk_f rxpow:6.33978(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.527220 SCP
DATA sid:0 Inid:9970 lcid:256
00 | 00 00 5e 05 00 00 9b c8 2e[9c 90 07 42 11 00 06
10 | 00 00 00 06 00 00 00 06 00 00 00 00 06 00 00 00
20 | 00 00 00 06 00 00 00 06 00 00 00 00 02 08 09
30 | 0c 07 e0 09 14 02 12]ab 13 f1 d5
[RX] 2016-09-20 18:14:57 6141.009740(0.026688) dbpsk_f rxpow:6.22023(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.565504 SCP
DATA sid:0 Inid:9970 lcid:256
00 | 00 00 5e 05 00 00 9b c8 2a[dd 90 07 83 00 00 00
10 | 80 00 80 11 00 06 00 00 00 00 06 00 00 00 00 06
20 | 00 00 00 00 06 00 00 00 06 00 00 00 00 06 00
30 | 00 00 00]17 7f e2 04
```

Figura 34 Ejemplo de trazas PRIME correspondiente a un bloque procedente de la capa de aplicación

Decisión VII. Identificado de nodo en el Simulador

El simulador no registra el par SID/LNID para la obtención del identificador único de cada nodo en la red por lo que se toma como referencia del nodo simulado para representar los datos obtenidos un identificador numérico incremental que corresponde con la columna en la que se encuentra el dato $TTRi$ simulado para dicho nodo.

Capítulo 4

Desarrollo

El objetivo de este capítulo es presentar los aspectos más técnicos del desarrollo, tomando como referencia el diseño presentado en el capítulo anterior. Así, en este capítulo se detalla la implementación de cada uno de los módulos en el orden en que fueron desarrollados para permitir comprender el proceso de forma clara y sencilla

4.1 Análisis de Trazas

Partiendo de las indicaciones presentadas en el capítulo anterior, nos disponemos a detallar cada uno de los pasos seguidos durante el desarrollo para la generación del módulo *standalone* y la extensión de la aplicación web del simulador para que muestre los resultados referentes al *TTRi* obtenidos en las simulaciones.

4.1.1 Módulo standalone para el cálculo del *TTRi*

El módulo desarrollado en este PFC es capaz de procesar la gran mayoría de los tipos de paquetes que utiliza PRIME para transferir la información. A pesar de ello la finalidad de este PFC no es procesar todos ellos si no unos paquetes concreto.

Para el análisis de trazas y obtención del *TTRi* se ha tenido en cuenta que la solicitud y recepción de información por parte del concentrador es secuencia, es decir, ante una solicitud de informe de consumo S02 por parte del concentrador a un contador concreto identificado por su NID (SID/LNID) ningún otro nodo puede transmitir al mismo tiempo.

Por otra parte, según lo desarrollado en el apartado 2.3, el formato de las trazas depende del fabricante, por lo tanto para hacerlo de forma óptima, es necesario hacer un "preprocesado" de la traza, dependiente del fabricante, que nos permita almacenar la información de la traza de manera uniforme/homogénea en base de datos o en RAM, para luego realizar el cálculo del *TTRi* a partir de dichos datos.

La Figura 35 muestra el flujo completo que realizará el módulo desarrollado.

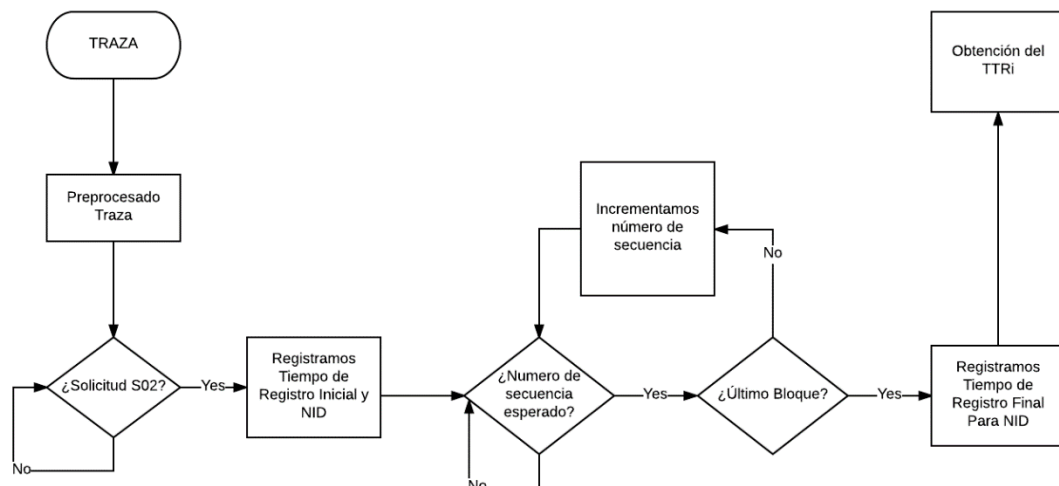


Figura 35 Explicación del proceso de extracción.

En la Figura 35 hacemos referencia a Preprocesado de la traza. Es en este punto donde la elección del fabricante es relevante. Una vez pre procesada la traza de acuerdo a los valores deseados el cálculo del *TTRi* es equivalente e independiente del fabricante.

El módulo desarrollado mantiene la relación entre el NID (identificador univoco del nodo) y el TTR_i obtenido para dicho nodo como muestra la Figura 36.

```
python TTRi.py
Has pulsado la opción 2...
pulsar una tecla para continuar
Archivo: circutor.csv
[["SID-LNID", "TTRi_time"], [{"109196",16.683}, {"28508",15.54}, {"08535",9.01199999999}, {"28343",23.791}, {"78586",12.381}, {"89019",14.269}, {"78615",14.091}, {"89137",11.79}, {"08575",9.14400000003}, {"109197",17.997}, {"78312",14.954}, {"78366",10.428}, {"08606",8.39399999997}, {"119187",13.771}, {"89181",11.608}, {"09157",10.086}, {"09026",9.21999999997}, {"178685",15.056}, {"08659",9.28000000003}, {"08282",8.63900000002}, {"88741",14.57}, {"89064",12.719}, {"78570",13.062}, {"79142",14.317}, {"48458",22.271}, {"78849",16.032}, {"48304",19.501}, {"178920",12.612}, {"178457",10.614}, {"08219",8.77500000002}, {"59010",13.072}, {"08287",8.47999999998}, {"18706",16.249}, {"69106",20.158}, {"08442",8.66899999999}, {"83996",12.269}, {"08275",8.83500000002}, {"58500",11.195}, {"08572",8.53900000005}, {"09014",8.20499999996}, {"68595",15.152}, {"68222",16.402}, {"58968",16.854}, {"58348",11.54}, {"89154",13.974}, {"68756",17.6200000001}, {"68823",16.162}, {"88725",16.255}, {"8293",10.759}, {"68836",20.33}, {"108303",16.935}, {"88352",14.9449999999}, {"178965",14.274}, {"48444",19.138}, {"78362",12.63}, {"28688",17.259}, {"08837",9.63099999999}, {"178900",12.47}, {"08775",8.57899999997}, {"08355",26.448}, {"48957",11.88}, {"88980",5.63099999999}], [{"5.630999999993946, 10.086000000010245, 13.061999999976251, 16.205500000010943, 26.447999999974854}]]
pulsar una tecla para continuar

*****Estadisticos*****
Numero de muestras 62
Media 13.5668709677
Varianza 17.7151732618
sigma 4.20893968379
Mediana 13.062
IC Parametrico [13.0068429098,14.1268990257]
IC No Parametrico [8.39399999997,23.791]
*****Datos Grafica Circutor*****
[5.630999999993946, 10.086000000010245, 13.061999999976251, 16.205500000010943, 26.447999999974854]
pulsar una tecla para continuar
```

Figura 36 Salida Shell del módulo standalone

Cada TTR_i obtenido será almacenado en memoria para ser procesado posteriormente, obteniendo tanto los estadísticos necesarios del análisis de las muestras como su representación gráfica. Ambos aspectos quedan reflejados en las Figuras 37 y 38.

```
python TTRi.py
Has pulsado la opción 2...
pulsar una tecla para continuar
Archivo: circutor.csv
[["SID-LNID", "TTRi_time"], [{"109196",16.683}, {"28508",15.54}, {"08535",9.01199999999}, {"28343",23.791}, {"78586",12.381}, {"89019",14.269}, {"78615",14.091}, {"89137",11.79}, {"08575",9.14400000003}, {"109197",17.997}, {"78312",14.954}, {"78366",10.428}, {"08606",8.39399999997}, {"119187",13.771}, {"89181",11.608}, {"09157",10.086}, {"09026",9.21999999997}, {"178685",15.056}, {"08659",9.28000000003}, {"08282",8.63900000002}, {"88741",14.57}, {"89064",12.719}, {"78570",13.062}, {"79142",14.317}, {"48458",22.271}, {"78849",16.032}, {"48304",19.501}, {"178920",12.612}, {"178457",10.614}, {"08219",8.77500000002}, {"59010",13.072}, {"08287",8.47999999998}, {"18706",16.249}, {"69106",20.158}, {"08442",8.66899999999}, {"83996",12.269}, {"08275",8.83500000002}, {"58500",11.195}, {"08572",8.53900000005}, {"09014",8.20499999996}, {"68595",15.152}, {"68222",16.402}, {"58968",16.854}, {"58348",11.54}, {"89154",13.974}, {"68756",17.6200000001}, {"68823",16.162}, {"88725",16.255}, {"8293",10.759}, {"68836",20.33}, {"108303",16.935}, {"88352",14.9449999999}, {"178965",14.274}, {"48444",19.138}, {"78362",12.63}, {"28688",17.259}, {"08837",9.63099999999}, {"178900",12.47}, {"08775",8.57899999997}, {"08355",26.448}, {"48957",11.88}, {"88980",5.63099999999}], [{"5.630999999993946, 10.086000000010245, 13.061999999976251, 16.205500000010943, 26.447999999974854}]]
pulsar una tecla para continuar

*****Estadisticos*****
Numero de muestras 62
Media 13.5668709677
Varianza 17.7151732618
sigma 4.20893968379
Mediana 13.062
IC Parametrico [13.0068429098,14.1268990257]
IC No Parametrico [8.39399999997,23.791]
*****Datos Grafica Circutor*****
[5.630999999993946, 10.086000000010245, 13.061999999976251, 16.205500000010943, 26.447999999974854]
pulsar una tecla para continuar
```

Figura 37 Obtención de estadísticos relativos al TTR_i

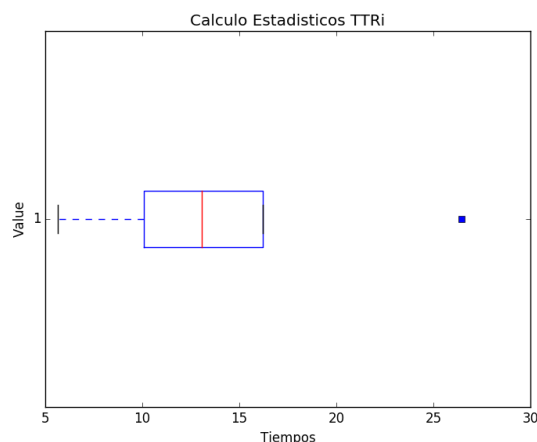


Figura 38 Diagrama de caja y bigote obtenido mediante el módulo standalone.

4.1.2 Integración del módulo standalone para el cálculo del TTRi en la herramienta PRIME Analytics

Prime Analytics es una aplicación web que permite detectar problemas de comunicaciones en redes PRIME en base al análisis de trazas de tráfico y otra información recopilada en campo (8), (9).

Integrar la funcionalidad desarrollada en la sección previa permite ampliar el alcance de la herramienta y otorga la posibilidad de utilizar su desarrollo previo para poder mantener los resultados referentes al *TTRi* calculados mediante el módulo *standalone* de manera persistente mediante la utilización de su base de datos.

El siguiente obstáculo al que nos enfrentamos es a la adquisición de conocimiento de dicha herramienta. El objetivo es homogenizar el formato de traza para que el cálculo del *TTRi* sea independiente del fabricante

Con el fin de aprovechar todo el desarrollo existente y mantener los datos ya simulados, para la incorporación del nuevo módulo se hace imprescindible no modificar la base de datos. Por lo que los parámetros extraídos de cada traza PRIME tienen que corresponderse con los utilizados en la declaración de la estructura de base de datos

El flujo de trabajo que experimenta la herramienta para la obtención del *TTRi* queda reflejado en la Figura 39.

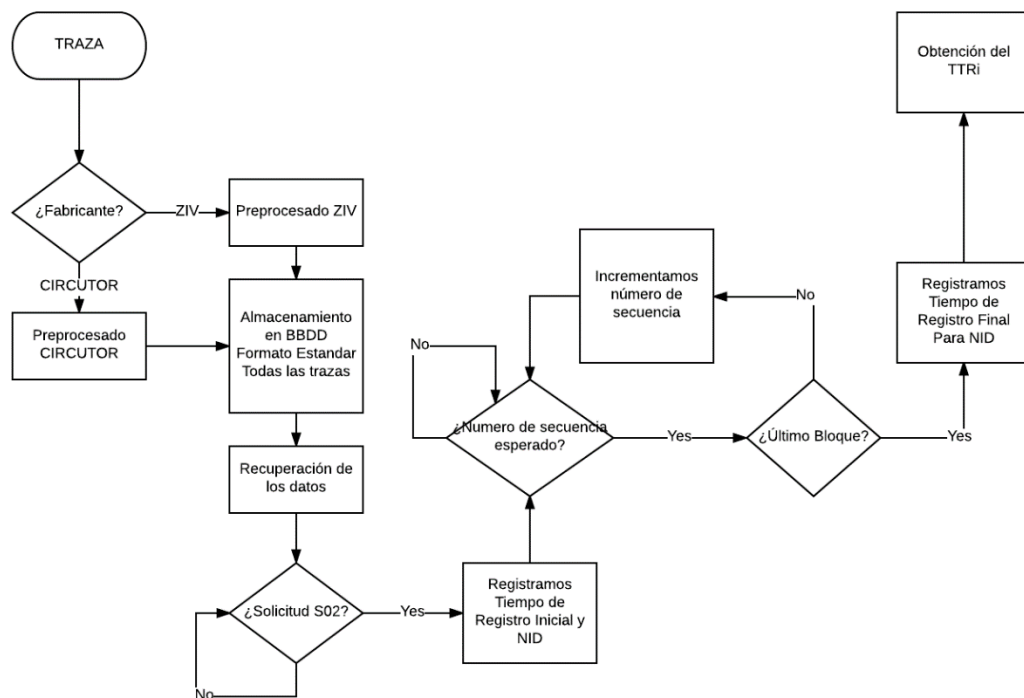


Figura 39 Diagrama de flujo PRIME ANALYTICS

La Figura 39 muestra la necesidad de crear un módulo de pre-procesado previo que dependerá del fabricante y que como salida otorgará un formato estándar para cada traza analizada.

Es en esta herramienta donde cobra sentido el haber parcheado trazas de tipo **PRO_REQ_S** y **REG_RSP**, ya que aunque la finalidad de este PFC no sea obtener estos datos, tenerlos registrados en la base de datos puede ser de utilidad para futuros desarrollos.

Podemos obtener la relación entre dirección MAC física del dispositivo y su identificador lógico NID mediante consultas a la BBDD.

Del mismo modo podemos obtener las direcciones MAC de los dispositivos de la red PRIME que actúan como *Switch* mediante la obtención del campo PNA de la base de datos.

Los resultados obtenidos quedan representados en la Figura 40.

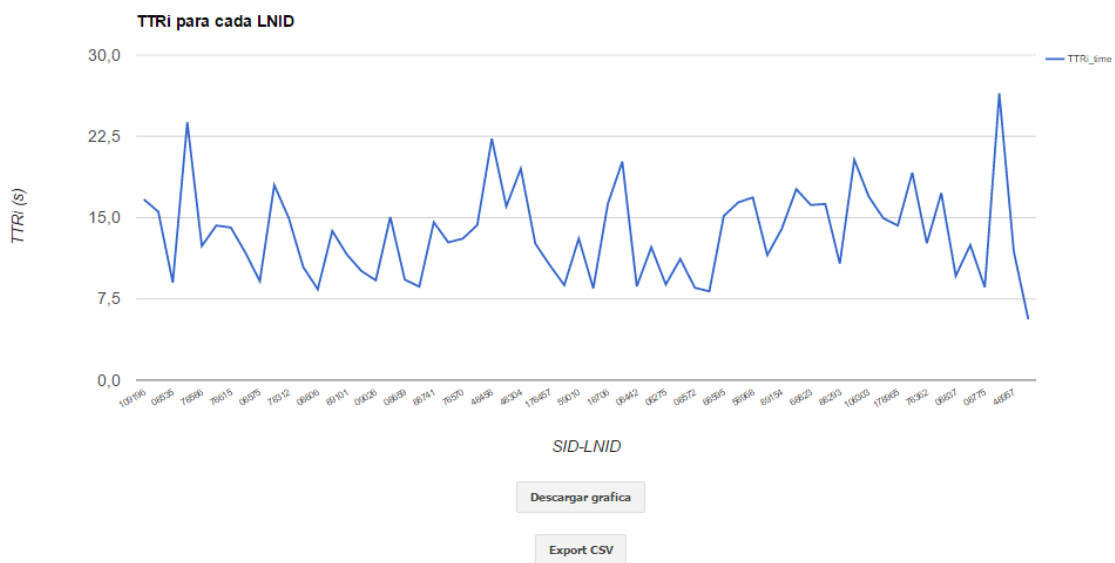


Figura 40 Salida Herramienta PRIME ANALYTICS para la obtención del TTRi

Los datos representados pueden ser exportados en formato csv, para su posterior procesado. La Figura 41 muestra el formato utilizado.

```

1 SID-LNID,TTRi_time
2 109196,16.683000000019092
3 28508,15.539999999979045
4 08535,9.011999999987893
5 28343,23.790999999968335
6 78586,12.380999999993946
7 89019,14.269000000029337
8 78615,14.091000000014901
9 89137,11.789999999979045
10 08575,9.144000000029337
11 109197,17.996999999973923
12 78312,14.9539999999688
13 78366,10.428000000014435
14 08606,8.393999999971129
    
```

Figura 41 Formato Datos TTRi exportados de PRIME ANALYTICS

De esta manera los datos ya quedan almacenado en la BBDD, por lo que lo siguiente a lo que nos enfrentamos es poder obtener los mismos. Para poder realizarlo es necesario modificar la parte **frontend**

Es necesario modificar el módulo correspondiente a la recuperación de los datos debido a que no está implementado. Será necesario añadir una nueva funcionalidad, que además de los datos que ya obtenía, pueda obtener los nuevos datos almacenados en BBDD y correspondientes al $TTRi$.

Una vez que disponemos de los datos únicamente nos queda la representación de los mismos, para ello usaremos un diagrama de caja (*box-and-whisker plot*)

Para representar los datos obtenidos mostraremos dos gráficas

- $TTRi$ único, calculado mediante el procesado de todos los $TTRi$ calculados para todos los nodos. La Figura 43 ilustra la representación gráfica de los resultados para el $TTRi$ global.

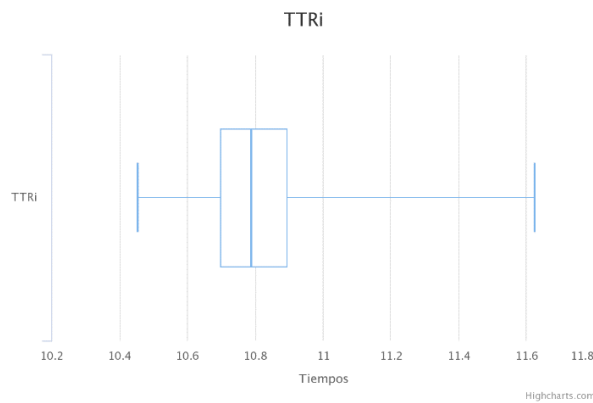


Figura 43 Obtención del $TTRi$ global en el Simulador

- $TTRi$ multinodo, Tantos diagramas de cajas como nodos de servicio (= contadores) haya en la red. Se representaría identificador de nodo (eje y) frente a TTR /diagrama caja (eje x). La Figura 44 ilustra la representación gráfica de los resultados para el $TTRi$ por nodo.

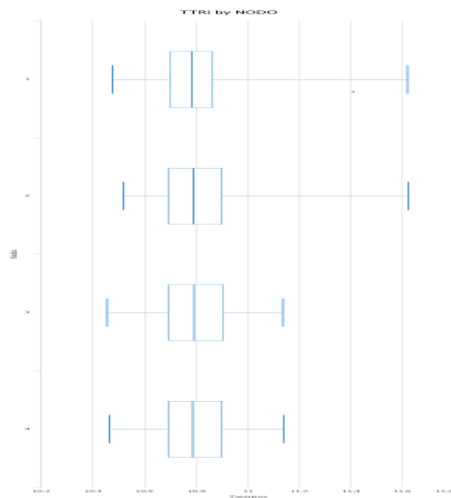


Figura 44 Obtención del $TTRi$ por nodo en el simulador

En suma, a la representación gráfica se añade la funcionalidad de poder exportar los datos para otorgar al usuario la posibilidad de analizarlos.

El flujo de trabajo para el cálculo del $TTRi$ con los datos obtenidos del simulador sería el que se representa en la Figura 45.

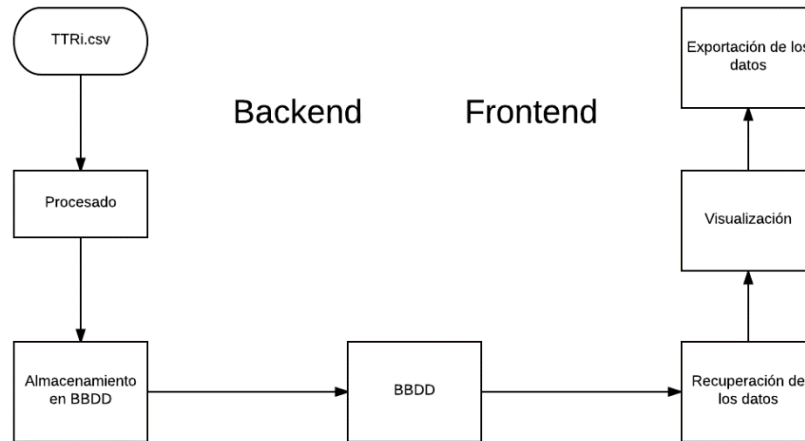


Figura 45. Flujo de trabajo para extender la funcionalidad del Simulador

Capítulo 5

Validación

Este capítulo detalla las diferentes validaciones realizadas para comprobar el correcto funcionamiento de los módulos *software* desarrollados, así como una comparativa de los valores obtenidos por el módulo de análisis de trazas y por el simulador, con el fin de determinar la validez de este último. En primer lugar, se realiza la validación del módulo *software* desarrollado para calcular el *TTRi* mediante el procesamiento de trazas de tráfico de redes PRIME reales. Posteriormente se comparan los valores obtenidos mediante simulaciones con los valores medidos en un caso real, comprobándose que encajan, lo que supone la principal conclusión de este trabajo. Por último, se analiza cómo varía la salida del simulador al cambiar ciertos parámetros de comunicaciones.

5.1 Validación del módulo *standalone* para el cálculo del *TTRi*

El objetivo que se plantea en esta sección es validar que los *TTRi* que calcula el módulo *software standalone* desarrollado son correctos. La metodología a seguir será comparar los resultados del *script* con la inspección manual de trazas de tráfico reales. Los parámetros en base a los que se va a realizar dicha validación son:

- Número total de *TTRi* en la traza
- *TTRi*
- Casos singulares (para comprobar que funciona correctamente ante anomalías en el intercambio de mensajes).

Como hemos visto en los apartados previos el formato de traza analizado no es estándar, sino que depende del fabricante, y el alcance de este PFC contempla soportar dos de los principales fabricantes del mercado (ZIV y CIRCUTOR). Por lo tanto, debemos validar que el módulo *standalone* para el cálculo del *TTRi* funciona bien en ambos casos.

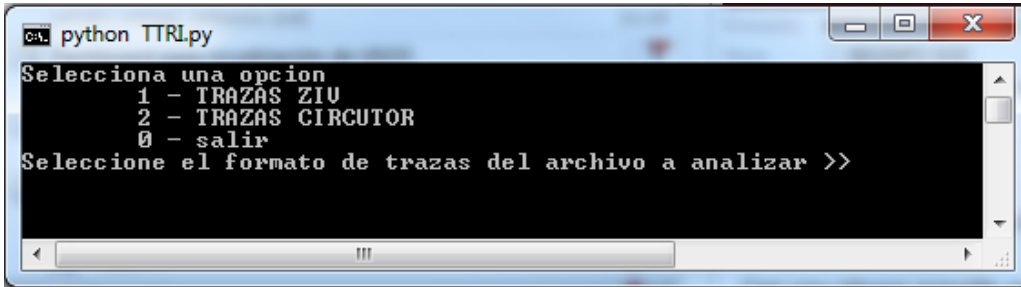
Para ello disponemos de dos archivos de trazas reales, uno de ZIV (panel de telemedida), que consta de 4 contadores, y otro de CIRCUTOR (obtenido de un entorno residencial real), que consta de 65 contadores. Ambos archivos registran una solicitud de informe S02 por contador, por lo que nuestro *script* debe devolvernos tantos valores *TTRi* calculados como contadores tenga la red PRIME analizada.

En el caso de ZIV vamos a comprobar los 3 puntos mencionados anteriormente (*TTRi*, número total de medidas y casos singulares). En el caso de CIRCUTOR, también, pero, debido al elevado número de medidas, el *TTRi* sólo lo comprobaremos para una muestra aleatoria de la traza.

En suma a lo anterior, se va a aprovechar la inspección manual de la traza para obtener parámetros relevantes para el simulador (MTU, WS) y para obtener el *TTRi* ideal.

El primer paso es analizar las trazas procedentes de una red PRIME con concentrador del fabricante ZIV. Conocemos de antemano que la red analizada está formada por 4 contadores inteligentes, por lo que el primer punto de análisis es verificar que obtenemos 4 medias. En las Figuras 46, 47 y 48 vemos la ejecución del módulo software desarrollado.

La Figura 46 muestra la ejecución del módulo *standalone*. Lo primero que es necesario indicar es el fabricante de los dispositivos que configuran la red PRIME.



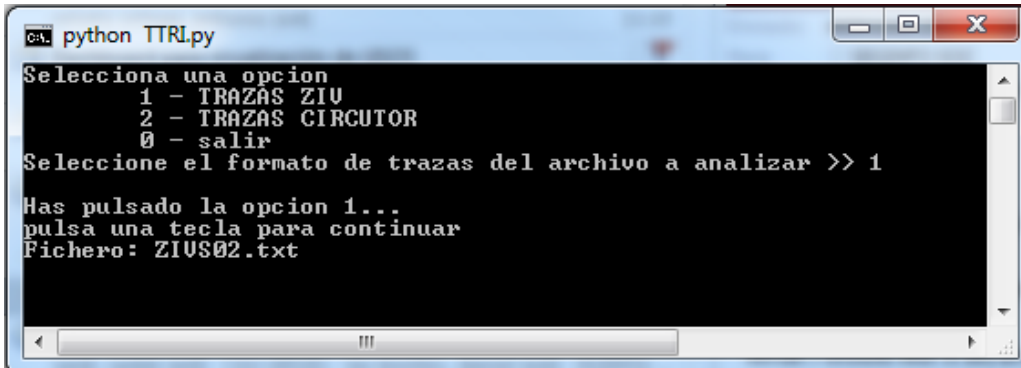
```

python TTRI.py
Selecciona una opcion
  1 - TRAZAS ZIU
  2 - TRAZAS CIRCUTOR
  0 - salir
Seleccione el formato de trazas del archivo a analizar >>

```

Figura 46 Ejecución Módulo Standalone para el análisis de trazas reales

La Figura 47 muestra la solicitud del archivo de trazas reales a analizar.



```

python TTRI.py
Selecciona una opcion
  1 - TRAZAS ZIU
  2 - TRAZAS CIRCUTOR
  0 - salir
Seleccione el formato de trazas del archivo a analizar >> 1
Has pulsado la opcion 1...
pulsas una tecla para continuar
Fichero: ZIUS02.txt

```

Figura 47 Elección del archivo de trazas reales en el módulo standalone desarrollado

La Figura 48 muestra por pantalla los resultados obtenidos del procesado de las trazas reales facilitadas en pares de SID/LNID (identificador único del nodo en la red PRIME) y el tiempo necesario para enviar el informe S02 de consumo por parte del nodo al recibir una solicitud a tal efecto.

```

ca: python TTRi.py
Selecciona una opcion
  1 - TRAZAS ZIV
  2 - TRAZAS CIRCUTOR
  0 - salir
Seleccione el formato de trazas del archivo a analizar >> 1

Has pulsado la opcion 1...
pulsar una tecla para continuar
Archivo: ZIVS02.txt
["SID-LNID", "TTRi_time"], [{"09459",3.793684}, {"09970",2.68378}, {"09567",2.805216}, {"08717",2.682328}]
*****
Numero de muestras 4
Media 2.991252
Varianza 0.28949319216
sigma 0.538045715678
Mediana 2.68378
IC Parametrico [2.84940033584,3.13310366416]
IC No Parametrico [2.68378,3.793684]
*****Datos Grafica Ziv*****
[2.68232799999998704, 2.68232799999998704, 2.6837800000000297, 3.2994499999999524, 3.79368399999993026]

```

Figura 48 Salida Módulo standalone para ZIV

De la Figura 48 podemos apreciar que se cumple el primer punto de validación: se han obtenido 4 muestras, las esperadas para el análisis de la red PRIME analizada. Por otro lado, cabe destacar que los valores obtenidos están muy próximos.

Continuando con la metodología descrita, y teniendo en cuenta que la red analizada es pequeña, podemos comparar los resultados del script con la inspección manual de las trazas de tráfico reales. Para ello es necesario tener en cuenta lo explicado en la sección 2.3. El *TTRi* será el resultado de calcular la diferencia entre el envío de una traza cuyo *Packet Payload* contenga la cadena hexadecimal “**c001c100070100630100ff**” (REQ S02 en la Figura 49) y la recepción de una traza cuyo *Packet Payload* contenga la cadena hexadecimal “**c402c101**” (RSP LAST BLOCK TRUE en la Figura 49) teniendo en cuenta los mecanismos implementados de control de traza adecuada.

La Figura 49 muestra el intercambio de mensaje existente entre CNTR y contador para el caso particular de ZIV y obtenido mediante el análisis de las trazas. Como vimos en el apartado 2.3, el nivel de aplicación, protocolo DLMS/COSEM, está basado en un sistema cliente-servidor, por lo que queda caracterizado por los siguientes tipos de mensajes:

- **Concentrador - Request**
 - Service.request → **c001c1**
 - Service.requestS02 → **c001c100070100630100ff**
 - Service.request.NextBlock → **c002c1**
- **Contador Inteligente - Response**
 - Service.response-lastblock_false → **c402c100** (los cuatro bytes siguientes indican el número de secuencia correspondiente)
 - Service.response-lastblock_true → **c402c101** (los cuatro bytes siguientes indican el número de secuencia correspondiente)

5.1 VALIDACIÓN DEL MÓDULO STANDALONE PARA EL CÁLCULO DEL TTRI

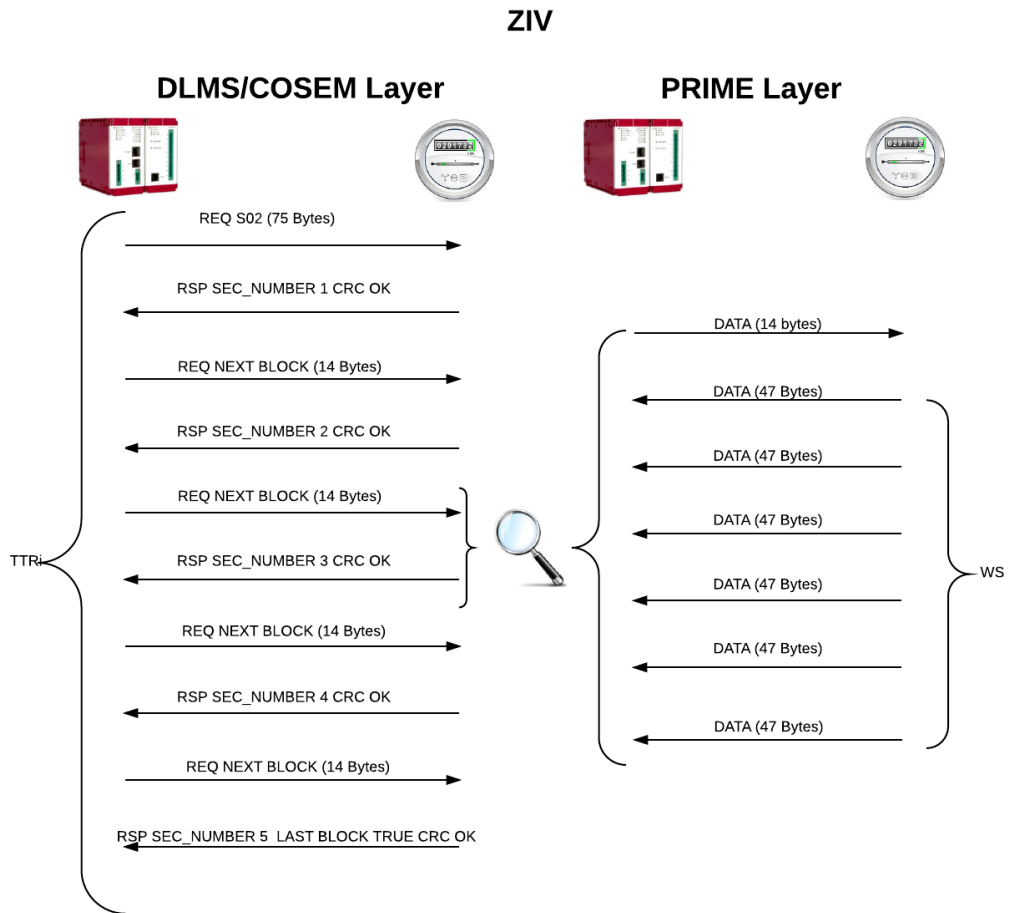


Figura 49 Intercambio de mensajes entre Concentrador y contador (ZIV)

El formato de cada traza en ZIV queda representado en la Figura 50.

```

-[RX] 2016-09-20 18:17:47 6311.488212(0.024448) dbpsk_f rxpow:6.61275(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.409024 SCP
DATA sid:0 Inid:9567 lcid:256
00 | 00 40 99 05 00 00 95 7c 26[81 90 01 01 90 01 02
10 | c0 01 c1 00 07 01 00 63 01 00 ff 02 01 01 02 04
20 | 02 04 12 00 08 09 06 00 00 01 00 00 ff 0f 02]d7
30 | 03 4b 27
  
```

MAC HEADER
PACKET HEADER
PAYLOAD
CRC

Figura 50 Formato trazas ZIV

Dada el número elevado de trazas registradas, en las siguientes figuras ilustramos en intercambio de mensajes realizado teniendo en cuenta la fragmentación originada en la capa de aplicación. La Figura 51 muestra la solicitud de S02 realizada por el CNTR al contador. En este caso la solicitud se ve fragmentada en dos trazas PRIME ya que los datos a transmitir superan el tamaño de MTU establecido.

```
[RX] 2016-09-20 18:17:47 6311.488212(0.024448) dbpsk_f rxpow:6.61275(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.409024 SCP
DATA sid:0 Inid:9567 lcid:256
00 | 00 40 99 05 00 00 95 7c 26[81 90 01 01 90 01 02
10 | c0 01 c1 00 07 01 00 63 01 00 ff 02 01 01 02 04
20 | 02 04 12 00 08 09 06 00 00 01 00 00 ff 0f 02]d7
30 | 03 4b 27
[RX] 2016-09-20 18:17:47 6311.519220(0.024448) dbpsk_f rxpow:6.72741(Vrms) evm:14.4(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.440032 SCP
DATA sid:0 Inid:9567 lcid:256
00 | 00 40 99 05 00 00 95 7c 25[c2 90 01 80 12 00 00
10 | 09 0c 07 e0 09 13 01 12 15 32 00 80 00 80 09 0c
20 | 07 e0 09 14 02 12 15 32 00 80 00 80 01 00]bd 91
30 | fc d6
[RX] 2016-09-20 18:17:48 6311.758360(0.011008) dbpsk_f rxpow:6.35537(Vrms) evm:10.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.060924 SCP
DATA sid:0 Inid:9567 lcid:256
00 | 00 00 5e 05 00 00 95 7c 03[81 90 03]5a 7f ea cf
```

Figura 51 Solicitud informe S02 ZIV

La Figura 52 muestra el envío por parte del contador de un bloque, identificado con número de secuencia 1 correspondiente al informe S02 previamente solicitado. Dicho bloque es fragmentado en varias trazas PRIME como vimos en la Figura 49. Una vez confirmada la entrega completa del bloque, el CNTR envía una solicitud de envío de un nuevo bloque al mismo tiempo que confirma la recepción del bloque anterior, tal y como se ilustra en la Figura 52 con el código c002c1 (NEXT BLOCK).

El intercambio de mensajes ilustrado en la Figura 52 se repite un total de 4 veces. El último bloque correspondiente a la transmisión del informe S02 completo se ilustra en la Figura 53.

```
[RX] 2016-09-20 18:17:48 6311.983496(0.026688) dbpsk_f rxpow:6.52766(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.286060 SCP
DATA sid:0 Inid:9567 Lcid:256
00 | 00 00 5e 05 00 00 95 7c 2f[81 90 03 05 90 02 01
10 | c4 02 c1 00 00 00 01 00 81 f1 01 18 02 08 09
20 | 0c 07 e0 09 13 01 13 00 00 00 80 00 80 11 00 06
30 | 00 00 00 00 06 00 00 00]ee 2c b5 c8
[RX] 2016-09-20 18:17:48 6312.018712(0.026688) dbpsk_f rxpow:6.45549(Vrms) evm:11.1(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.321276 SCP
DATA sid:0 Inid:9567 Lcid:256
00 | 00 00 5e 05 00 00 95 7c 2f[82 90 03 40 00 06 00
10 | 00 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00
20 | 00 00 02 08 09 0c 07 e0 09 13 01 14 00 00 00 80
30 | 00 80 11 00 06 00 00 00]44 bd 15 27
[RX] 2016-09-20 18:17:48 6312.056464(0.026688) dbpsk_f rxpow:6.40725(Vrms) evm:11.1(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.359028 SCP
DATA sid:0 Inid:9567 Lcid:256
00 | 00 00 5e 05 00 00 95 7c 2f[83 90 03 41 00 06 00
10 | 00 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00
20 | 00 00 06 00 00 00 00 02 08 09 0c 07 e0 09 13 01
30 | 15 00 00 00 80 00 80 11]91 58 01 04
[RX] 2016-09-20 18:17:48 6312.090436(0.026688) dbpsk_f rxpow:6.34722(Vrms) evm:11.1(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.393000 SCP
DATA sid:0 Inid:9567 Lcid:256
00 | 00 00 5e 05 00 00 95 7c 2f[84 90 03 42 00 06 00
10 | 00 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00
20 | 00 00 06 00 00 00 00 06 00 00 00 02 08 09 0c
30 | 07 e0 09 13 01 16 00 00]df ce 61 2e
[RX] 2016-09-20 18:17:48 6312.123252(0.026688) dbpsk_f rxpow:6.54230(Vrms) evm:11.1(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.425816 SCP
DATA sid:0 Inid:9567 Lcid:256
00 | 00 00 5e 05 00 00 95 7c 2f[85 90 03 43 00 80 00
10 | 80 11 00 06 00 00 00 00 06 00 00 00 00 06 00 00
20 | 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00 00
30 | 00 02 08 09 0c 07 e0 09]a9 a5 2b ca
[RX] 2016-09-20 18:17:48 6312.156900(0.026688) dbpsk_f rxpow:6.42887(Vrms) evm:11.4(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.459464 SCP
DATA sid:0 Inid:9567 Lcid:256
00 | 00 00 5e 05 00 00 95 7c 2c[c6 90 03 84 13 01 17
10 | 00 00 00 80 00 80 11 00 06 00 00 00 00 06 00 00
20 | 00 00 06 00 00 00 00 06 00 00 00 00 06 00 00 00
30 | 00 06 00 00 00]01 e6 5b 71
[RX] 2016-09-20 18:17:48 6312.254204(0.015488) dbpsk_f rxpow:6.76755(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.556768 SCP
DATA sid:0 Inid:9567 Lcid:256
00 | 00 40 99 05 00 00 95 7c 0e[c3 90 07 00 90 01 02
10 | c0 02 c1 00 00 00 01]02 8d 43 da
```

Figura 52 Transmisión de un bloque completo a nivel de aplicación ZIV

```
[RX] 2016-09-20 18:17:50 6314.293428(0.026688) dbpsk_f rxpow:6.26969(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.123028 SCP
DATA sid:0 Inid:9567 Lcid:256
00 | 00 00 5e 05 00 00 95 7c 2e[99 90 07 04 90 02 01
10 | c4 02 c1 01 00 00 05 00 81 c0 02 08 09 0c 07
20 | e0 09 14 02 0f 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00]eb c0 54 d9
```

Figura 53 Fin transmisión Informe S02 ZIV

Una vez conocido el formato y la metodología de comunicación establecida para ZIV procedemos al análisis manual de las trazas de mensajes reales proporcionadas.

El módulo software desarrollado nos proporciona el par de valores SID/LNID y $TTRi$ por lo que será fácil comparar los resultados obtenidos mediante el análisis manual de las trazas con los obtenidos mediante el procesado automático de las mismas.

- **Nodo 09459**

- Solicitud infome S02 (RQ): c0 01 c1 00 07 01 00 63 01 00 ff

```
[RX] 2016-09-20 18:08:34 5758.087032(0.024448) dbpsk_f rxpow:6.70239(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.334956 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 40 99 05 00 00 93 cc 26[81 90 21 01 90 01 02
10 | c0 01 c1 00 07 01 00 63 01 00 ff 02 01 01 02 04
20 | 02 04 12 00 08 09 06 00 00 01 00 00 ff 0f 02]50
30 | 7a c6 3b
```

Figura 54 Solicitud S02 al contador 09459

- Fin transmisión S02 (RSP): c4 c2 c1 01

```
[RX] 2016-09-20 18:08:38 5761.880716(0.026688) dbpsk_f rxpow:6.27013(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.419184 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2e[b9 90 07 04 90 02 01
10 | c4 02 c1 01 00 00 00 05 00 81 c0 02 08 09 0c 07
20 | e0 09 14 02 0f 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00]da 89 e9 e5
```

Figura 55 Evidencia recepción último bloque informe S02 del contador 09459

Con los datos representados podemos obtener el valor del $TTRi$ para el contador analizado según la siguiente ecuación:

$$TTRi_{09459} = 5761.880716 - 5758.087032 = 3.793684$$

- **Nodo 09970**

- Solicitud infome S02 (RQ): c0 01 c1 00 07 01 00 63 01 00 ff

```
[RX] 2016-09-20 18:14:54 6138.166316(0.024448) dbpsk_f rxpow:6.67637(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.195052 SCP
DATA sid:0 Inid:9970 lcid:256
00 | 00 40 99 05 00 00 9b c8 26[81 90 01 01 90 01 02
10 | c0 01 c1 00 07 01 00 63 01 00 ff 02 01 01 02 04
20 | 02 04 12 00 08 09 06 00 00 01 00 00 ff 0f 02]1c
30 | fc be 31
```

Figura 56 Solicitud S02 al contador 09970

- Fin transmisión S02 (RSP): c4 c2 c1 01

```
[RX] 2016-09-20 18:14:57 6140.850096(0.026688) dbpsk_f rxpow:6.27722(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.405860 SCP
DATA sid:0 Inid:9970 lcid:256
00 | 00 00 5e 05 00 00 9b c8 2e[99 90 07 04 90 02 01
10 | c4 02 c1 01 00 00 00 05 00 81 c0 02 08 09 0c 07
20 | e0 09 14 02 0f 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00]f5 e5 61 03
```

Figura 57 Evidencia recepción último bloque informe S02 del contador 09970

Con los datos representados podemos obtener el valor del $TTRI$ para el contador analizado según la siguiente ecuación:

$$TTRI_{09970} = 6138.166316 - 6140.850096 = 2.68378$$

- **Nodo 09567**

- Solicitud infome S02 (RQ): c0 01 c1 00 07 01 00 63 01 00 ff

```
[RX] 2016-09-20 18:17:47 6311.488212(0.024448) dbpsk_f rxpow:6.61275(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.409024 SCP
DATA sid:0 Inid:9567 lcid:256
00 | 00 40 99 05 00 00 95 7c 26[81 90 01 01 90 01 02
10 | c0 01 c1 00 07 01 00 63 01 00 ff 02 01 01 02 04
20 | 02 04 12 00 08 09 06 00 00 01 00 00 ff 0f 02]d7
30 | 03 4b 27
```

Figura 58 Solicitud S02 al contador 09567

- Fin transmisión S02 (RSP): c4 c2 c1 01

```
[RX] 2016-09-20 18:17:50 6314.293428(0.026688) dbpsk_f rxpow:6.26969(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.123028 SCP
DATA sid:0 Inid:9567 lcid:256
00 | 00 00 5e 05 00 00 95 7c 2e[99 90 07 04 90 02 01
10 | c4 02 c1 01 00 00 00 05 00 81 c0 02 08 09 0c 07
20 | e0 09 14 02 0f 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00]eb c0 54 d9
```

Figura 59 Evidencia recepción último bloque informe S02 del contador 09567

Con los datos representados podemos obtener el valor del $TTRI$ para el contador analizado según la siguiente ecuación:

$$TTRI_{09567} = 6314.293428 - 6311.488212 = 2.805216$$

- **Nodo 08717**

- Solicitud infome S02 (RQ): c0 01 c1 00 07 01 00 63 01 00 ff

```
[RX] 2016-09-20 18:20:37 6481.354592(0.024448) dbpsk_f rxpow:6.75956(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.258688 SCP
DATA sid:0 Inid:8717 lcid:256
00 | 00 40 99 05 00 00 88 34 26[81 90 01 01 90 01 02
10 | c0 01 c1 00 07 01 00 63 01 00 ff 02 01 01 02 04
20 | 02 04 12 00 08 09 06 00 00 01 00 00 ff 0f 02]bf
30 | ff 3d b7
```

Figura 60 Solicitud S02 al contador 08717

- Fin transmisión S02 (RSP): c4 c2 c1 01

```
[RX] 2016-09-20 18:20:40 6484.036920(0.026688) dbpsk_f rxpow:6.51675(Vrms) evm:11.1(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.468052 SCP
DATA sid:0 Inid:8717 lcid:256
00 | 00 00 5e 05 00 00 88 34 2e[99 90 07 04 90 02 01
10 | c4 02 c1 01 00 00 00 05 00 81 c0 02 08 09 0c 07
20 | e0 09 14 02 0f 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00]d9 47 dc 02
```

Figura 61 Evidencia recepción último bloque informe S02 del contador 08717

Con los datos representados podemos obtener el valor del $TTRi$ para el contador analizado según la siguiente ecuación:

$$TTRi_{08717} = 6484.0369220 - 6481.354592 = 2.682328$$

De los cálculos realizados podemos apreciar que se cumple el segundo punto de validación: las cuatro muestras analizadas son iguales para ambos métodos de análisis. La Tabla 2 muestra los resultados obtenidos.

N	TTRi	TTRi	Módulo
odo	Manual	Standalone	
0	3.79368	3.793684	
9459	4		
0	2.68378	2.68378	
9970			
0	2.80521	2.805216	
9567	6		
0	2.68232	2.682328	
8717	8		

Tabla 2 TTRi ZIV

5.1 VALIDACIÓN DEL MÓDULO STANDALONE PARA EL CÁLCULO DEL TTRI

Por último, queda verificar que el cálculo automático es solvente ante anomalías en la red, como por ejemplo retransmisiones. Este hecho, para el caso particular de la red PRIME analizada se puede ver reflejado en el intercambio de mensajes realizado entre el CNTR y el contador 09459 según el siguiente fragmento de traza en el que se incluyen las más relevantes.

```
[RX] 2016-09-20 18:08:34 5758.087032(0.024448) dbpsk_f rxpow:6.70239(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.334956 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 40 99 05 00 00 93 cc 26[81 90 21 01 90 01 02
10 | c0 01 c1 00 07 01 00 63 01 00 ff 02 01 01 02 04
20 | 02 04 12 00 08 09 06 00 00 01 00 00 ff 0f 02]50
30 | 7a c6 3b
[RX] 2016-09-20 18:08:34 5758.508800(0.026688) dbpsk_f rxpow:6.30373(Vrms) evm:10.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.138476 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[a1 90 03 05 90 02 01
10 | c4 02 c1 00 00 00 00 01 00 81 f1 01 18 02 08 09
20 | 0c 07 e0 09 13 01 13 00 00 00 80 00 80 11 00 06
30 | 00 00 00 06 00 00 00]77 62 f9 ef
[RX] 2016-09-20 18:08:35 5758.800540(0.015488) dbpsk_f rxpow:6.78304(Vrms) evm:14.7(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.430216 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 40 99 05 00 00 93 cc 0e[c3 90 27 00 90 01 02
10 | c0 02 c1 00 00 00 01]16 f9 d6 44
[RX] 2016-09-20 18:08:35 5759.034432(0.026688) dbpsk_f rxpow:6.07842(Vrms) evm:10.5(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.045872 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[a7 90 04 05 90 02 01
10 | c4 02 c1 00 00 00 00 02 00 81 f1 00 02 08 09 0c
20 | 07 e0 09 14 02 00 00 00 00 80 00 80 11 00 06 00
30 | 00 00 06 00 00 00 00]63 d9 ad c9
[RX] 2016-09-20 18:08:35 5759.329336(0.015488) dbpsk_f rxpow:6.80192(Vrms) evm:14.6(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.340776 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 40 99 05 00 00 93 cc 0e[c4 90 2d 00 90 01 02
10 | c0 02 c1 00 00 00 02]f8 e6 5b 51
[RX] 2016-09-20 18:08:35 5759.541560(0.026688) dbpsk_f rxpow:6.06834(Vrms) evm:10.5(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.553000 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[ad 90 05 05 90 02 01
10 | c4 02 c1 00 00 00 00 03 00 81 f0 02 08 09 0c 07
20 | e0 09 14 02 05 00 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00 06]ab df 14 3f
[RX] 2016-09-20 18:08:36 5759.854548(0.015488) dbpsk_f rxpow:6.77337(Vrms) evm:14.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.247744 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 40 99 05 00 00 93 cc 0e[c5 90 33 00 90 01 02
10 | c0 02 c1 00 00 00 03]eb 3f ea 63
[RX] 2016-09-20 18:08:36 5760.079384(0.026688) dbpsk_f rxpow:6.07423(Vrms) evm:10.3(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be (Invalid CRC) UP level:0 frametime:0.472580 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[b3 90 06 05 90 02 01
10 | c4 02 c1 00 00 00 00 04 00 81 f0 02 08 09 0c 07
20 | e0 09 14 02 0a 00 00 00 80 00 80 f2 37 84 14 cd
30 | c3 3e 7b c9 20 a8 4c ce]ce 88 ef 06
[RX] 2016-09-20 18:08:36 5760.619712(0.026688) dbpsk_f rxpow:6.07120(Vrms) evm:10.5(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.394660 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2f[f3 90 06 05 90 02 01
10 | c4 02 c1 00 00 00 00 04 00 81 f0 02 08 09 0c 07
20 | e0 09 14 02 0a 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00 06]9e 76 9b c5
[RX] 2016-09-20 18:08:38 5761.690264(0.015488) dbpsk_f rxpow:6.77325(Vrms) evm:14.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be DO level:0 frametime:0.228732 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 40 99 05 00 00 93 cc 0e[c6 90 39 00 90 01 02
10 | c0 02 c1 00 00 00 04]0d 3a 37 19
[RX] 2016-09-20 18:08:38 5761.880716(0.026688) dbpsk_f rxpow:6.27013(Vrms) evm:10.9(dB) channels:phy0
GPDU: sna:40:40:22:02:27:be UP level:0 frametime:0.419184 SCP
DATA sid:0 Inid:9459 lcid:256
00 | 00 00 5e 05 00 00 93 cc 2e[b9 90 07 04 90 02 01
10 | c4 02 c1 01 00 00 00 05 00 81 c0 02 08 09 0c 07
20 | e0 09 14 02 0f 00 00 00 80 00 80 11 00 06 00 00
30 | 00 00 06 00 00 00 00]da 89 e9 e5
```

Figura 62. Ejemplo de tráfico anómalo en ZIV

Como vemos, el concentrador recibe una traza errónea, marcada en rojo. Dicha traza es descartada por el programa, con lo que no se incrementa el número de secuencia esperado, hecho que hace que el siguiente bloque con número de secuencia 4 se procese correctamente, permitiendo incrementar el número de secuencia a 5, en lugar de 6, que será la traza utilizada para el cálculo del $TTRi$ de dicho nodo. Si no se tuvieran en cuenta este tipo de anomalías el $TTRi$ para dicho nodo nunca tendría valor, ya que no recibiría un *LAST BLOCK TRUE* con el número de secuencia esperado.

Podemos concluir que el programa funciona correctamente, solventa los problemas detectados en el procesado de las trazas y calcula el $TTRi$ correctamente para el caso de ZIV.

Por último, con la información obtenida del análisis de trazas, el intercambio de mensajes que se origina entre concentrador y contador, la velocidad de transmisión a nivel PHY y lo que ocupa la cabecera PHY podemos calcular el $TTRi$ teórico. Dicho cálculo no es el centro de nuestra validación, sino que simplemente nos sirve para hacernos una idea de que en un escenario sencillo, el $TTRi$ medido en campo se aproxima más al ideal que en un escenario real, ya que en el segundo el efecto de muchos otros factores en la calidad de las comunicaciones se hace más notable. Para afianzar los datos obtenidos vamos a comparar los resultados obtenidos con el $TTRi$ teórico.

Para obtener el cálculo del $TTRi$ teórico es necesario conocer el intercambio completo que se produce entre Concentrador y Nodo en la solicitud de un informe de consumo S02. Esta información queda plasmada en las Figuras 49 y 65 para el caso de fabricante ZIV y CIRCUTOR respectivamente.

Del estándar PRIME podemos determinar todos los parámetros que componen una traza.

$$PREAMBLE = 2,048ms$$

$$PHY\ HEADER = 4,048ms$$

$$Raw\ Data\ Rate = 21,4 \frac{bit}{ms}$$

De esta manera podemos calcular el tiempo que tarda en transmitirse cada una de las trazas descritas en la Figura 49. La Tabla 3 muestra el número total de bytes por cada tipo de traza PRIME que interviene en la solicitud de un informe completo de consumo (S02) para ZIV.

DATA	REQ S02	REQ NEXT BLOCK	RSP DATA
Payload (bytes)	75	14	47
PACKET HEADER (Bytes)	12	6	6
MAC HEADER (Bytes)	6	3	3
CRC (Bytes)	8	4	4
Total Bytes	101	27	60

Tabla 3 Formato de traza ZIV en bytes

El tiempo que tarda en transmitirse cada tipo de traza se calcula mediante la siguiente formula:

$$T_{tx} = \frac{(NBytes * 8bits)}{Raw Data Rate} + 4,048ms + 2,048ms$$

DATA	REQ S02	REQ NEXT BLOCK	RSP DATA
Nº Trazas	1	4	30
Total Bytes	101	27	60
T _{tx} (ms)	43,853009	16,1894579	28,5259065
Total T _{tx} (ms)	43,853009	64,7578316	855,777196

Tabla 4 Cálculo tiempo transmisión por tipo de traza (ZIV)

Por lo tanto el *TTRI* teórico se calcula en base a la información mostrada en la Tabla 4 de la siguiente manera:

$$TTRI = \sum T_{tx} = 964.388036 ms$$

El tiempo calculado correspondería al *TTRI* en un entorno ideal, sin pérdidas, sin retransmisiones y sin retornos en los medios de conexión por lo que nos sirve para compararlo con los resultados obtenidos de las pruebas realizadas en el panel del laboratorio.

Verificamos que los tiempos obtenidos mediante el programa se aproximan al cálculo teórico. La diferencia se debe a que, a pesar de tratarse de un entorno de laboratorio y baja tasa de error, se detectan en las trazas capturadas retransmisiones, por lo que los tiempos en el caso real se verán incrementadas respecto al teórico.

Para validar los resultados obtenidos del análisis de trazas de CIRCUTOR, procedemos de la misma manera. El primer paso es analizar las trazas procedentes de una red PRIME con dispositivos del fabricante CIRCUTOR. Conocemos de antemano que la red analizada está formada por 65 contadores inteligentes por lo que el primer punto de análisis es verificar que obtenemos 65 medias. En la Figura 63 vemos los resultados obtenidos para este caso.

```

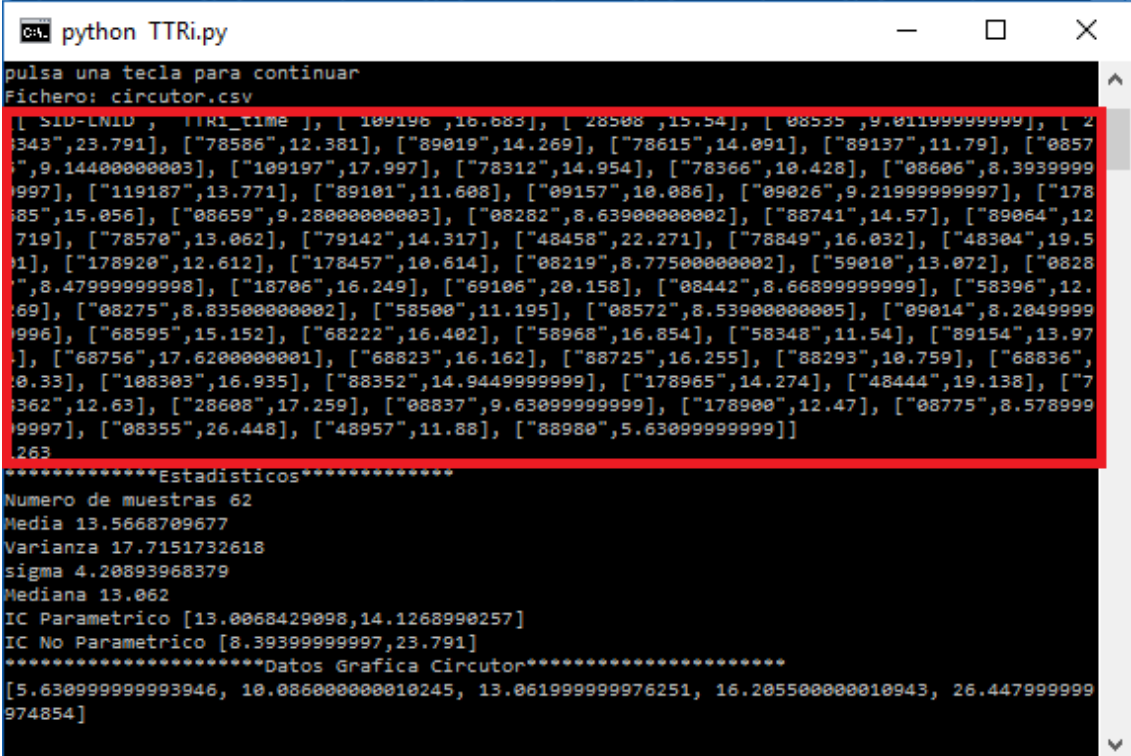
C:\WINDOWS\system32\cmd.exe - python TTRi.py
Has pulsado la opcion 2...
pulsas una tecla para continuarcircutor.csv
Fichero: circutor.csv
[[["SID-LNID", "TTRi_time"], [{"109196",16.683}, {"28508",15.54}, {"08535",9.011999999999}, {"28343",23.791}, {"78586",12.381}, {"89019",14.269}, {"78615",14.091}, {"89137",11.79}, {"08575",9.144000000003}, {"109197",17.997}, {"78312",14.954}, {"78366",10.428}, {"08606",8.393999999997}, {"119187",13.771}, {"89101",11.608}, {"09157",10.086}, {"09026",9.219999999997}, {"178685",15.056}, {"08659",9.280000000003}, {"08282",8.639000000002}, {"88741",14.57}, {"89064",12.719}, {"78570",13.062}, {"79142",14.317}, {"48458",22.271}, {"78849",16.032}, {"48304",19.501}, {"178920",12.612}, {"178457",10.614}, {"08219",8.775000000002}, {"59010",13.072}, {"08287",8.479999999998}, {"18706",16.249}, {"69106",20.158}, {"08442",8.668999999999}, {"58396",12.269}, {"08275",8.835000000002}, {"58500",11.195}, {"08572",8.539000000005}, {"09014",8.204999999996}, {"68595",15.152}, {"68222",16.402}, {"58968",16.854}, {"58348",11.54}, {"89154",13.974}, {"68756",17.620000000001}, {"68823",16.162}, {"88725",16.255}, {"88293",10.759}, {"68836",20.33}, {"108303",16.935}, {"88352",14.944999999999}, {"178965",14.274}, {"48444",19.138}, {"78362",12.63}, {"28608",17.259}, {"08837",9.630999999999}, {"178900",12.47}, {"08775",8.578999999997}, {"08355",26.448}, {"48957",11.88}, {"88980",5.630999999999}]]
1263
*****Datos Grafica CIRCUTOR*****
Numero de muestras 62
Media 13.3000705077
Varianza 17.7151732618
sigma 4.20893968379
Mediana 13.062
IC Parametrico [13.0068429098,14.1268990257]
IC No Parametrico [8.393999999997,23.791]
*****Datos Grafica CIRCUTOR*****
[5.6309999999999946, 10.086000000010245, 13.061999999976251, 16.205500000010943, 26.447999999974854]
    
```

Figura 63 Salida Módulo standalone para CIRCUTOR

5.1 VALIDACIÓN DEL MÓDULO STANDALONE PARA EL CÁLCULO DEL TTRI

De la Figura 63 podemos apreciar que se han cálculo 62 muestras a pesar de que la red PRIME consta de 65. A pesar de ello, el dato es correcto, ya que analizando la captura de trazas verificamos que únicamente existen 62 solicitudes de informe S02, por lo que la captura no contiene trazas de todos los contadores. Por otro lado, también cabe destacar que los valores obtenidos están próximos entre sí.

La Figura 64 muestra por pantalla los resultados obtenidos del procesado de las trazas reales facilitadas en pares de SID/LNID (identificador único del nodo en la red PRIME) y el tiempo necesario para enviar el informe S02 de consumo por parte del nodo al recibir una solicitud a tal efecto.



```
python TTRi.py
pulsar una tecla para continuar
Fichero: circutor.csv
[ ["SID-LNID", "TTRI_time"], ["109196", 16.665], ["28508", 15.54], ["08535", 9.011999999999], ["2343", 23.791], ["78586", 12.381], ["89019", 14.269], ["78615", 14.091], ["89137", 11.79], ["0857", 9.144000000003], ["109197", 17.997], ["78312", 14.954], ["78366", 10.428], ["08606", 8.393999999997], ["119187", 13.771], ["89101", 11.608], ["09157", 10.086], ["09026", 9.219999999997], ["17885", 15.056], ["08659", 9.280000000003], ["08282", 8.639000000002], ["86741", 14.57], ["89064", 12.719], ["78570", 13.062], ["79142", 14.317], ["48458", 22.271], ["78849", 16.032], ["48304", 19.541], ["178920", 12.612], ["178457", 10.614], ["08219", 8.775000000002], ["59010", 13.072], ["0828", 8.479999999998], ["18706", 16.249], ["69106", 20.158], ["08442", 8.668999999999], ["58396", 12.69], ["08275", 8.835000000002], ["58500", 11.195], ["08572", 8.539000000005], ["09014", 8.204999999996], ["68595", 15.152], ["68222", 16.402], ["58968", 16.854], ["58348", 11.54], ["89154", 13.97], ["68756", 17.620000000001], ["68823", 16.162], ["88725", 16.255], ["88293", 10.759], ["68836", 10.33], ["108303", 16.935], ["88352", 14.944999999999], ["178965", 14.274], ["48444", 19.138], ["7362", 12.63], ["28608", 17.259], ["08837", 9.630999999999], ["178900", 12.47], ["08775", 8.578999999997], ["08355", 26.448], ["48957", 11.88], ["88980", 5.630999999999]
]
263
*****Estadisticos*****
Numero de muestras 62
Media 13.5668709677
Varianza 17.7151732618
sigma 4.20893968379
Mediana 13.062
IC Parametrico [13.0068429098,14.1268990257]
IC No Parametrico [8.393999999997,23.791]
*****Datos Grafica Circutor*****
[5.630999999999999, 10.0860000000010245, 13.0619999999976251, 16.205500000010943, 26.447999999974854]
```

Figura 64. TTRI obtenido en el análisis de trazas reales para CIRCUTOR

Continuando con la metodología descrita, y teniendo en cuenta que la red analizada es grande, no podemos comparar los resultados del script con la inspección manual de las trazas de tráfico reales en su totalidad, por lo que se elegirán 4 identificadores aleatorios para verificar sus resultados.

La Figura 65 muestra el intercambio de mensaje existente entre CNTR y contador para el caso particular de CIRCUTOR, obtenido mediante el análisis de las trazas. Como vimos en el apartado 2.3, el nivel de aplicación, protocolo DLMS/COSEM, está basado en un sistema cliente-servidor, por lo que queda caracterizado por los siguientes tipos de mensajes:

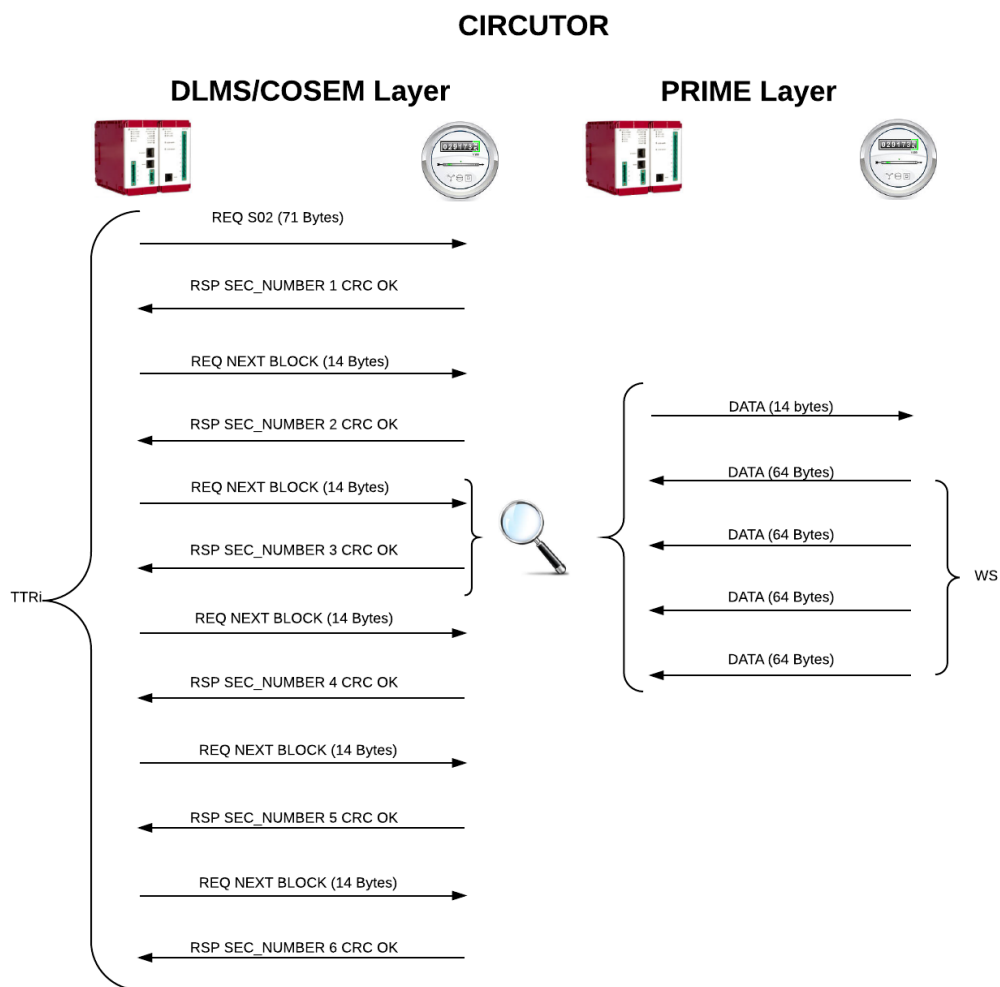


Figura 65 Intercambio de mensajes entre CNTR y contador (CIRCUTOR)

Dicho intercambio puede verse en formato real de trazas en la Figura 66.

El intercambio de mensajes visto en la Figura 68 se repite un total de 5 veces. El último bloque correspondiente a la transmisión del informe S02 completo se ilustra en la Figura 69.

```
00:47:08.6,370010117,822,Data,UP,000,OK,02,21a0,4,64,OK,LCID: 0x0100 DATA:
b8811302900101c402c10100000006006f0600000000060000000006000000440208090c07e0011c04170000008
000041100060000014f060000000006000000,BPSK-FEC,15,7,-,
```

Figura 69 Fin transmisión Informe S02 CIRCUTOR

Una vez conocido el formato y la metodología de comunicación establecida para CIRCUTOR, procedemos al análisis manual de las trazas de mensajes reales proporcionadas.

El módulo software desarrollado nos proporciona el par de valores SID/LNID y TTR_i por lo que será fácil comparar los resultados obtenidos mediante el análisis manual de las trazas con los obtenidos mediante el procesado automático de las mismas.

- **Nodo 109196**
 - Solicitud infome S02 (RQ): c001c100070100630100ff

```
00:25:13.2,368694687,92,Data,DW,000,OK,0a,23ec,4,71,OK,LCID: 0x0100 DATA:
cd902400900101c001c100070100630100ff0201010204020412000809060000010000ff0f02120000090c07
e0011c0400000100800000090c07e0011d05001900008000000100,BPSK-FEC,-,-,-,
```

Figura 70 Solicitud S02 al contador 109196

- Fin transmisión S02 (RSP): c4c2c101

```
00:25:29.9,368711370,500,Data,UP,000,OK,0a,23ec,4,64,OK,LCID: 0x0100 DATA:
f8811302900101c402c10100000006006f060000000006000000000600000010208090c07e0011c0417000
000800004110006000002fd060000000006000000,BPSK-FEC,15,7,-,
```

Figura 71 Evidencia recepción último bloque informe S02 del contador 109196

Con los datos representados podemos obtener el valor del TTR_i para el contador analizado según la siguiente ecuación:

$$TTR_{i_{109196}} = 368711370 - 368694687 = 16683 \text{ ms} = 16,683 \text{ s}$$

La Figura 72 muestra el cálculo obtenido en el módulo *standalone* donde se verifica que el resultado es el mismo para el contador analizado.

Figura 72 Validación de resultados CIRCUTOR para el contador con NID 109196

- **Nodo 59010**
 - 1.- Solicitud infome S02 (RQ): c001c100070100630100ff

```
00:36:59.0,369400514,89,Data,DW,000,OK,05,2332,4,71,OK,LCID: 0x0100 DATA:
f6900900900101c001c100070100630100ff0201010204020412000809060000010000ff0f02120000090c07e
0011c0400000100800000090c07e0011d05001900008000000100,BPSK-FEC,-,-,
```

Figura 73 Solicitud S02 al contador 59010

- 2.- Fin transmisión S02 (RSP): c4c2c101

```
00:37:12.1,369413586,679,Data,UP,000,OK,05,2332,4,64,OK,LCID: 0x0100 DATA:
9d813c02900101c402c10100000006006f060000000060000000060000000510208090c07e0011c0417000
0008000041100060000012806000000006000000,BPSK-FEC,15,7,-,
```

Figura 74 Evidencia recepción último bloque informe S02 del contador 59010

Con los datos representados podemos obtener el valor del *TTRI* para el contador analizado según la siguiente ecuación:

$$TTRI_{59010} = 369413586 - 369400514 = 13072 \text{ ms} = 13,072 \text{ s}$$

La Figura 75 muestra el cálculo obtenido en el módulo *standalone* donde se verifica que el resultado es el mismo para el contad analizado.

```

python TTRi.py
Has pulsado la opcion 2...
pulsas una tecla para continuar
Archivo: circutor.csv
[["SID-LNID", "TTRi_time", ["199196",16.683], ["28508",15.54], ["08535",9.01199999999], ["28343",23.791], ["78586",12.381], ["89019",14.269],
["78615",14.091], ["89137",11.79], ["08575",9.14400000003], ["109197",17.997], ["78312",14.954], ["78366",10.428], ["08606",8.39399999997],
["119187",13.771], ["89101",11.608], ["09157",10.086], ["09026",9.21999999997], ["178685",15.056], ["08659",9.28000000003], ["08282",8.63900
000002], ["88741",14.57], ["89064",12.719], ["78570",13.062], ["79142",14.317], ["48458",22.271], ["78849",16.032], ["48304",19.501], ["17892
0",12.612], ["178457",10.614], ["08219",8.77500000002], ["59010",13.072], ["08287",8.47999999998], ["18706",16.249], ["69106",20.158], ["0844
2",8.66899999999], ["58396",12.269], ["08275",8.83500000002], ["83300",11.195], ["08572",8.53900000005], ["09014",8.20499999996], ["68595",15
.152], ["68222",16.402], ["58968",16.854], ["58348",11.54], ["89154",13.974], ["68756",17.6200000001], ["68823",16.162], ["88725",16.255], ["
88293",10.759], ["68836",20.33], ["108303",16.935], ["88352",14.9449999999], ["178965",14.274], ["48444",19.138], ["78362",12.63], ["28608",1
7.259], ["08837",9.63099999999], ["178900",12.47], ["08775",8.57899999997], ["08355",26.448], ["48957",11.88], ["88980",5.63099999999]]
1263
*****Estadisticos*****
Numero de muestras 62
Media 13.5668709677
Varianza 17.7151732618
sigma 4.20893968379
Mediana 13.062
IC Parametrico [13.0068429098, 14.1268990257]
IC No Parametrico [8.39399999997, 23.791]
*****Datos Grafica Circutor*****
[5.630999999999946, 10.086000000010245, 13.061999999976251, 16.205500000010943, 26.447999999974854]
    
```

Figura 75 Validación de resultados CIRCUTOR para el contador con NID 59010

- **Nodo 68222**

- Solicitud infome S02 (RQ): c001c100070100630100ff

```

00:40:41.8,369623282,98,Data,DW,000,OK,06,201e,4,71,OK,LCID: 0x0100 DATA:
c3900600900101c001c100070100630100ff0201010204020412000809060000010000ff0f0212000090c07
e0011c0400000100800000090c07e0011d05001900008000000100,BPSK-FEC,-,-,
    
```

Figura 76 Solicitud S02 al contador 68222

- Fin transmisión S02 (RSP): c4c2c101

```

00:40:58.2,369639684,947,Data,UP,000,OK,06,201e,4,64,OK,LCID: 0x0100 DATA:
9a810902900101c402c10100000006006f060000000006000000006000000020208090c07e0011c041700
000080000411006000001dd060000000006000000,BPSK-FEC,15,7,-,
    
```

Figura 77 Evidencia recepción último bloque informe S02 del contador 68222

Con los datos representados podemos obtener el valor del *TTRi* para el contador analizado según la siguiente ecuación:

$$TTRi_{68222} = 369639684 - 369623282 = 16402 \text{ ms} = 16,402 \text{ s}$$

La Figura 78 muestra el cálculo obtenido en el módulo *standalone* donde se verifica que el resultado es el mismo para el contador analizado.

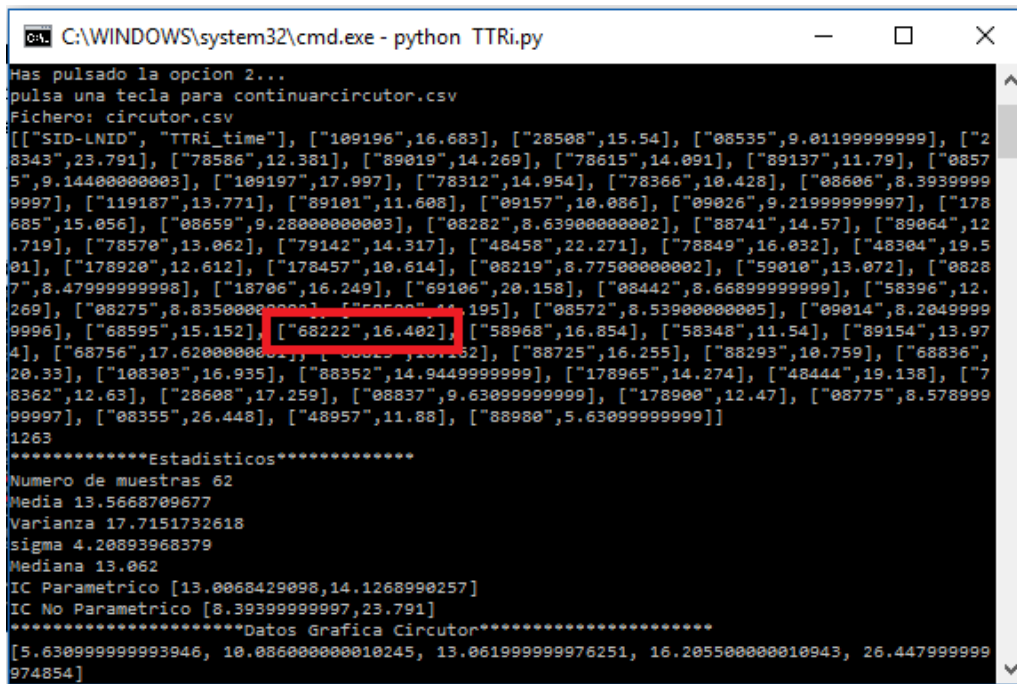


Figura 78 Validación de resultados CIRCUTOR para el contador con NID 68222

- **Nodo 178900**

- Solicitud infome S02 (RQ): c0 01 c1 00 07 01 00 63 01 00 ff

```

00:47:34.7,370036207,131,Data,DW,000,OK,11,22c4,4,71,OK,LCID: 0x0100 DATA:
d7900200900101c001c100070100630100ff0201010204020412000809060000010000ff0f0212000090c07
e0011c0400000100800000090c07e0011d05001900008000000100,BPSK-FEC,-,-,
    
```

Figura 79 Solicitud S02 al contador 178900

- Fin transmisión S02 (RSP): c4 c2 c1 01

```

00:47:47.2,370048677,119,Data,UP,000,OK,11,22c4,4,64,OK,LCID: 0x0100 DATA:
96811d02900101c402c10100000006006f060000000060000000060000001c0208090c07e0011c041700
00008000041100060000006e06000000006000000,BPSK-FEC,15,7,-,
    
```

Figura 80 Evidencia recepción último bloque informe S02 del contador 178900

Con los datos representados podemos obtener el valor del *TTRI* para el contador analizado según la siguiente ecuación:

$$TTRI_{178900} = 370048677 - 370036207 = 12470 \text{ ms} = 12,47 \text{ s}$$

En la Figura 83 vemos un intercambio de trazas en la que la solicitud de S02 se realiza repetidas veces debido a un fallo en la comunicación. El $TTRi$ correcto sería el resultado de calcular el tiempo transcurrido entre la última solicitud S02 y la recepción del último bloque del informe completo que correspondería con las dos trazas marcadas en negro.

$$TTRi = 369983766 - 369971136 = 12630 \text{ ms} = 12,63 \text{ s}$$

Dicho cálculo es exactamente igual al obtenido mediante el análisis automático, como se ilustra en la Figura 84.

```

Selecciona una opcion
 1 - TRAZAS ZIU
 2 - TRAZAS CIRCUTOR
 0 - salir
Seleccione el formato de trazas del archivo a analizar >> 2

Has pulsado la opcion 2...
pulsas una tecla para continuarcircutor.csv
Archivo: circutor.csv
[["SID-LNID", "TTRi_time"], ["109196",16.683], ["28500",15.54], ["08535",9.011999999999], ["2834
3",23.791], ["70586",12.381], ["89019",14.269], ["78615",14.091], ["89137",11.79], ["08575",9.1
4400000003], ["109197",17.997], ["78312",14.954], ["78366",10.428], ["08606",8.393999999999], ["
119187",13.771], ["89101",11.608], ["09157",10.086], ["09026",9.219999999999], ["178685",15.056],
["08659",9.28000000003], ["08282",8.63900000002], ["88741",14.57], ["89064",12.719], ["78570",
13.062], ["79142",14.317], ["48458",22.271], ["78849",16.032], ["48304",19.501], ["178920",12.
612], ["178457",10.614], ["08219",8.77500000002], ["59010",13.072], ["08287",8.47999999998], ["
18706",16.249], ["69106",20.158], ["08442",8.66899999999], ["58396",12.269], ["08275",8.8350000
0002], ["58500",11.195], ["08572",8.53900000005], ["09014",8.20499999996], ["68595",15.152], ["
68222",16.402], ["58968",16.854], ["58348",11.54], ["89154",13.974], ["68756",17.62000000001], ["
68823",16.162], ["88725",16.255], ["88293",10.759], ["08507",20.999], ["108303",16.935], ["8835
2",14.9449999999], ["178965",14.274], ["48444",19.138], ["78362",12.63], ["28600",17.259], ["08
837",9.63099999999], ["178900",12.47], ["08775",8.57899999999], ["08000",10.000], ["08000",10.000], ["
81", ["88980",5.63099999999]]
1263
    
```

Figura 84 Salida del módulo standalone para CIRCUTOR con intercambio anómalo

Para el cálculo se han descartado todas aquellas trazas cuyo número de secuencia no era el esperado. Con estas acciones validamos el módulo software desarrollado, ya que podemos indicar que es robusto frente a errores en la transmisión.

Por último, nos queda obtener el $TTRi$ teórica para el caso de CIRCUTOR. Continuando con la metodología seguida para el caso de ZIV podemos calcular el tiempo que tarda en transmitirse cada una de las trazas descritas en la Figura 65. La Tabla 6 muestra el número total de bytes por cada tipo de traza PRIME que interviene en la solicitud de un informe completo de consumo (S02) para CIRCUTOR.

DATA	REQ S02	REQ NEXT BLOCK	RSP DATA
Payload (bytes)	71	14	64
PACKET HEADER (Bytes)	6	6	6
MAC HEADER (Bytes)	3	3	3
CRC (Bytes)	4	4	4
Total Bytes	84	27	77

Tabla 6 Formato de traza CIRCUTOR en bytes

El tiempo que tarda en transmitirse cada tipo de traza se calcula mediante la siguiente formula:

$$T_{tx} = \frac{(NBytes * 8bits)}{Raw Data Rate} + 4,048ms + 2,048ms$$

DATA	REQ S02	REQ NEXT BLOCK	RSP DATA
Nº Trazas	1	5	24
Total Bytes	84	27	77
T _{tx} (ms)	37,497869	16,189458	34,881047,78
Total T _{tx} (ms)	37,497869	80,94729	837,145121

Tabla 7 Cálculo tiempo transmisión por tipo de traza (CIRCUTOR)

La Tabla 7 muestra el tiempo de transmisión utilizada para transmitir las trazas de cada tipo. Por lo tanto el *TTRI* teórico es:

$$TTRI = \sum T_{tx} = 955.590258 \text{ ms}$$

El tiempo calculado correspondería al *TTRI* en un entorno ideal, sin pérdidas, sin retransmisiones y sin retornos en los medios de conexión por lo que nos sirve para compararlo con los resultados obtenidos de las pruebas realizadas en un escenario real.

En este caso los valores obtenidos mediante el análisis de trazas correspondientes a un escenario real son más dispares, esto se debe precisamente a que los medios no son ideales, PLC es un medio hostil, por lo que la diferencia será mayor que en el caso de ZIV.

Una vez verificado el módulo standalone desarrollado, y habiendo analizado el flujo de trazas que intervienen en la solicitud de un informe de consumo (S02) es conveniente obtener dos parámetros importantes en el análisis de trazas en redes PRIME, MTU y WS, que como vimos en secciones previas no vienen fijados por el estándar PRIME.

La Tabla 8 ilustra los valores que toman estos parámetros teniendo en cuenta el fabricante. Estos datos son relevantes a la hora de realizar las simulaciones, ambos parámetros son pasados al simulador como datos de entrada.

Fabricante	WS	MTU
ZIV	6	47 Bytes
CIRCUTOR	4	71 Bytes

Tabla 8 Parámetros PRIME

5.2 Validación del simulador frente a un escenario real

El objetivo de esta segunda sección es validar si los resultados del simulador se ajustan a las medidas reales. Para ello, disponemos del archivo shapefile de un escenario real y de su correspondiente traza de tráfico.

Una vez validado el módulo software desarrollado en la sección previa, nos encontramos en condiciones de poder utilizarlo para validar los resultados obtenidos en el simulador. Para ello se ha tomado como referencia la topología de red real mostrada en la Figura 85, obtenida de un entorno residencial operativo, y de la cual se ha obtenido una captura de trazas real y el archivo shapefile asociado para poder validar los resultados obtenidos.



Figura 85 Escenario real utilizado para la validación de los resultados del simulador

Para la validación se han realizado simulaciones un número de veces lo suficientemente alto como para poder calcular intervalos de confianza al 95% utilizando para ello la ecuación mostrada a continuación, donde μ es la media de las muestras, σ la desviación típica y N el número de muestras:

$$IC = \mu \pm \left(1.96 \frac{\sigma}{\sqrt{N}} \right)$$

Esta misma fórmula se utiliza para calcular intervalos de confianza de los TTR_i calculados en la traza real medida en campo.

El intercambio de mensajes entre el CNTR y los contadores para Circutor quedó reflejado en la Figura 65.

Los valores de WS y MTU son esenciales para poder realizar las simulaciones con características similares al entorno real analizado.

La topología comprende 62 contadores inteligentes. La Tabla 9 muestra los resultados obtenidos mediante el análisis de trazas reales y su correspondiente simulación utilizando como parámetros de entrada los indicados anteriormente (ver Tabla 8).

WS = 4	Análisis de trazas	Simulación MTU Ajustada
N	62	4374
Media	13,5668709677	13,39230382
Mediana	13,062	12,71
Desviación Típica	4,208939684	1,479523905
IC Parametrico	[13,0068429098-14,1268990257]	[13,34845694-13,4361507]

Tabla 9 Estadísticos Relevantes Obtenidos en el análisis de los resultados de TTRi

Con los datos de la Tabla 9 podemos concluir que los resultados del simulador se ajustan a los medidos en campo puesto que los intervalos de confianza se solapan. A pesar de que para extraer conclusiones sólidas sería necesario realizar este mismo ejercicio para muchos escenarios más, esto queda fuera del ámbito de este proyecto, pero el resultado obtenido respalda la fiabilidad de SimPRIME en una primera aproximación.

Una vez validados los resultados, podemos intentar optimizar el funcionamiento de la red PRIME analizada. Para ello se ha procedido de la misma manera que en el caso anterior: se han realizado simulaciones un número de veces alto para poder obtener estadísticos relevantes.

En la Tabla 10 se han realizado simulaciones variando únicamente la MTU y estableciendo el WS a 1, siendo la columna “MTU Ajustada” las simulaciones asociadas al haber utilizado la MTU obtenida en el análisis previo como parámetro de entrada al simulador.

WS = 1	Análisis de trazas	Simulación MTU Ajustada	Simulación MTU Máxima
N	62	3563	6542
Media	13,5668709677	16,86137582	12,7390911
Mediana	13,062	15,7396	11,9698
Desviación Típica	4,208939684	3,664393555	2,558516533
IC Parametrico	[13,0068429-14,1268990]	[16,74059387-16,98215777]	[12,67709148-12,80109072]

Tabla 10 Tabla comparativa estadísticos TTRi con WS=1

En la Tabla 11 se han realizado simulaciones variando únicamente la MTU y estableciendo el WS a 4, parámetro obtenido en la sección previa, siendo la columna “MTU Ajustada” las simulaciones asociadas al haber utilizado la MTU obtenida en el análisis previo como parámetro de entrada al simulador.

WS = 4	Análisis de trazas	Simulación MTU Ajustada	Simulación MTU Máxima
N	62	4374	5077
Media	13,5668709677	13,39230382	11,3761143
Mediana	13,062	12,71	11,1339
Desviación Típica	4,208939684	1,479523905	0,685495527
IC Parametrico	[13,0068429-14,1268990]	[13,34845694-13,4361507]	[11,35725797-11,39497063]

Tabla 11 Tabla comparativa estadísticos TTRi con WS=4

Analizando la dependencia del TTR_i con respecto a la MTU, puede observarse como una disminución de la MTU, produce un incremento del TTR_i calculado, esto implica que el escenario simulado es un escenario con baja tasa de errores de transmisión. Esto se debe a que, en canales sin errores, aumentar la MTU reduce el *overhead*. Por lo tanto la MTU debe elegirse teniendo en cuenta las condiciones del canal.

Analizando la dependencia del TTR_i con respecto al WS, un incremento en el WS tiene como consecuencia un decremento del TTR_i . Esto sucede porque el simulador implementa el mecanismo de Rechazo Simple (como se indica en las especificaciones del PRIME), donde sólo los paquetes enviados tras el último paquete asentido son reenviados. Por lo tanto, incrementar el WS siempre se traduce en un mejor rendimiento de la red en términos de latencia.

Teniendo en cuenta todo lo anterior podemos categorizar el entorno como de bajas pérdidas, por lo que utilizar el valor de MTU máxima establecido por PRIME y el mayor tamaño de ventana configurable sería la que nos permitiría obtener el menor TTR_i , configuración óptima para la red analizada. Este análisis ilustra el potencial del simulador SimPRIME y el valor que puede aportar a las distribuidoras eléctricas, teniendo en cuenta que los resultados que arroja pueden considerarse como significativos en base a la validación contra un escenario real que se ha presentado en este mismo apartado.

Capítulo 6

Conclusiones y Trabajos Futuros

Este capítulo resume las principales conclusiones obtenidas tras la ejecución del presente Proyecto Fin de Carrera, además de una serie de posibles mejoras que se podrían realizar en el sistema en un futuro.

6.1 Conclusiones

En el presente Proyecto Fin de Carrera se ha desarrollado una solución *software* que permite obtener parámetros de redes PRIME a partir del análisis de trazas reales. Dicho *software* se ha integrado en la herramienta de análisis ya existente *PRIME Analytics* y se ha utilizado para validar los resultados obtenidos del simulador de redes PRIME SimPRIME. La métrica que se ha utilizado para ello ha sido el *TTRi* (tiempo que tarda un contador en enviar un informe S02 de consumo tras recibir una solicitud por parte del concentrador a tal efecto), habiéndose tenido que extender la funcionalidad de la aplicación web de SimPRIME de la que se partía para que calculase dicho parámetro.

Desde el punto de vista del diseño ha sido necesario adquirir amplios conocimientos, técnicos y funcionales de los protocolos PRIME y DLMS/COSEM, estudiar el modo de operación y conocer el formato de mensajes utilizado para la comunicación. Conocer el formato exacto de las trazas registradas en los equipos es muy importante ya que serán éstas las que nos proporcionarán los datos necesarios para desarrollar este PFC. Dicho formato no es estándar, por lo que decidimos soportar dos de los principales fabricantes del mercado (ZIV y CIRCUTOR).

Una vez claros los parámetros relevantes y la forma de obtenerlos de las trazas reales de ambos fabricantes, se elaboró un diseño del mismo, en el que se establecían los requisitos que debían cumplir y la forma de almacenarlos para su posterior procesado. Empezar la implementación habiendo hecho un diseño claro, facilitó mucho la tarea.

Una vez se definieron las subfases del proyecto, el siguiente paso fue desarrollarlo siguiendo lo más fielmente posible dicha definición. A continuación, el módulo desarrollado se validó adecuadamente para comprobar que los valores de *TTRi* calculados eran correctos. Asimismo, dicho módulo se ha utilizado para validar los resultados obtenidos en SimPRIME frente a los medidos en un escenario real comprobándose que coinciden, lo que supone el principal objetivo y la conclusión más relevante de este trabajo respectivamente.

Para realizar esta validación ha sido necesario extender la funcionalidad de la aplicación web de SimPRIME. Desde el punto de vista del diseño y la integración de la funcionalidad en la interfaz web, ha sido necesario conocer el funcionamiento de la misma, estudiar el formato de los resultados y aprovechar el desarrollo existente para incluir la nueva funcionalidad.

Como ya se ha mencionado, una vez se incluyó el cálculo del *TTRi* en la aplicación web de SimPRIME, se procedió a comparar los resultados obtenidos por ambas herramientas, comprobándose que los intervalos de confianza calculados se solapaban, lo que indica que los resultados del simulador se ajustan a las medidas tomadas en campo. A continuación, se procedió a realizar simulaciones variando la MTU y la WS, comprobándose que estos parámetros afectan al rendimiento de la red PRIME e ilustrándose el potencial del simulador y el valor que puede aportar a la industria, teniendo en cuenta que los resultados que arroja el simulador pueden considerarse como significativos en base a la validación contra un escenario real que se ha presentado en este PFC.

Cabe destacar que el trabajo ha sido desarrollado dentro del ámbito del proyecto de investigación nacional OSIRIS (Optimización de la Supervisión Inteligente de la Red de Distribución), financiado por el Ministerio de Economía y Competitividad y liderado por Unión Fenosa Distribución (tercera distribuidora eléctrica a nivel nacional).

En cuanto a mi visión personal, las principales conclusiones al respecto se detallan a continuación:

Un desafío importante ha sido el hecho de utilizar expresiones regulares para procesar cada tipo de traza estudiada, lo que me ha permitido adquirir mayor destreza en su utilización, a la que sin duda sacaré partido en mi proyección profesional. Adquirir conocimientos de los protocolos analizados junto con el formato de traza ha sido el hito más costoso durante el desarrollo.

Una vez superado el hito de poder analizar cada tipo de traza de forma individual proseguí con el diseño y puesta en marcha de un proyecto basado en el desarrollo de varios módulos software e incluirlos en herramientas ya desarrolladas. Esto ha supuesto todo un reto, ya que personalmente desde hace unos cuantos años me dedico profesionalmente a un campo alejado del desarrollo de software.

Con el fin de acelerar el desarrollo y despliegue de aplicaciones distribuidas he adquirido conocimientos sobre la herramienta Docker. Una vez configurada correctamente, la puesta en marcha de una aplicación resulta sencilla tal y como hemos podido ver con las herramientas PRIME Analytics y SimPrime. Docker se ha convertido en una herramienta cada vez más imprescindible en el mundo profesional ya que reduce considerablemente los costes de desarrollo y despliegue, por lo que adquirir conocimientos sobre la misma, seguro me será de utilidad.

Por último, teniendo en cuenta lo desarrollado en es PFC, el aspecto más relevante del análisis es indicar que el formato de las trazas establecido para redes PRIME no es estándar, lo que dificulta su procesamiento. Estandarizar el formato en que se capturan las trazas de tráfico sería un hito muy importante para facilitar el análisis de las mismas.

6.2 Trabajos futuros

El presente Proyecto Fin de Carrera ha proporcionado nueva funcionalidad tanto a la herramienta *PRIME Analytics* como al simulador SimPRIME.

La aplicación web original del simulador ha sido extendida para poder obtener y representar datos referentes al cálculo del *TTRi*, lo cual ha supuesto una mejora considerable para el simulador de redes SimPRIME ya que ha permitido validar los resultados obtenidos con los medidos en campo. No obstante, siguen existiendo posibilidades para seguir enriqueciendo esta aplicación, entre las que destacamos las siguientes:

- Convendría repetir la validación frente a muchos otros escenarios reales, de forma que se pudieran extraer conclusiones sólidas sobre la validez de los resultados de las simulaciones.
- Se podría incluir la representación gráfica y exportación de datos asociados al identificador único NID. En lo que atañe al presente PFC se han obtenido datos asociados a un identificador numérico incremental atendiendo al orden en el que el simulador reordena los contadores en la red PRIME simulada. Poder simular y obtener datos de un contador identificado en la red de forma unívoca ayudaría a detectar anomalías y corregir errores en un equipo físico concreto.
- También se podría incluir un módulo software adicional que permita procesar los datos obtenidos referentes al *TTRi* para realizar el cálculo de los estadísticos utilizados en este PFC de forma automática de tal manera que facilite investigaciones futuras.

La herramienta *PRIME Analytics* ha sido extendida para poder procesar trazas de dos de los principales fabricantes del mercado de las AMI y almacenar los datos obtenidos en formato estándar, lo cual ha supuesto una mejora considerable para obtener datos de redes reales ya que permite procesar las trazas sin prestar atención al formato. No obstante, para seguir enriqueciendo sería necesario tener en cuenta los siguientes aspectos:

- La funcionalidad de *PRIME Analytics* y del módulo *standalone* desarrollado podría ser extendida para procesar un mayor número de fabricantes, siempre y cuando *PRIME Alliance* no estandarice el formato de registro de las trazas.
- Por otra parte, al igual que en el caso del simulador SimPRIME sería necesario incluir un módulo *software* adicional que permita procesar los datos obtenidos referentes al *TTRi* para realizar el cálculo de los estadísticos utilizados en este PFC de forma automática de tal manera que facilite investigaciones futuras al mismo tiempo que se incluyen mejoras en la representación de los datos.

Referencias

1. Gregorio López, Francisco J. Herrera, José I. Moreno, Fernando Martín, Marta Bocos. Arquitectura de red para despliegues masivos de Infraestructuras de Medición Avanzada. *JITEL 2013*. 2013.
2. Seijo, Miguel, y otros. Herramientas TIC para la mejora del rendimiento y la planificación de redes PLC para Smart Grids. *JITEL 2015*. 2015.
3. Prime-Alliance. [En línea] [Citado el: 1 de 12 de 2016.] <http://www.prime-alliance.org/>.
4. Matanza, Javier. Improvements in the PLC Systems for Smart Grids Environments. *Tesis doctoral*. 10 2013.
5. Ferré, A. Aplicación web para la automatización y visualización de simuladores de redes PRIME. *Proyecto Fin de carrera. Universidad Carlos III de Madrid*. 2015.
6. Alberto, Sánchez Martínez. Desarrollo de un módulo de análisis de mapas shapefile para un simulador de redes PRIME. *Proyecto Fin de carrera. Universidad Carlos III de Madrid*. 2016.
7. OSIRIS. [En línea] <http://www.proyecto-osiris.com/es>.
8. D3.3, Entregable. Prototipo OSIRIS de plataforma de monitorización y emulación de redes de comunicaciones. *Proyecto OSIRIS*. 2015.
9. M.Sanz. Evaluación de técnicas de optimización aplicables a entornos AMI. *Trabajo Fin de Grado. Universidad Carlos III de Madrid*. 2016.
10. FERC - Federal Energy Regulatory Commission. [En línea] [Citado el: 1 de 12 de 2016.] <https://www.ferc.gov/EventCalendar/Files/20070423091937-EPRI%20-%20IntelliGrid.pdf>.
11. PRICE. [En línea] <http://www.futured.es/2012/05/22/price-proyecto-de-redes-inteligentes-en-el-corredor-de-henares/>.
12. Technology roadmap: Smart Grids. [En línea] International Energy Agency, 2011. [Citado el: 1 de 12 de 2016.] <http://www.iea.org/publications/freepublications/publication/smartgrid>.
13. *An evaluation of two-way communication means for advanced metering in Flanders (Belgium)*. Deconinck, G. Victoria, Vancouver Island, Canada : s.n., 2008. IEEE Int. Conf. on Instrumentation and Measurement Technology. págs. 900-905.
14. KEMA Consulting. *Smart Meter Requirements, Dutch Smart Meter specification and tender dossier*. 02 2008.
15. *Design of the overall system architecture, Tech. Rep. 3.1*. Ankou, A, Romero, G y Mauri, G. 02 02 2010. OPEN meter Consortium.
16. *Powerline Carrier - The basis for advanced metering*. Schaub, Thomas. Viena : s.n., 2007. 19th International Conference on Electricity Distribution.
17. Malaga smartcity. [En línea] <http://www.smartcitymalaga.es/>.
18. *Deployment of Power Line Communication by European Utilities in Advanced Metering Infrastructure*. Aidine, A, Tabone, A y Muller, J. Johannesburg, South Africa : s.n., 2013. IEEE ISPLC 2013.

19. *Narrowband orthogonal frequency division multiplexing power line communication transceivers for PRIME networks*. ITU-T standard G.9904 : s.n., 2012.
20. *G.hnem: the new itu-t standard on narrowband plc technology*. Oksman, V y Zhang, J. 12, 09 2011, Communications Magazine, IEEE, Vol. 49, págs. 36-44.
21. *A switch promotion algorithm for improving PRIME PLC network latency*. Alonso, E, y otros. 2014 doi:10.1109/ISPLC.2014.6812350. 18th IEEE International Symposium on Power Line Communications. págs. 278-283.
22. *Narrowband PLC for LV Smart Grid Services*. Sendin, A, y otros. 2014 doi:10.1109/ISPLC.2014.6812376. 18th IEEE International Symposium on Power Line Communications. págs. 168-172.
23. DLMS User Association. Excerpt from Compation Specification for Energy Metering - Architecture and Protocols. [En línea] 2014. http://dlms.com/documents/Excerpt_GB8.pdf.
24. —. *Excerpt from COSEM - Identification System and Interface Classes*. 2010.
25. Union Fenosa Distribución (LINTER). [En línea] [Citado el: 1 de 12 de 2016.] <http://www.unionfenosadistribucion.com/es/redes+inteligentes/investigacion+y+desarrollo/1297137260045/conozca+nuestro+laboratorio.html>.
26. Riverbed modeler. [En línea] <http://www.riverbed.com/products/steelcentral/steelcentral-riverbed-modeler.html>.
27. NS-3. [En línea] [Citado el: 2015 de 10 de 06.] <https://www.nsnam.org/overview/what-is-ns-3/>.
28. *An overview of the OMNeT++ simulation environment*. Varga, A y Horning, R. Bruselas : s.n., 2008. Proceedings of the Simutools08 1st International Conference on Simulation tools and Techniques for Communications, Networks and Systems & Workshops.
29. M. Seijo, G. López, J.I. Moreno, J. Matanza, S. Alexandres, C. Rodríguez-Morcillo. Online Evaluation and Planning Tool for Medium Voltage-Broadband over Power Line Cells. *IEEE International Energy Conference (ENERGYCON)*. 2016.
30. Lakasta, E. *Aplicación web para automatización y visualización de simulaciones de celdas MV-BPL*. Proyecto Fin de Carrera. Universidad Carlos III de Madrid : s.n., 2016.
31. Simulador PLC basado en NS3. [En línea] http://www.ece.ubc.ca/~faribaa/User_Guide.pdf.
32. M.-S. Kim, D.-M. Son, Y.-B. Ko, Y.-H. Kim. *A simulation study of the plc-mac performance using network simulator-2*. s.l. : IEEE Symposium on Power Line Communications and Its Applications (ISPLC), 2008. doi:10.1109/ISPLC.2008.4510406..
33. A. Sanz, P. Piero, D. Montoro, J. Garcia. High-accuracy distributed simulation environment for PRIME networks analysis and improvement. *IEEE Symposium on Power Line Communications and Its Applications (ISPLC)*. doi:10.1109/ISPLC.2012.6201298, 2012.
34. A. Gogic, A. Mahmutbegovic, D. Borovina, I. Cavdar, N. Suljanovic. Simulation of the narrow-band plc system implementing PRIME standard. *IEEE Energy Conference (ENERGYCON)*. doi:10.1109/ENERGYCON.2014.6850624, 2014.
35. J. Larrañaga, J. Legarda, I. Urrutia, A. Sendin. An experimentally validated PRIME subnetwork simulation model for utility applications. *IEEE Symposium on Power Line Communications and Its Applications (ISPLC)*. 2015.
36. Sistemas de Información Geográfica, tipos y aplicaciones empresariales. [En línea] <http://sig.cea.es/SIG>.
37. *A channel model for the residential power circuit used as a digital communications medium*. Hooijen, O. G. 4, 1998, IEEE Transactions on Electromagnetic Compatibility, Vol. 40, págs. 331-336.
38. What is Python? Executive Summary | Python.org. [En línea] [Citado el: 17 de 12 de 2016.] <https://www.python.org/doc/essays/blurb/>.
39. About the Django Software Foundation | Django. [En línea] [Citado el: 17 de 10 de 2016.] <https://www.djangoproject.com/foundation/>.

40. Django Overview | Django. [En línea] [Citado el: 17 de 10 de 2016.]
<https://www.djangoproject.com/start/overview/>.
41. Django appears to be a MVC framework, but you call the Controller the “view”, and the View the “template”. How come you don’t use the standard names? | FAQ: General | Django documentation | Django. [En línea] [Citado el: 17 de 10 de 2016.]
<https://docs.djangoproject.com/en/1.8/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-view-the-template-how-come-you-don-t-use-the-standard-names>.
42. Celery 4.0.1 documentation. [En línea] [Citado el: 17 de 10 de 2016.]
<http://docs.celeryproject.org/en/latest/>.
43. Introduction to Celery - Celery 4.0.1 documentation . [En línea] [Citado el: 17 de 10 de 2016.] <http://docs.celeryproject.org/en/latest/getting-started/introduction.html>.
44. Interactive JavaScript charts for your webpage | Highcharts. [En línea] [Citado el: 17 de 10 de 2016.] <http://www.highcharts.com/>.
45. Google Charts. [En línea] [Citado el: 1 de 12 de 2016.]
<https://developers.google.com/chart/>.
46. Google Chart Galery. [En línea] [Citado el: 1 de 12 de 2016.]
<https://developers.google.com/chart/interactive/docs/gallery>.
47. RabbitMQ - Messaging that just works. [En línea] [Citado el: 13 de 10 de 2016.]
<http://www.rabbitmq.com/>.
48. PostgreSQL: About. [En línea] [Citado el: 17 de 10 de 2016.]
<http://www.postgresql.org/about/>.
49. Docker. [En línea] <https://www.docker.com/what-docker>.

ANEXO I

Presupuesto

Este anexo presenta la planificación y fases de desarrollo del Proyecto Fin de Carrera y el presupuesto del mismo, desglosándolo en costes tangibles e intangibles y en costes de personal.

I.1 Planificación

El presente Proyecto Fin de Carrera se ha dividido en las siguientes tareas:

MEJORA Y VALIDACIÓN DE UN SIMULADOR DE REDES PRIME

- Análisis
 - Adquisición de conocimiento (PRIME, DLMS/COSEM)
 - Estudio Formato Trazas ZIV y CIRCUTOR
- Diseño y Desarrollo Módulo StandAlone
 - Diseño
 - Desarrollo
 - Validación
- Diseño y Desarrollo Modulo Prime Analytics
 - Adquisición de conocimiento
 - Integración módulo Cálculo TTRi
 - Validación
- Diseño y Desarrollo Modulo SimPrime
 - Adquisición de conocimiento
 - Integración módulo Cálculo TTRi
- Validación
 - Validación Simulador SimPrime
- Documentación
 - Memoria

Las Figuras 86 y 87 ilustran la ejecución del mismo.

Nombre de la tarea	Fecha de inicio	Fecha de final	Duración
MEJORA Y VALIDACIÓN DE UN SIMULADOR DE REDES PRIME	12/09/16	09/12/16	65d
Análisis	12/09/16	19/09/16	6d
Adquisición de conocimiento (PRIME, DLMS/COSEM)	12/09/16	14/09/16	3d
Estudio Formato Trazas ZIV y CIRCUTOR	15/09/16	19/09/16	3d
Diseño y Desarrollo Módulo StandAlone	20/09/16	07/10/16	14d
Diseño	20/09/16	21/09/16	2d
Desarrollo	22/09/16	03/10/16	8d
Validación	04/10/16	07/10/16	4d
Diseño y Desarrollo Modulo Prime Analytics	10/10/16	21/10/16	10d
Adquisición de conocimiento	10/10/16	12/10/16	3d
Integración módulo Cálculo TTRi	13/10/16	18/10/16	4d
Validación	19/10/16	21/10/16	3d
Diseño y Desarrollo Modulo PrimeSinPrime	21/10/16	01/11/16	8d
Adquisición de conocimiento	21/10/16	25/10/16	3d
Integración módulo Cálculo TTRi	26/10/16	01/11/16	5d
Validación	02/11/16	18/11/16	13d
Validación Simulador SimPrime	02/11/16	18/11/16	13d
Documentación	21/11/16	09/12/16	15d
Memoria	21/11/16	09/12/16	15d

Figura 86 Resumen de tareas

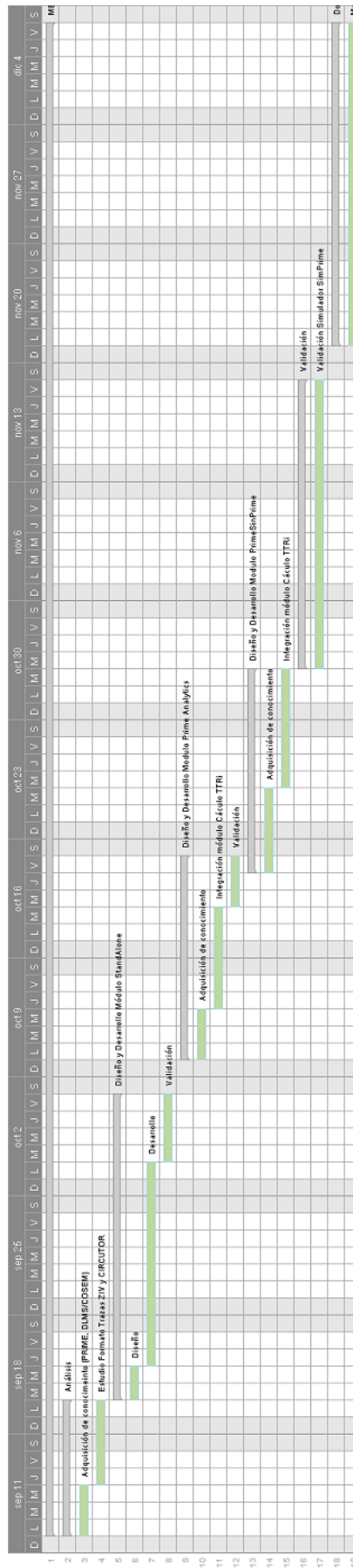


Figura 87 Diagrama de Gantt

I.2 Recursos y Costes

I.2.1 Recursos tangibles e intangibles

Para la realización del presente Proyecto Fin de Carrera, se ha requerido del uso de los siguientes recursos:

- Ordenador clónico con un coste de 1000€ (incluye monitor y equipamiento de red).

Como el ordenador tiene un tiempo de amortización esperado de 5 años, y sólo se ha utilizado para el proyecto durante una fracción de ese periodo, su coste debe ser amortizado. También debe tenerse en cuenta que durante la realización del proyecto, el ordenador no estaba dedicado en exclusiva al mismo, por lo que debe imputársele sólo el 100% del tiempo.

$$\text{Amortización} = \frac{\frac{1000 \text{ €}}{5}}{12 \text{ meses}} \times 4 \text{ meses} \cdot 0.5 = 33.33 \text{ €}$$

Elemento	Coste unitario	Cantidad	Coste Total
Ordenador(meses)	8,33	4	33,33 €

Tabla 12 Coste de los recursos tangibles e intangibles.

Las siguientes licencias académicas de *software* han sido proporcionadas de forma gratuita por los distintos fabricantes, siendo el resto del *software* utilizado libre y gratuito:

- Windows 10
- Office 2010
- Lucidchart
- smartsheet

Por tanto el coste total de los recursos tangibles e intangibles ha sido de 48,01€ (cuarenta y ocho euros con un céntimo).

I.2.2 Recursos humanos

A efectos presupuestarios, la cantidad de trabajo se ha medido en días, sumando un total de 65 días laborables con una jornada laboral de 8 horas. El coste por persona y hora establecido por la Universidad Carlos III de Madrid para un ingeniero es de 35€ la hora, siendo este un “coste de empresa” (el salario bruto del empleado más un 35% aproximadamente en impuestos y otros gastos).

Por tanto, el coste total de los recursos humanos ha sido de dieciocho mil cuatrocientos ochenta euros.

Tarea	Coste / Trabajo	Trabajo	Horas	Coste
Análisis	35 € / hora	6 días	48	1680 €
Diseño y Desarrollo Módulo StandAlone		14 días	112	3920 €
Diseño y Desarrollo Modulo Prime Analytics		10 días	80	2800 €
Diseño y Desarrollo Modulo SinPrime		8 días	64	2240 €
Validación		13 días	104	3640 €
Documentación		15 días	120	4200 €
TOTAL			65 días	528

Tabla 13 Coste de recursos humanos.

I.2.3 Presupuesto total

El presupuesto final del proyecto incluye ambos tipos de gastos, por lo que el coste total asciende a dieciocho mil quinientos trece euros con treinta y tres céntimos.

$$\text{Coste total} = 33.33 \text{ €} + 15.840,00 \text{ €} = 18513.33 \text{ €}$$