Universidad del País Vasco

Euskal Herriko Unibertsitatea

Escuela Técnica Superior de Ingeniería de Bilbao

Departamento de Tecnología Electrónica

TESIS DOCTORAL

Arquitecturas System-on-Chip para Cyber Physical System Gateway en Smart Grid

Autor: Wilmer Marcelo Urbina Gamboa

(wmurbina@espe.edu.ec)

Director: Dr. Jesús Lázaro Arrotegui

Dr. Armando Astarloa Cuellar

Bilbao, Mayo de 2019

A mi esposa y a mi hijo

Agradecimientos

En primer lugar quiero agradecer a mis directores de tesis, Jesús Lázaro y Armando Astarloa, por su valiosa guía y asesoramiento durante toda la investigación. Al grupo de investigación APERT y a la empresa SoC-e por haber confiado en mí y haberme dado la oportunidad de realizar esta tesis.

Gracias a mi esposa Tatiana y mi hijo Emiliano por su cariño y apoyo incondicional durante todo este proceso. A mi hermana Wendy por siempre desear y anhelar lo mejor para mi. A mis padres por cada consejo y por cada una de sus palabras que me guiaron durante mi vida. A mis suegros y toda la familia por ayudarnos siempre con todo y estar pendientes en cada momento.

Gracias a todos los profesores e investigadores del grupo APERT, por que siempre estuvieron ahí para darme una mano cuando lo necesitaba. Gracias a todas las personas que me apoyaron y creyeron en la realización de esta tesis.

Gracias, a la Universidad de las Fuerzas Armadas ESPE por su apoyo y confianza al financiar esta tesis mediante su Programa de becas y ayudas económicas para Estudios de Doctorado.

Gracias a la vida por este nuevo triunfo.

Resumen

La forma en que funciona la red eléctrica no ha cambiado mucho desde su creación en la década de 1930. En general, los métodos y medios de transmisión de los datos actuales siguen siendo similares a los de principios de la década de 1930. Aunque la infraestructura general permanece inalterada, algunas tecnologías han cambiado desde entonces, y el ritmo de este cambio ha aumentado significativamente en la última década. Por ejemplo, la introducción de las Tecnologías de la Información y las Comunicaciones *(TICs)* en la operación de las redes eléctricas ha dado como resultado la denominada *Smart Grid*. En términos generales, el sistema eléctrico actual consiste en una compleja red en la que están interconectadas las centrales de generación eléctrica, la infraestructura de transporte de electricidad, la infraestructura de distribución, y la carga.

Desde un punto de vista tecnológico, la *Smart Grid* puede ser vista como una superposición de una red de comunicación sobre la red eléctrica. La red de comunicaciones de la *Smart Grid* requiere tecnologías de comunicaciones capaces de proporcionar servicios avanzados, como el envío de datos de sensores en tiempo real, la redundancia y la ciber-seguridad. Se implementa utilizando una variedad de redes y medios de comunicación, incluyendo el mismo cableado eléctrico, redes inalámbricas y otras infraestructuras de comunicaciones existente, como las redes *Ethernet* basadas en cables de cobre o fibra óptica. Existen ventajas y desventajas asociadas a cada opción y en la realidad los tres enfoque se combinan para conformar las comunicaciones en la *Smart Grid*. Como resultado, de esta integración, los equipos utilizados para gestionar las comunicaciones son completamente heterogéneos. Por ello, desde una perspectiva global que favorezca la interoperabilidad, es imprescindible disponer de dispositivos de comunicaciones que combinen requisitos de procesamiento en tiempo real, sincronización avanzada, alta disponibilidad en las comunicaciones, reconfigurabilidad y ciber-seguridad. Este concepto se conoce comúnmente como *Cyber Physical System (CPS)*.

A modo de resumen, un *CPS* típico se compone de varios dispositivos conectados a través de redes cableadas e inalámbricas. Estos dispositivos abarcan desde plataformas embebidas, sistemas en tiempo real, sensores y actuadores, hasta dispositivos en red. Por lo tanto, los *CPS* se benefician de los continuos desarrollos de nuevas plataformas de computación y sensórica de bajo coste, las comunicaciones inalámbricas, las redes de comunicación de gran ancho de banda y sistemas que permiten realizar una gestión más eficiente de la energía de los dispositivos.

La propuesta de investigación presentada en esta tesis busca realizar contribuciones en el campo de los sistemas embebidos, planteando una arquitectura común de nodos que sirva como referencia de arquitectura *CPS* para la *Smart Grid*. Esta arquitectura deberá dar solución a la integración directa de los nodos en la red, permitiendo a su vez procesamiento en tiempo real, necesario en ciertas secciones y operaciones de la *Smart Grid*.

En primer lugar, se presentará una visión general de la red eléctrica actual *(Smart Grid)*. En particular, se describirán los elementos fundamentales de una subestación, y se presentarán los estándares de comunicación utilizados para garantizar y satisfacer los requisitos de interoperatividad que deben cumplir las redes de transmisión y distribución modernas. A continuación, se describirán los requisitos y las características de funcionamiento que debe cumplir un dispositivo *CPS Gateway* para poder ser utilizado en la red eléctrica inteligente. Por otra parte, se definirá un *CPS* y se describirán sus partes, características y campos de aplicación. A continuación, se realizará un estudio detallado de varias arquitecturas existentes que representan ventajas significativas para su utilización en la *Smart Grid*. En segundo lugar, se propondrán arquitecturas *CPS Gateway* sobre plataformas reconfigurables *System-on-Chip* que garanticen procesamiento en tiempo real, necesario en ciertas secciones y operaciones de la *Smart Grid*. También, deberá incorporar mecanismos avanzados de sincronización, comunicaciones de alta disponibilidad mediante comunicaciones redundantes, compatibilidad con la infraestructura de automatización de subestaciones actualmente en fase de despliegue *(IEC 61850)* y ciber-seguridad para las tramas *SV* y *GOOSE*.

Para finalizar, se validarán las arquitecturas propuestas en dispositivos *SoC* reconfigurables. En este sentido, se implementarán tres arquitecturas para verificar el funcionamiento del *CPS Gateway*. La primera arquitectura, tendrá como finalidad validar los requisitos de sincronización, interoperabilidad y alta disponibilidad. En la segunda arquitectura se implementará un protocolo y un módulo de comunicaciones que permita la configuración remota del *CPS Gateway*. Finalmente en la tercera arquitectura se propondrá el uso de cifrado simétrico como mecanismo de ciber-seguridad para las tramas *SV* y *GOOSE*.

Índice

Lista de Figuras

Lista de Tablas

Lista de Acrónimos

ACSI	Abstract Communication Service Interface
AE	Authenticated Encryption
AEAD	Encryption Algorithm that allow Additional Data authentication
AES	Advanced Encryption Standard
AP-SoC	All Programmable SoC
ASIC	Application Specific Integrated Circuit
AXI	Advanced eXtensible Interface
CIM	Common Information Model
CIS	Component Interface Specifications
COE	Configuration Over Ethernet
COEsec	Configuration Over Ethernet Secure
CPS	Cyber-Physical System
DANHs	Doubly Attached Bridging Nodes HSR
DANs	Dual Attached Nodes
DAS	Data Acquisition Systems
DDS	Data Distribution Service
DER	Distributed Energy Resources
DNP3	Distributed Network Protocol version 3

EMR	Electronic Medical Records
EMS-API	Energy Management System - Application Program Interface
EPA	Enhanced Performance Architecture
EPRI	Electric Power and Research Institute
FPGA	Field Programmable Gate Array
FSBL	First Stage Boot Loader
GOOSE	Generic Object Oriented Substation Events
HMAC	Hash-based Message Authentication Code
HSR	High-availability Seamless Redundancy
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Devices
IIoT	Industrial Internet of Things
IRIG-B	Inter-Range Instrumentation Group
ISO	International Standards Organization
LAN	Local Area Network
LRE	Link Redundancy Entity
M2M	Machine-to-Machine
MCD	Módulo de Cifrado y Descifrado
MMS	Manufacturing Message Specification
MpSoC	Multiprocessor System-on-Chip
MRS	Managed Redundant Switch
MTU	Master Terminal Unit
NIC	Network Interface Cards
NITRD	National Information Technology Research and Development
NREC	North American Equipment Council

NSM	Network and System Management
NTP	Network Time Protocol
OSI	Open System Interconnect
PL	Programmable Logic
PMU	Phasor Measurement Units
PRP	Parallel Redundancy Protocol
PS	Processing System
PTP	Precision Time Protocol
QoS	Quality of Service
RBAC	Role-Based Access Control
RedBox	Redundancy Boxes
RSA	Rivest, Shamir y Adleman
RSTP	Rapid Spanning Tree Protocol
RTC	Real Time Clock
RTU	Remote Terminal Units
SANs	Single Attached Nodes
SAS	Substation Automation System
SCADA	Supervisory Control And Data Acquisition
SCL	Substation Configuration Language
SCP	Smart Connected Products
SCSM	Specific Communication Service Mapping
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SoC	System-on-Chip

SV	Sample Values
TCP/IP	Transmission Control Protocol/Internet Protocol
TIC	Tecnologías de la Información y la Comunicación
TLS	Transport Layer Security
UCA	Utility Communication Architecture
UML	Unified Modeling Language
VDAN	Virtual-DAN
XML	Extensible Markup Language
ZPL	Zero Packet Loss

Capítulo 1

Introducción

1.1. Planteamiento del problema

La forma de operación de la red eléctrica no ha cambiado mucho desde su invención en la década de 1930, los métodos y medios de transmisión de los datos actuales siguen manteniendo una similitud con los utilizados en sus inicios. A pesar de que la infraestructura general sigue siendo la misma, algunas tecnologías han cambiado desde entonces, y el ritmo de cambio ha aumentado considerablemente en las últimas décadas [1]. Por ejemplo, la introducción de las Tecnologías de la Información y la Comunicación *(TIC)* en la operación las redes eléctricas ha dado origen a una red compleja denominada *Smart Grid*, en la que se encuentran interconectadas las redes de generación, transporte y distribución de energía [2].

Desde un punto de vista tecnológico, la *Smart Grid* puede ser vista como una superposición de una red de comunicación sobre la red eléctrica. La red de comunicaciones en la *Smart Grid* consistiría en un tipo de red que permita ofrecer servicios de alto nivel, como el envío de datos de sensores en tiempo real, redundancia y ciber-seguridad. Se implementaría utilizando varias tecnologías de redes y medios de comunicación, entre los que se incluyen el mismo cableado eléctrico, las redes inalámbricas y otras infraestructuras de comunicación existentes como las redes *Ethernet* basadas en cobre o fibra óptica. Hay ventajas y desventa-

jas asociadas con cada una de las opciones y es probable que los tres enfoques se puedan utilizar para las comunicaciones en la *Smart Grid*. Para determinar que tecnología utilizar es importante evaluar la infraestructura en varios factores clave, como por ejemplo: ancho de banda, latencia, seguridad y confiabilidad. Las tecnologías de red tiene propiedades estadísticas diferentes para cada uno de los factores antes mencionados, y deben ser integrados cuidadosamente para satisfacer los niveles de confiabilidad que la *Smart Grid* necesita.

La *Smart Grid* evoluciona constantemente, impulsada por la búsqueda de una mayor eficiencia, por el crecimiento de la generación de energía distribuida y por la creciente automatización del sistema, incluida la medición del consumo por parte de los usuarios. Como resultado de estos avances, las redes involucradas eventualmente se integran en el sistema, lo que obliga a que el equipo utilizado para gestionar las comunicaciones sea completamente heterogéneo. Por lo tanto, desde una perspectiva global que favorezca la interoperabilidad, es necesario contar con dispositivos que combinen requisitos de procesamiento en tiempo real, sincronización avanzada, comunicaciones de alta disponibilidad y ciber-seguridad. Estos dispositivos se conocen comúnmente como *Cyber Physical System (CPS).*

El término *CPS* fue acuñado por científicos de diferentes disciplinas, principalmente de sistemas de tiempo real, redes de comunicación y sistemas de control, para describir aquellos sistemas (*Systems*) que requieren conectar el mundo de los computadores (*Cyber*) con el mundo físico (*Physical*). Más concretamente, los *CPS* permiten integrar computación y procesos físicos [3, 4].

Cabe señalar que estos sistemas son cada vez más frecuentes y relevantes en la actualidad en gran parte debido a las necesidades de utilizar cada vez más información en el control de los procesos industriales. Para el caso particular de los *CPS* en el sector industrial, se puede encontrar en [5] un interesante análisis de la evolución y de sus necesidades de integración. Conviene señalar que los nuevos avances y la reducción de costes en el dominio de las tecnologías de la información y la computación hacen posible la construcción de este tipo de sistemas en la actualidad.

Sin embargo, la construcción de este tipo de sistemas no es fácil. Algunos de sus problemas más apremiantes vienen determinados por su naturaleza, ya que, por ejemplo, la interacción con procesos físicos hace que el tiempo en el que se ejecutan los algoritmos sea relevante, tratándose, además, de sistemas intrínsecamente concurrentes. Por otro lado, la mayoría de los *CPS* requieren la integración de diferentes tecnologías utilizadas en diversos campos como la informática, la

comunicación y el control, incluyendo algunas de las últimas tecnologías y metodologías disponibles en el mercado, como las redes inalámbricas, computación distribuida, sistemas en tiempo real, la seguridad y la ingeniería orientada a modelos.

Un *CPS* típico se compone de varios dispositivos conectados a través de redes cableadas e inalámbricas. Estos dispositivos incluyen plataformas empotradas, sistemas de tiempo real, sensores y actuadores así como dispositivos de red. Por tanto, estos sistemas se benefician de los continuos avances y convergencia en estos campos. De hecho, los *CPS* están siendo impulsados por la proliferación de: (1) nuevas plataformas de computación y sensórica de bajo coste, pequeño tamaño y avanzadas prestaciones; (2) las comunicaciones inalámbricas; (3) redes de comunicación de gran ancho de banda; y (4) sistemas que permiten realizar una gestión más eficiente de la energía de los dispositivos.

En este escenario, los diseñadores de *CPS* deben satisfacer simultáneamente unos requisitos que han sido analizados en [3,4]. Entre otros retos se debe dar respuesta a:

1. La integración eficiente de plataformas y tecnologías de comunicación heterogéneas.
2. La gestión eficiente de los recursos de los sistemas, p.e. capacidad de cómputo, uso de la energía, ancho de banda, etc.
3. Las restricciones temporales de las aplicaciones.
4. La capacidad de reconfigurarse y adaptarse a nuevas situaciones.
5. Tolerancia a fallos y robustez del sistema completo frente a fallos ocurridos en los dispositivos individuales.
6. La seguridad frente a ciber-ataques.
7. La integración horizontal y vertical de la información con otros sistemas o subsistemas.

Con respecto a las comunicaciones, algunos autores [3] proponen la adopción de diseños radicalmente nuevos que se ajusten a los requisitos específicos de los *CPSs*. Por el contrario, otros autores [6] recomiendan usar enfoques más prácticos, al menos en el corto y medio plazo. En particular, [6] propone el uso de tecnologías

estándar de uso amplio en el dominio de los *CPS* como los protocolos *TCP/IP* y redes *IEEE802.3 (Ethernet)* [7] y *IEEE802.11 (WiFi)* [8]. Aunque estos estándares son relativamente pobres en términos de calidad de servicio *(Quality of Service, QoS)* se han propuesto algunas modificaciones que resuelven algunos problemas sobre *TCP/IP* que constituyen una nueva capa, la capa *Cyber-Physical Layer* [9].

Está comprobado que el uso de *middleware* en aplicaciones distribuidas complejas produce una reducción de hasta el 50 % en los costes de desarrollo del *software* [10]. En un primer acercamiento, el planteamiento anterior permitiría utilizar tecnologías *middleware* estándar para construir *CPSs* sobre la pila de protocolos de TCP/IP, tales como *J2EE, .NET, CORBA* [11], *ICE* [12], *Data Distribution Service (DDS)* [13], Servicios *Web* [14], *IEC 61850* para estandarización en entornos de automatización de subestaciones eléctricas y *OPC* en entornos industriales [15]. Sin embargo, estas tecnologías de *middleware* requieren mejoras para adaptarse a las necesidades específicas de los *CPSs* [9, 16, 17] ya que presentan dos inconvenientes principales: (1) tienden a ser una fuente de sobrecarga [16] y (2) no se ajustan a los requisitos específicos (funcionales y no funcionales) de los *CPS* [6], por lo que los programadores tendrían que resolver repetidamente los mismos problemas o similares.

Lo anteriormente expuesto conduce a la necesidad de una metodología para el diseño y una arquitectura generalizable de los sistemas embebidos, que permitan esconder los detalles de implementación de bajo nivel facilitando la construcción de nuevas aplicaciones. Algunos ejemplos se pueden encontrar en [15, 18] donde se proponen una metodología y un conjunto de abstracciones que facilitan el uso de un *middleware* de distribución de alto rendimiento como *DDS* [19], este *middleware* está orientado para el desarrollo de aplicaciones de comunicación de datos en el entorno industrial y en las subestaciones eléctricas.

Otro aspecto clave que hay considerar en el diseño de *CPS* es la ciber-seguridad. En la *Smart Grid* las redes de comunicación junto con la ciber-seguridad son partes críticas en el despliegue de Sistemas de Automatización de Subestaciones *(SAS)* fiables y eficientes. Este problema de seguridad es muy complejo y necesita ser enfrentado desde un enfoque multicapa: dispositivos, sistemas, redes, usuarios, las aplicaciones de *software*, etc. En los últimos diez años, la *International Electrotechnical Commission (IEC)* ha realizado un gran esfuerzo en relación con la ciber-seguridad en la industria de servicios eléctricos [20]. El estándar *IEC 62351* plantea normas para resolver problemas de seguridad en las diferentes áreas de operación y comunicación del sector eléctrico, que fueron definidas por el grupo

de trabajo *IEC TC57* [21]. El grupo *IEC TC57* se ha centrado en los protocolos y aplicaciones específicas utilizadas en *SAS*. En particular, en los estándares *IEC 61850* [22] y *IEC 62351-6* [23].

En el estándar *IEC 62351-6* se especifican los mecanismos de seguridad para los sistemas de comunicación que se definen en el estándar *IEC 61850* [24] que no están basadas en *TCP/IP*. Además, especifica la protección de las tramas *Generic Object Oriented Substation Events (GOOSE)* y *Sample Values (SV)* con códigos de autenticación mediante el protocolo *Secure Hash Algorithm (SHA)*, que son firmados digitalmente utilizando el sistema de encriptación de clave publica *Rivest, Shamir y Adleman (RSA)* para proporcionar la autenticidad de origen. Sin embargo, el método de firma digital *RSA* tienen tiempos de ejecución largos que no permiten satisfacer los requisitos de tiempo que establece el estándar *IEC 61850* [25]. A pesar de que en las pruebas para validar el método de encriptación *RSA* se hayan utilizado procesadores *ARM* de gama alta con un *crypto* acelerador, la firma *RSA* con claves de 1024 bits no puede ser calculada y verificada dentro de los $3\,ms$, que es el tiempo máximo de transferencia requerida por algunos mensajes *GOOSE* [26].

En este sentido, está previsto para el 2020 [23] que el estándar *IEC 62351-6* se actualice, y se propone la utilización de criptografía simétrica en lugar de firmas digitales. El uso de criptografía simétrica tiene como finalidad minimizar el impacto negativo que tiene las medidas de seguridad sobre el rendimiento de los dispositivos de campo.

La propuesta de investigación presentada en esta tesis busca realizar contribuciones en el campo de los sistemas embebidos, planteando una arquitectura común de nodos que sirva como referencia de arquitectura *CPS* para la *Smart Grid*. Esta arquitectura deberá dar solución a la integración directa de los nodos en la red, permitiendo a su vez procesamiento en tiempo real, necesario en ciertas secciones y operaciones de la *Smart Grid*. Deberá incorporar mecanismos avanzados de sincronización, comunicaciones de alta disponibilidad normalmente mediante comunicaciones redundantes, compatibilidad con la infraestructura de automatización de subestaciones actualmente en fase de despliegue *(IEC 61850)* y se propone como mecanismo de ciber-seguridad el uso de cifrado *(AES- GCM)* a nivel de Capa 2, para ofrecer seguridad a las tramas *SV* y *GOOSE*. Las ventajas tecnológicas que las *Field Programmable Gate Arrays (FPGAs)* ofrecen en la actualidad, permiten desarrollar arquitecturas como la planteada en esta tesis. Las *FPGAs* ha pasado de ser una simple herramienta para la creación de prototipos a ser una solución esencial para el desarrollo de dispositivos que requieren altas

capacidades de procesamiento, requisitos de operación en tiempo real, interoperabilidad, flexibilidad, seguridad y alta disponibilidad. Por otro lado, las *FPGAs* actuales, además de los grandes recursos que integran (millones de celdas lógicas, varios tipos de memoria e interfaces periféricas), también integran procesadores *ARM* implementados en silicio que permiten ejecutar desde simples aplicaciones especificas hasta complejos sistemas operativos. Para el desarrollo de esta tesis se aprovechó la experiencia del grupo de investigación en el diseño de sistemas electrónicos *System-on-Chip (SoC)*, *Multiprocessor System-on-Chip (MPSoC)*, el conocimiento en comunicaciones industriales y la implementación de algoritmos criptográficos, con el objetivo de validar la arquitectura *SoC* de un *CPS Gateway* utilizando *FPGAs* de *Xilinx*.

1.2. Objetivos

El objetivo general de este trabajo es proponer una arquitectura de referencia de un *Cyber Physical System Gateway (CPS Gateway)* para *Smart Grid* que pueda ser implementada en plataformas *SoC*, con capacidad de operación en tiempo real, mecanismos avanzados de sincronización, comunicaciones de alta disponibilidad, análisis complejo de datos y ciber-seguridad.

Objetivos específicos

Para poder articular el objetivo general es necesario definir objetivos específicos que permitan secuenciar la investigación propuesta.

Los objetivos específicos propuestos son:

1. Estudiar el estado del arte de la tecnología de forma detallada: *Smart Grid* arquitectura y protocolos de comunicaciones estandarizados. *CPS* arquitectura, características y aplicaciones.
2. Identificar los requerimientos de operación de los dispositivos utilizados en las redes de comunicaciones del sector eléctrico, considerando las recomendaciones planteadas por los estándares involucrados en la *Smart Grid*. En este estudio también se analizan los mecanismos y tecnologías existentes para dar solución a los requerimientos de operación identificados previamente.

3. Proponer una arquitectura de un *CPS Gateway (hardware*, *software* e interfaz de comunicación) basada en plataformas *SoC*, que de solución a todos los requerimientos de operación que exige la *Smart Grid*.

4. Diseñar varias implementaciones basadas en *FPGAs* y diferentes configuraciones de prueba para validar y demostrar el funcionamiento de la arquitectura propuesta.

1.3. Organización

La presente tesis consta de seis capítulos agrupados en tres partes. Además de este capítulo introductorio, el contenido del documento está dividido de la siguiente forma:

- **Estado del arte.** El capítulo 2 ofrece una visión general de la red eléctrica actual *(Smart Grid)*, se describen los elementos fundamentales de una subestación, y se presentan los estándares de comunicación utilizados para garantizar y satisfacer los requisitos de interoperatividad que deben cumplir las redes de transmisión y distribución modernas. Después de resumir la información encontrada en la literatura sobre los estándares de comunicaciones involucrados en la *Smart Grid*. En el capítulo 3 se describen los requisitos y características que debe cumplir un dispositivo *(CPS Gateway)* para ser utilizado en la *Smart Grid*. En el capítulo 4 se introducen y se definen los *CPS* y se realiza una descripción de sus partes, características y los campos de aplicación. Para finalizar el capítulo 4 se identifican y describen varias arquitecturas de dispositivos inteligentes aplicados al ámbito de las redes eléctricas, y se realiza una comparación en base a los requisitos identificados en el capítulo 3.

- **Contribución.** Basándonos en mecanismos y tecnologías existentes, en el capítulo 5 se desarrollan y proponen arquitecturas de dispositivos de comunicación *(CPS Gateway)* que permitan satisfacer las necesidades operativas identificadas en el capítulo 3, entre las que podemos mencionar: Procesamiento y operación de datos en tiempo real, interoperatividad, alta disponibilidad, reconfigurabilidad y ciber-seguridad.

- **Validación.** En el capítulo 6 se demuestra la viabilidad de implementar las arquitecturas *CPS Gateway* propuestas en las plataformas *SoC*. En la

primera parte del capítulo 6 se describe el *hardware* utilizado para realizar los experimentos. El resto del capítulo se centra en la descripción de los experimentos realizados para validar las arquitecturas. En este sentido, se implementaron tres arquitecturas para verificar el funcionamiento del *CPS Gateway*. En la primera, con la finalidad de validar los requisitos de interoperabilidad y alta disponibilidad, se implementaron los protocolos de comunicaciones industriales *Profinet, Profibus* y *Modbus*, así como los protocolos de alta disponibilidad *HSR* y *PRP*. En la segunda arquitectura se implementó un protocolo y un módulo de comunicaciones que permita la configuración remota del *CPS Gateway*. Finalmente en la tercera arquitectura se propone el uso de cifrado a nivel de Capa 2, para ofrecer seguridad a las tramas *SV* y *GOOSE* cumpliendo con las restricciones de tiempo que establece el estándar *IEC 61850*.

Finalmente, el capítulo 7 presenta las conclusiones extraídas de la presente tesis, así como las principales aportaciones de la misma. A su vez, se describen las publicaciones derivadas del presente trabajo y se enumeran varias líneas de investigación propuestas por el autor para dar continuidad al trabajo abordado en la presente tesis.

Capítulo 2

Smart Grid

2.1. Introducción

En la actualidad, tanto los países en desarrollo como los desarrollados han comenzado a promover la creación de una red eléctrica inteligente para hacer frente a la creciente demanda de energía por parte de la industria y de los hogares. Para la implementación de este nuevo concepto de red inteligente, es necesario el uso de tecnologías avanzadas de computación y de comunicación que permitan la ejecución de algoritmos de control distribuido. Para ello, es necesario definir arquitecturas de dispositivos electrónicos novedosos con la finalidad de que la generación, transmisión y distribución de energía eléctrica sea más eficiente, confiable y óptima.

En este capítulo se describen brevemente aspectos relativos a la *Smart Grid*: su definición y características, junto con las tecnologías de comunicación necesarias para la implementación de la *Smart Grid*. También se realiza la descripción de una subestación eléctrica y su arquitectura. Finalmente en la sección 2.3 y 2.4 se identifican y describen los estándares de automatización y control aplicables en la implementación de una red inteligente en la que se garantice un procesamiento en tiempo real, la interoperabilidad entre dispositivos, la seguridad, y la alta disponibilidad.

2.2. Red eléctrica

La arquitectura básica de la red eléctrica ha cambiado poco durante los últimos 100 años. Sin embargo, las tecnologías de automatización, control y comunicación en la última década han evolucionado rápidamente, permitiendo la definición del concepto de red inteligente [1]. En términos generales, el sistema eléctrico actual consiste en una compleja red en la que se encuentran interconectadas las plantas generadoras de energía, la infraestructura de transporte de electricidad, la infraestructura de distribución, y la carga. La red inteligente *(Smart Grid)* representada en la Figura 2.1, plantea un esquema de comunicación con flujo de la energía bidireccional, que hace más eficiente la transmisión, distribución, seguimiento y control del flujo de la electricidad [2]. Por ejemplo, una arquitectura de comunicación como la utilizada en los contadores inteligentes permite implementar métodos de predicción de la demanda.

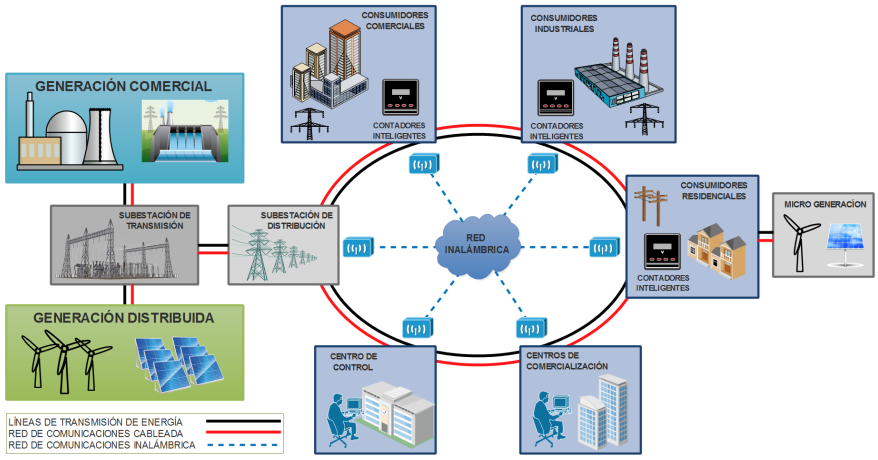


Figura 2.1: Esquema de Red inteligente

Un aspecto que ha influido en la necesidad de cambiar la forma de operación de la red eléctrica, es la integración a la red de las energías renovables, como la eólica, hidroeléctrica y la solar. Estas fuentes de energía tienen diferentes características de potencia y están limitadas por las condiciones ambientales. Esta heterogeneidad introduce una variabilidad significativa en el suministro. Además, las dinámicas de funcionamiento de estos sistemas de generación no están definidas en su totalidad, y se desconoce como afectarán en el comportamiento de la red.

Aunque hay diferentes definiciones para *Smart Grid*, se puede decir que, una red inteligente básicamente permite que diferentes partes de la red eléctrica se comuniquen entre sí. Es una comunicación máquina a máquina, canalizada a través de Internet u otro conducto, optimizando el rendimiento de la red. Una comunicación similar ya existe a menor escala, pero las redes inteligentes podrían conectar más dispositivos, desde el sistema de alumbrado hasta los electrodomésticos del hogar.

Desde un punto de vista tecnológico, la *Smart Grid* puede ser vista como una superposición de una red de comunicación sobre la red eléctrica. La red de comunicaciones en la *Smart Grid* es un tipo de red que puede ofrecer servicios de alto nivel, como el envío de datos de sensores en tiempo real, redundancia y ciberseguridad. Se implementa utilizando varias tecnologías de redes y medios de comunicación, entre los que se incluyen el mismo cableado eléctrico, las redes inalámbricas y otras infraestructuras de comunicación existentes como las redes *Ethernet* basadas en cobre o fibra óptica. Hay ventajas y desventajas asociadas con cada una de las opciones y es probable que los tres enfoques se puedan utilizar para las comunicaciones en la *Smart Grid*. Para determinar que tecnología utilizar es importante evaluar la infraestructura en varios factores clave, como por ejemplo: ancho de banda, latencia, seguridad y confiabilidad. Las tecnologías de red tiene propiedades estadísticas diferentes para cada uno de los factores antes mencionados, y deben ser integrados cuidadosamente para satisfacer los niveles de confiabilidad que la *Smart Grid* necesita.

Una gran parte de la eficiencia de la red eléctrica puede derivarse de un mejor sistema de comunicación. Primero, se requiere un sistema comunicación ubicua de baja latencia para recolectar datos desde los sensores y estimar el estado de la red con precisión. De esta forma se puede garantizar que la red permanezca estable y detectar cualquier anomalía antes que ocurra una falla en el sistema. En segundo lugar, un sistema comunicación permite realizar una estimación más precisa de la demanda, de modo que la oferta pueda controlarse con mayor precisión. Además, la comunicación bidireccional ayuda a gestionar la variación de la demanda (picos y valles) cambiando dinámicamente los patrones de uso a través del control directo de los dispositivos de los usuarios. Por último, la infraestructura de comunicación es necesaria para la monitorización remota y el diagnóstico de la infraestructura eléctrica. La modernización de la *Smart Grid*, aunque tiene como objetivo mejorar la entrega y la fiabilidad en la distribución de la energía eléctrica, depende en gran medida de las tecnologías de la información y la comunicación para aumentar la eficiencia.

2.2.1. Subestación eléctrica

Las subestaciones son componentes clave de la red eléctrica, facilitan el transporte y la distribución eficiente de la electricidad [27]. Proporcionan la interconexión entre las instalaciones generadoras, las redes de transmisión, distribución y los consumidores finales. Para realizar el seguimiento y control del flujo de la energía se implementan los sistemas de automatización de subestaciones *(Substation Automation System, SAS)* [28]. Los *SAS* desempeñan un papel vital y hacen posible el control y monitorización en tiempo real de los elementos que conforman una subestación (sensores, actuadores, equipos de comunicación y control). Además, ayudan a maximizar la disponibilidad, eficiencia, fiabilidad, seguridad e integración de datos que se transmiten por la red de comunicaciones [29].

En un principio, los *SAS* utilizaron el cableado de cobre de la red telefónica como medio de transmisión. Las unidades de control remoto basadas en la red telefónica conmutada estaban disponibles ya en el año 1930 y eran capaces de proporcionar información concerniente al estado y el control de unos pocos puntos. A medida que las comunicaciones digitales se convirtieron en una opción viable en la década de 1960, se utilizaron sistemas de adquisición de datos *(Data Acquisition Systems, DAS)* para recoger automáticamente los datos de medición de las subestaciones [30]. Considerando el limitado ancho de banda que se disponía, los protocolos de comunicación que se utilizaban en los *DAS* fueron optimizados para funcionar en canales de comunicación de bajo ancho de banda, pero sacrificando el tiempo que tarda el protocolo en configurar, mapear y almacenar la ubicación de los distintos bits de datos recibidos.

También, en los *SAS* se emplearon algunos protocolos estandarizados utilizados en la automatización industrial, como el protocolo de red distribuida versión 3 *(Distributed Network Protocol version 3, DNP3)* [31]. *DNP3* es un protocolo de comunicación de código abierto, que tiene dos tipos de dispositivos: 1) las estaciones máster *(Master Terminal Unit, MTU)*, y 2) las estaciones remotas *(Remote Terminal Units, RTU)*. Las *MTU* son dispositivos con cierta potencia de procesamiento y almacenamiento de datos, están ubicadas en un centro de control. Las *RTU* son dispositivos situados en la red, por ejemplo en líneas de transmisión, subestaciones y en transformadores. Estas unidades son responsables de recoger los datos de los sensores y enviarlos a la estación central [32].

Los actuales *SAS* están formados por diversos dispositivos electrónicos inteligentes *(Intelligent Electronic Devices, IED)* basados en microprocesadores. A modo

de ejemplo son equipos *IEDs* los interruptores de circuito de alta tensión, seccionadores, interruptores de puesta a tierra, transformadores, etc. Los *IED* pueden realizar todas las tareas locales como medición, monitoreo, protección y control en una estructura más descentralizada. La función de comunicación de la *RTU* se implementa a menudo en un equipo *IED* de puerta de enlace, que se encarga de convertir los protocolos de comunicación en ambas direcciones.

2.2.2. Arquitectura de un *Substation Automation System*

Los diferentes dispositivos que conforman un *SAS* se encuentran distribuidos en tres niveles. Hay equipos de nivel de proceso, de bahía y de estación. En la Figura 2.2. se muestra un diagrama típico de un *SAS* en donde se muestran los tres niveles.

El nivel de proceso esta compuesto por los equipos primarios, tales como seccionadores, interruptores y transformadores de corriente y tensión. La lógica de control en este nivel se implementa y se lo realiza de manera local en el propio mando de los interruptores o seccionadores.

En el nivel de la bahía se alberga los *IEDs* de protección y control. Estos *IEDs* se encargan de establecer comunicación directamente con los dispositivos del nivel de proceso, mediante entradas y salidas analógicas, digitales o mediante interfaces de comunicación.

A nivel de la estación, encontramos equipos de base de datos, estaciones de trabajo e interfaces de comunicación remotas. Desde este nivel se realizan tareas de supervisión, planificación y control de toda la subestación, utilizando los sistemas de supervisión control y adquisición de datos *(Supervisory Control And Data Acquisition, SCADA)*. Para realizar el control de la subestación desde centros remotos, tales como los centros de control de las compañías eléctricas, se utilizan equipos *gateway* que permiten interconectar la red interna de la subestación con los protocolos y arquitectura de la red remota.

Los diferentes niveles se conectan mediante dos buses de comunicación: el bus de proceso y el bus de estación. El bus de proceso conecta los niveles de proceso y de bahía, transporta información con requerimientos de tiempo crítico. El bus de estación enlaza el nivel de bahía con el nivel de la estación, transporta tanto información de gestión como de tiempo crítico.

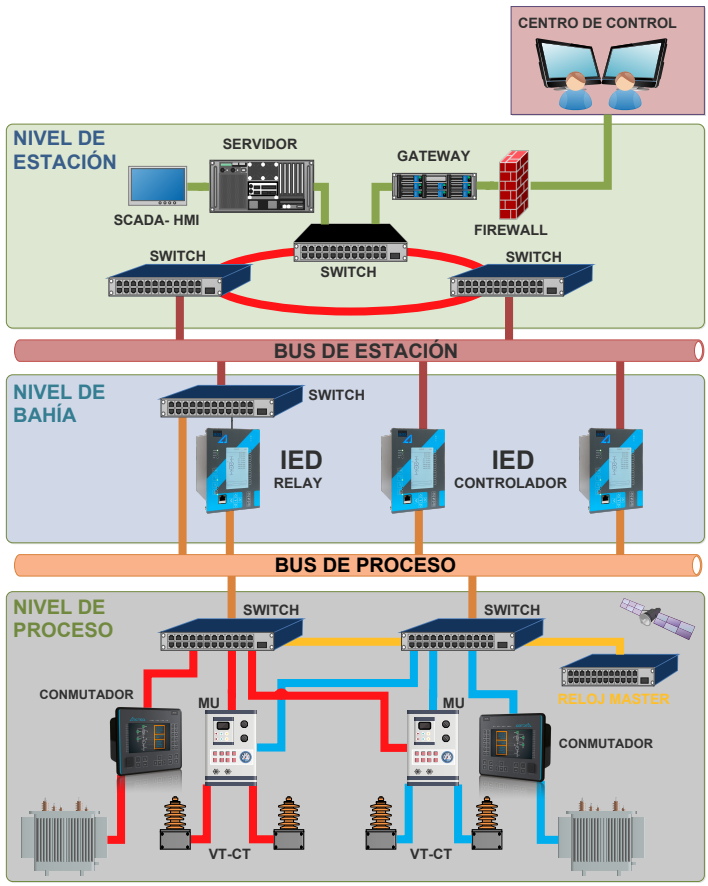


Figura 2.2: Niveles de control de un *SAS*

El flujo de la información a través de los diferentes niveles de la subestación es coordinado mediante los sistemas *SCADA*. En los sistemas *SCADA* tradicionales, la existencia de muchas soluciones propietarias dificultaba la interoperatividad entre dispositivos, incluso entre diferentes versiones del dispositivo del mismo proveedor. Por lo que era necesario implementar costosos y complejos convertidores de protocolo para mitigar el problema de la interoperabilidad. Otro problema que se ha detectado en los sistemas *SCADA* es la seguridad, en [33] indican que los incidentes cibernéticos y ataques dirigidos a infraestructuras críticas que utilizan sistemas *SCADA* han aumentado en los últimos años. Se espera que estos

incidentes y ataques aumenten en número y severidad debido a que los sistemas y protocolos *SCADA* no fueron diseñados para funcionar en entornos inseguros como Internet [34].

Para solventar problemas como la interoperatividad y la seguridad, en 1994, la *International Electrotechnical Commission (IEC)* comenzó a trabajar en el desarrollo de un estándar para los sistemas automatización y control de las subestaciones. La *IEEE*, en 1988 inició un trabajo similar en el desarrollo de un marco común de comunicación para las diversas capas del modelo *Open System Interconnect (OSI)* de la *International Standards Organization (ISO)*. Esta arquitectura dió lugar a la definición de un perfil de protocolos, modelos de datos y definiciones abstractas de servicios que se conocieron como *Utility Communication Architecture (UCA)*. Posteriormente, en 1997, con los conceptos y el trabajo realizado en la *UCA* y por los Grupos de Trabajo 10,11 y 12 del *IEC TC57* se desarrollo la norma *IEC 61850*, titulada *"Communication networks and systems for power utility automation"* [30]. Los principales objetivos de la norma eran asegurar la interoperabilidad entre dispositivos, una arquitectura abierta y ofrecer estabilidad a largo plazo [35].

2.3. Estándares de comunicaciones en la *Smart Grid*

Los requisitos de interoperabilidad del sistema, así como la enorme cantidad y tipología de datos que deben ser interconectados entre las partes y componentes que intervienen en un ecosistema *Smart Grid* requieren el uso de estándares de comunicación. La aplicación de estos estándares permiten la creación de una red inteligente de transmisión y distribución basada en las *TIC*.

La Figura 2.3 ofrece una visión general de los estándares utilizados en un ecosistema *Smart Grid*. En ella se identifican seis estándares para intercambiar información entre los diferentes niveles de la *Smart Grid*.

- *IEC 61968/61970/62325*: El objetivo de estas normas es proporcionar un modelo de datos para la transmisión de información entre los centros de control y los centros de gestión.
- *IEC 61850*: Comunicaciones en el ámbito de subestación y Recursos Ener-

géticos Distribuidos (RED).

- *IEC 60870*: Protocolos de telecontrol para establecer comunicación entre la subestación y los centros de control, se espera que sean obsoletos y se reemplacen por las especificaciones del estándar *IEC 61850*.
- *IEC 62351*: Define normas para dotar de mecanismos de seguridad a los protocolos de comunicación utilizados en la *Smart Grid*.

Cada estándar tiene sus partes individuales y subpartes, en las secciones siguientes se describe con más detalle cada uno de los estándares.

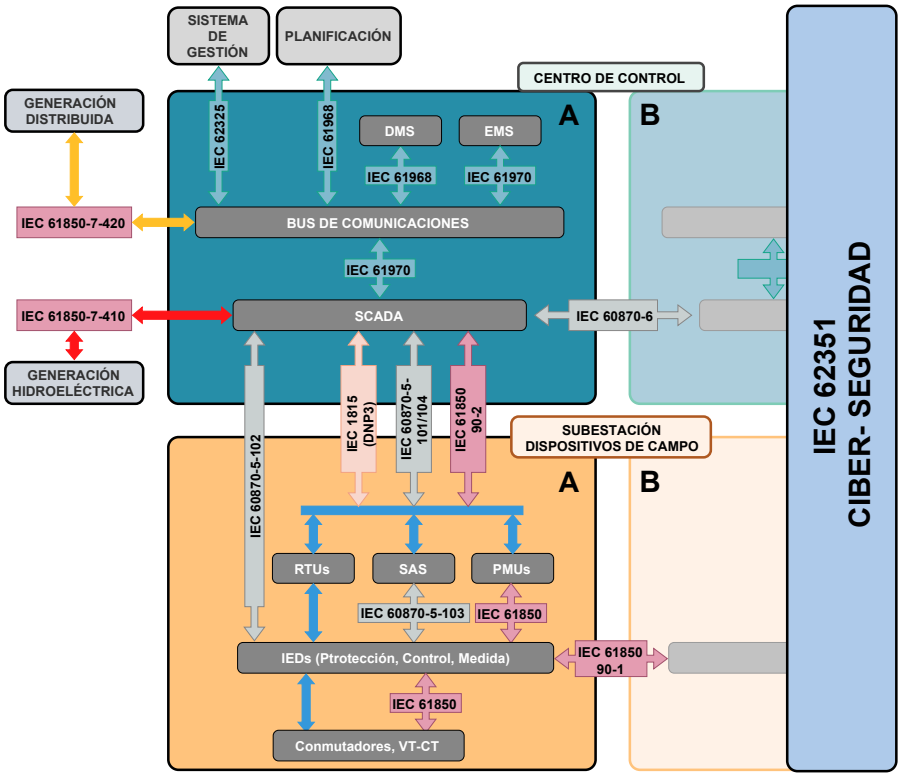


Figura 2.3: Estándares de comunicaciones en la *Smart Grid*

2.3.1. Estándar *IEC 61850*

La norma *IEC 61850* es un estándar internacional que surge por la necesidad de unificar las comunicaciones para conseguir interoperabilidad entre fabricantes [36]. Las distintas partes del estándar *IEC 61850* se publicaron por primera vez entre 2002 y 2005, teniendo como objetivo principal normalizar las comunicaciones entre los equipos de una *SAS*. La primera edición de la norma se enfocó principalmente a los aspectos de protección, control y monitoreo [22]. A partir de 2009, las partes originales de la serie *IEC 61850* han sido actualizadas y ampliadas para cubrir aspectos referentes a la medición y la calidad de energía. El estándar *IEC 61850* define los diferentes aspectos de la red de comunicaciones en una subestación en 14 apartados, agrupadas en 10 capítulos, tal como se muestra en la Tabla 2.1.

Tabla 2.1: Alcance y estructura inicial de la norma *IEC 61850*

Parte Nº	Título	Edición	Fecha
1	Introduction and Overview	2	2013/03
2	Glossary	1	2003/08
3	General requirements	2	2013/12
4	System and project management	2	2011/04
5	Communication requirements for functions and device models	2	2013/01
6	Configuration description language for communication in electrical substations related to IEDs	2	2009/12
7	Basic communication structure for substation and feeder equipment		
7-1	- Principles and models	2	2011/07
7-2	- Abstract Communication Service Interface *(ACSI)*	2	2010/08
7-3	- Common data classes	2	2010/12
7-4	- Compatible logical node classes and data classes	2	2010/03
8	Specific Communication Service Mapping *(SCSM)*		
8-1	-Mappings to MSS *(ISO 9506-1 and ISO 9506-2)* and to *ISO/IEC 8802-3*	2	2011/06
9	Specific Communication Service Mapping (SCSM)		
9-2	- Sampled Values over *ISO/IEC 8802-3*	2	2011/09
9.3	- Precision time protocol profile for power utility automation	1	2016/05
10	Conformance Testing	2	2012/12

Los conceptos definidos en *IEC 61850* se han aplicado a niveles más allá del dominio de subestaciones, tal y como se muestra en la Tabla 2.2. Por ejemplo, el modelado de las centrales hidroeléctricas *(IEC 61850-7-410)*, de recursos energéticos distribuidos *(IEC 61850-7-420)* y de aerogeneradores han sido estan-

darizados dentro de la serie *IEC 61400-25 "Communications for monitoring and control of wind power plants"*. También el estándar especifica las comunicaciones entre las subestaciones *(IEC 61850-90-1)*, la comunicación de la subestación con el centro de control *(IEC 61850-90-2)* y la comunicación directa entre los *Phasor Measurement Units (PMU)* con el centro de control *(IEC/TR 61850-90-5)*.

Tabla 2.2: Apartados adicionales del estándar *IEC 61850*

Parte Nº	**Título**	**Edición**	**Fecha**
7-410	Basic communication structure - Hydroelectric power plants - Communication for monitoring and control	2	2012/10
7-420	Basic communication structure - Distributed energy resources logical nodes	1	2009/03
7-500	Basic information and communication structure - Use of logical nodes for modeling application functions and related concepts and guidelines for substations	1	2017/07
7-510	Basic communication structure - Hydroelectric power plants - Modelling concepts and guidelines	1	2012/03
80-1	Guideline to exchanging information from a CDC-based data model using *IEC 60870-5-101* or *IEC 60870-5-104*	2	2016/07
80-3	Mapping to web protocols - Requirements and technical choices	1	2015/11
80-4	Translation from the COSEM object model *(IEC 62056)* to the *IEC 61850* data model	1	2016/03
90-1	Use of *IEC 61850* for the communication between substations	1	2010/03
90-2	Using *IEC 61850* for communication between substations and control centres	1	2016/02
90-3	Using *IEC 61850* for condition monitoring diagnosis and analysis	1	2016/05
90-4	Network engineering guidelines	1	2013/08
90-5	Use of *IEC 61850* to transmit synchrophasor information according to *IEEE C37.118*	1	2012/05
90-6	Use of *IEC 61850* for Distribution Automation Systems	1	2018/09
90-7	Object models for power converters in Distributed Energy Resources (DER) systems	1	2013/02
90-8	Object model for E-mobility	1	2016/04
90-10	Models for scheduling	1	2017-10
90-12	Wide area network engineering guidelines	1	2015/04
90-17	Using *IEC 61850* to transmit power quality data	1	2017-05

Adicionalmente en el diagrama de bloques de la Figura 2.4, se muestra el ámbito de aplicación de las diferentes partes de la norma *IEC 61850*.

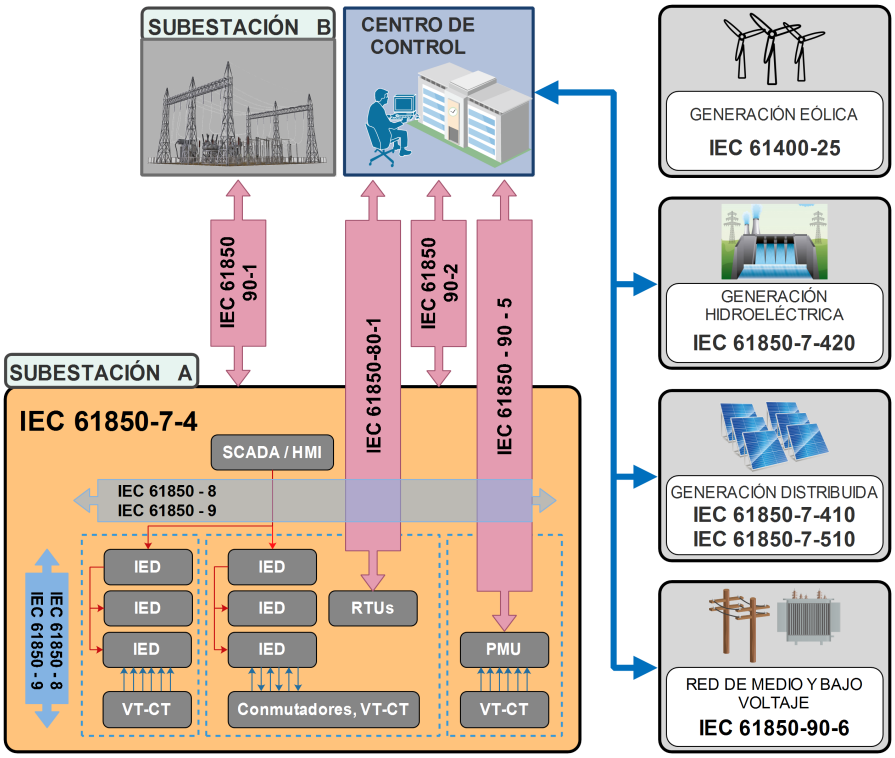


Figura 2.4: Estándar *IEC 61850*

En la parte 1 se encuentra una introducción y una visión general del estándar, la parte 2 contiene la terminología y definiciones utilizadas en el contexto de los *SAS*. Las partes 3, 4 y 5 identifican los diferentes requisitos generales y funcionales de los sistemas de comunicación de las subestaciones, así como la gestión de los proyectos.

La Parte 6 especifica el *Substation Configuration Language (SCL)*. *SCL* es un lenguaje que permite describir la configuración de los *IEDs* y del sistema de comunicaciones. Este lenguaje esta basado en el *Extensible Markup Language (XML)*. La parte 7-1 introduce los conceptos de datos comunes, métodos de mo-

delado, principios de comunicaciones y modelos de información que se utilizaran en las distintas partes de la serie *IEC 61850-7-x*. En la parte 7-2 se presentan las definiciones de clases y servicios a través de *Abstract Communication Service Interface (ACSI)*. Las clases de datos comunes se detallan en la Parte 7-3 y la Parte 7-4 especifica las reglas para establecer los nombres de las clases y datos que se definen en los nodos lógicos [2, 30].

Utilizando los *Specific Communication Service Mapping (SCSM)*, las definiciones abstractas de datos y servicios *(IEC 61850-7-x)* se mapean sobre un protocolo de comunicaciones en particular. La Parte 8 especifica un método para el intercambio de datos con y sin restricciones de tiempo utilizando como medio de transmisión las redes de área local *(Local Area Network, LAN)*, mediante el mapeo de *ACSI* en *Manufacturing Message Specification (MMS)* y en tramas *ISO/IEC 8802.3*. Las parte 9 define el mapeo de los *Sample Values (SV)* sobre enlaces seriales unidireccionales punto a punto *(IEC 61850-9-1)* o sobre una arquitectura de bus basada en *Ethernet (IEC 61850-9-2)*. La red de comunicación *Ethernet* basada en *IEC 61850-9-2* debe facilitar la transmisión de mensajes con bajos tiempos de latencia, como por ejemplo, tramas *Generic Object Oriented Substation Event (GOOSE)* y *SV* [30]. El resto de partes definen servicios de comunicación, modelos de datos y mapeo de servicios a diferentes protocolos de red para comunicaciones dentro o fuera de la subestaciones.

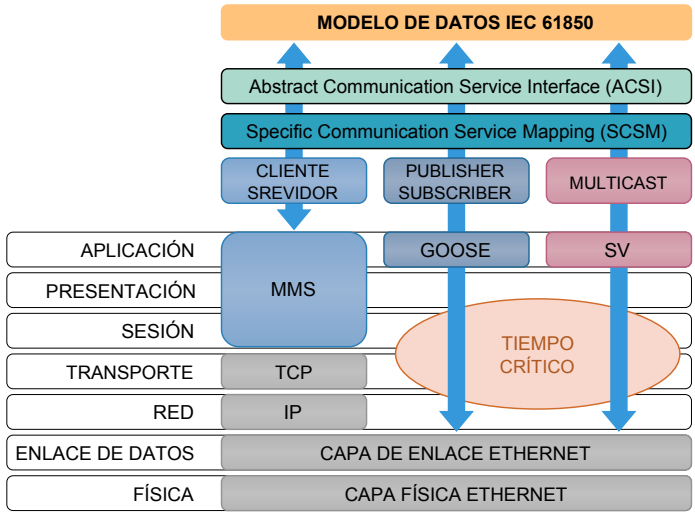


Figura 2.5: Arquitectura de comunicaciones *IEC 61850*

Para lograr que el estándar *IEC 61850* en un futuro siga cumpliendo con los requisitos de estabilidad a largo plazo, interoperabilidad, escalabilidad y de configuración abierta, el estándar plantea una arquitectura independiente del mecanismo de comunicaciones que se utilice, en la que sus modelos de objetos con sus datos y servicios (funciones) están separados de los protocolos de comunicación [37,38].

En la Figura 2.5 se presenta el mapeo de la pila de comunicación del estándar *IEC 61850* sobre el modelo de referencia *OSI*. Como ya se ha mencionado, *ACSI* permite realizar la definición de las objetos y servicios que se implementan en los *IED* utilizando el lenguaje de descripción *SCL*. Mientras que *SCSM* realiza el mapeo de los *ACSI* (objetos y servicios) sobre un protocolo en particular [39,40]. En el estándar *IEC 61850-8-1* y *IEC 61850-9-2* se definen tres tipos de tráfico [22]:

- *Manufacturing Message Specification (MMS)*, esta definido en *IEC 61850-8-1* y permite a un cliente *MMS* como un sistema *SCADA*, un servidor *OPC* o un *gateway* acceder verticalmente a todos los *IED*. Este tráfico fluye tanto en el bus de estación como en el bus de proceso, aunque algunos *IED* de bus de proceso no son compatibles con *MMS*.
- *Generic Object Oriented Substation Events (GOOSE)*, esta definido en *IEC 61850-8-1* y permite a los *IED* intercambiar datos horizontalmente entre el nivel de bahía o verticalmente entre el nivel de proceso y el nivel de bahía. Su utilización se centra en las señales de estado y de disparo, y a menudo para el enclavamiento.
- *Sample Values (SV)*, esta definido en *IEC 61850-9-2* y transporta datos de muestras de voltaje y corriente. Este tráfico fluye normalmente sobre el bus de proceso, pero también puede fluir sobre el bus de estación, por ejemplo, para la protección de barra colectora y la medición de los fasores.

El estándar *IEC 61850* ofrece tres tipos de modelos de comunicación: 1) Modelo cliente/servidor, 2) Modelo *"publisher/subscriber"* y 3) Modelo *multicast*. En el estándar *IEC 61850* la mayoría de las comunicaciones se basan en el modelo cliente/servidor. Para ello se utiliza *MMS* para el mapeo de los modelos y funciones *ACSI* y *Transmission Control Protocol/Internet Protocol (TCP/IP)* como protocolo de comunicaciones. Un ejemplo de este modelo es la transmisión de datos fallas y eventos hacia una base de datos. Para los mensajes de tiempo crítico, como *GOOSE* y *SV*, el mapeo se lo realiza directamente en la capa de enlace

de datos *Ethernet* (Capa 2) del modelo *OSI*. Los mensajes *GOOSE* utilizan el modelo de comunicación *"publisher/subscriber"*, generalmente transmiten datos binarios tales como estados, alarmas y señales de disparo. Los *SV*s se utilizan el modelo *multicast* para transmitir muestras sin procesar de corriente y voltaje de transformadores de corriente y voltaje *(CTs/VTs)* a los *IEDs*.

Las funciones de un *SAS* se refieren a tareas que se realizan en la subestación. Existen funciones para controlar, monitorizar y proteger los equipos de la subestación y sus alimentadores. También, existen funciones que son necesarias para tareas de mantenimiento en los *SAS*, es decir, para la configuración del sistema, la gestión de las comunicaciones y para tareas de sincronización de tiempo. Las funciones de un *SAS* pueden asignarse lógicamente a los tres niveles de automatización (proceso, bahía o estación) y para transmitir la información entre los diferentes niveles se utilizan las interfaces lógicas (1 a 11).

2.3.2. Estándar *IEC 62351* - Seguridad

En los últimos diez años, la *International Electrotechnical Commission (IEC)* ha realizado un gran esfuerzo en relación con la ciber-seguridad en la industria de servicios eléctricos [20]. Como resultado de este esfuerzo el *IEC* plantea el estándar *IEC 62351* para resolver problemas de seguridad en las diferentes áreas de operación y comunicación del sector eléctrico. El objetivo principal del estándar *IEC 62351* es definir normas para dotar de mecanismos de seguridad a los protocolos de comunicación que fueron definidos por el grupo de trabajo *IEC TC57* [21], concretamente la serie *IEC 60870-5*, *IEC 60870-6*, *IEC 61850*, *IEC 61970* e *IEC 61968*.

La primera versión del estándar *IEC 62351* fue publicado en 2007, y constaba de 8 partes.

La primera parte contiene un resumen general de la norma, describe el objetivo y presenta brevemente los diferentes capítulos. También proporciona información general sobre temas como: seguridad, amenazas a la seguridad, contramedidas de seguridad, evaluaciones de riesgos, gestión de claves y procesos de seguridad, entre otros.

La segunda parte se refiere al glosario de términos.

La tercera parte se enfoca en la seguridad de los protocolos basados en *TCP/IP* que se utilizan en los sistemas de automatización, específicamente en el ámbito de los sistemas de distribución eléctrica. Recomienda el uso de *Transport Layer Security (TLS)* con certificados X.509 para garantizar la autenticidad, integridad y confidencialidad de los datos en la capa de trasporte.

La cuarta parte se refiere a la seguridad de los perfiles *MMS* (por ejemplo, *IEC 61850-8-1* e *IEC 60870-6*). Las recomendaciones de seguridad en esta parte se analizan en términos de perfiles, el perfil de aplicación *A-Profile* (protocolos y requisitos de las capas 5-7 del modelo *OSI*) y perfil de transporte *T-Profile* (protocolos y requisitos de las capas 1-4 del modelo *OSI*). Para el *A-Profile*, la norma *IEC 62351-4* describe el uso de certificados *X.509* para autenticación, mientras que para el *T-Profile* la norma describe cómo utilizar *TLS* como una capa intermedia entre la capa de red y la capa de transporte, con el objetivo de utilizar un puerto *TCP* diferente para conexiones seguras.

La quinta parte de la norma *IEC 62351* trata la seguridad en los protocolos relacionados con el estándar *IEC 60870-5* y derivados como *DNP-3*.

La sexta parte (*IEC 62351-6*) especifica los mecanismos de seguridad para los sistemas de comunicación definidos en el estándar *IEC 61850* [24] y que no están basadas en *TCP/IP*. En concreto se proponen mecanismos de seguridad para protocolos *GOOSE* y *SV (IEC 61850-9)*. Las aplicaciones basadas en *MMS* deben incluir la confidencialidad de los datos además de la autenticación, de esta manera se garantiza la seguridad en los niveles de aplicación y de transporte [41].

La séptima parte *(IEC 62351-7)* utiliza los conceptos desarrollados en los estándares *Simple Network Management Protocol (SNMP)* y *Network and System Management (NSM)* para definir modelos de datos específicos para gestión y operación de los sistemas eléctrico. Estos modelos de datos *NSM* se utilizan para supervisar el estado de las redes y los sistemas, detectar posibles amenazas a la seguridad, gestionar el rendimiento y la fiabilidad de la infraestructura de comunicaciones.

La octava parte *(IEC 62351-8)* de la norma define el control de acceso basado en roles *(Role-Based Access Control, RBAC)* para los usuarios y agentes informáticos del sistema eléctrico, no es realmente un concepto nuevo, es utilizado por muchos sistemas operativos para controlar el acceso a los recursos del sistema. *RBAC* es una alternativa al modelo de superusuario todo o nada, planteando el principio de menor privilegio, según el cual un usuario no debe tener más derechos de

acceso que los necesarios para desempeñar su trabajo. Este concepto no solo se aplica a los usuarios, también es utilizado en la asignación de privilegios a los agentes informáticos que funcionan independientemente de las interacciones de los humanos. En [42] sus autores exponen con mayor detalle las diferentes partes (1-8) del estándar *IEC 62351*.

En los últimos años se han añadido cinco nuevas partes. La novena parte *(IEC 62351-9)* para abordar la gestión de claves y de certificados. La décima parte *(IEC 62351-10)* proporciona directrices generales sobre la arquitectura de seguridad de los sistemas de energía, y como aplicar los diferentes estándares para implementar de forma segura sistemas de generación, transmisión y distribución de energía. La parte *IEC 62351-11* especifica los esquemas, procedimientos y algoritmos necesarios para proteger y realizar un intercambio seguro de los archivos *XML*. La parte *IEC 62351-12* discute las recomendaciones de ciber-seguridad, las estrategias de ingeniería y operación para mejorar la capacidad de adaptación de los sistemas de energía con los recursos de energía distribuida *(Distributed Energy Resources, DER)*. La parte *IEC 62351-13* proporciona directrices sobre los temas de seguridad que podrían o deberían cubrirse en cualquier norma (*IEC*, *ISO* o otro tipo) que se van a utilizar en la industria energética.

En la actualidad el grupo de trabajo *IEC TC57* se encuentra desarrollando varias propuestas de seguridad que no se consideraron en un principio, como por ejemplo las partes 100-x tienen como objetivo proporcionar una descripción de pruebas de conformidad, para garantizar que las implementaciones operan conforme a la norma.

2.3.3. Estándar *IEC 60870*

El estándar *IEC 60870* fue desarrollado por grupo *IEC 57*. El protocolo *IEC 60870* define los sistemas utilizados para realizar telecontrol (control, supervisión, adquisición de datos y sus comunicaciones asociadas) en redes de transmisión de energía eléctrica u otros sistemas con amplia distribución geográfica [43]. El estándar *IEC 60870* tiene seis partes, representadas en la Tabla 2.3, que definen la información general relacionada con el estándar, las condiciones de operación, las interfaces eléctricas, los requisitos de rendimiento y los protocolos de transmisión.

La quinta parte del estándar es crucial para las comunicaciones en el sector energético, especialmente en los protocolos de comunicaciones de los sistemas

Tabla 2.3: Estructura del estándar *IEC 60870*

Parte Nº	Título	Edición	Fecha
1	General considerations	1	1988/12
2	Operating conditions	2	1995/12
3	Interfaces (electrical characteristics)	1	1989/05
4	Performance requirements	1	1990/04
5	Transmission protocols	1	1990/02
6	Telecontrol protocols compatible with ISO standards and ITU-T recommendations	1	1995/05

SCADA. Consta de siete secciones básicas, en las que se definen: los formatos de las tramas de transmisión, los servicios de transmisión a nivel de capa de enlace de datos, estructura de los datos a nivel de aplicación, definición y codificación de elementos de información, funciones básicas a nivel de aplicación, directrices para las pruebas de conformidad y extensiones de seguridad (aplicación del estándar *IEC 62351*). Considerando el avance de la tecnología y los nuevos requerimientos de comunicaciones el comité técnico *IEC 57* también ha generado estándares complementarios. Entre ellos, dos son los más importantes, el estándar *IEC 60870-5-101* y *IEC 60870-5-104*, estas partes definen un conjunto de métodos que permiten el intercambio de información entre una *RTU* ubicada en la red eléctrica y una estación *SCADA* [43, 44]. En la Tabla 2.4, se observa la estructura actual del estándar *IEC 60870-5*.

El estándar *IEC 60870-5-101* se ha generalizado su uso en el sector eléctrico, se basa en la arquitectura *Enhanced Performance Architecture (EPA)* y define unicamente las capas física, de enlace y de aplicación del modelo *OSI* [45]. El protocolo está diseñado para ser utilizado en sistemas donde existen conexiones directas y permanentes entre la estación de control (maestro) y las estaciones controladas (esclavos). *IEC 60870-5-101* se usa principalmente con medios de transmisión seriales relativamente lentos en la interfaz asíncrona *V.24 (EIA-232D o ITU-T)*. La norma garantiza tasas de transmisión de 9600 - 115200 bit/s. La trama de datos de este protocolo se denomina *FT1.2*. Esta trama puede ser de longitud fija o variable, dependiendo si es para el establecimiento y control de un proceso de comunicación (longitud fija) o datos de usuario (longitud variable). El estándar permite el uso de dos tipos de comunicación, balanceado y no balanceado. En el modo balanceado las estaciones que intervienen en la comunicación *(SCADA y RTU)* son dispositivos primarios y cualquiera de ellos puede inicializar una transmisión. En el modo no balanceado hay una estación de control llamada estación primaria, que puede inicializar la comunicación mientras que las otras estaciones

llamadas secundarias están en espera de los datos de la estación primaria [46].

Tabla 2.4: Estructura del estándar *IEC 60870-5*

Parte Nº	Título	Edición	Fecha
5-1	Transmission Frame Formats	1	1990/02
5-2	Data Link Transmission Services	1	1992/04
5-3	General Structure of Application Data	1	1992/09
5.-4	Definition and Coding of Information Elements	1	1993/08
5-5	Basic Application Functions	1	1995/06
5-6	Guidelines for conformance testing for the *IEC 60870-5* companion standards	1	2006/03
5-7	Security extensions to *IEC 60870-5-101* and *IEC 60870-5-104* protocols (applying *IEC 62351*)	1	2013/07
5-101	Companion standard for basic telecontrol tasks	2.1	2015/11
5-102	Companion standard for the transmission of integrated totals in electric power systems	1	1996/06
5-103	Companion standard for the informative interface of protection equipment	1	1997/12
5-104	Network access for *IEC 60870-5-101* using standard transport profiles	2.1	2016/06
5-601	Conformance test cases for the *IEC 60870-5-101* companion standard	2	2015/10
5-604	Conformance test cases for the IEC 60870-5-104 companion standard	2	2016/06

El estándar *IEC 60870-5-104* a diferencia del estándar *IEC 60870-5-101* utiliza las redes de datos y el protocolo *TCP/IP* para intercambiar los mensajes de telecontrol entre los centros de control y *RTU*. Agrega las capas de transporte y de red al modelo *EPA* para proporcionar la totalidad de accesos a la red [47]. El protocolo *IEC 60870-5-104* es la extensión del protocolo *IEC 60870-5-101*, pero no todas las funciones que soporta el *IEC 60870-5-101* son compatibles con *IEC 60870-5-104*. Por ejemplo el *IEC 60870-5-104* no soporta marcas de tiempo o *timestamp* (con un formato de 3 *byte*). Aunque ambos protocolos no son compatibles en su totalidad, muchas veces se combinan porque se pueden sincronizar fácilmente. La mayor diferencia entre los dos protocolos es que el *IEC 60870-5-104* permite transmitir datos simultáneamente entre servidores y dispositivos esclavos.

2.4. Otros estándares de comunicaciones

Aunque la norma *IEC 61850* está siendo introducida en todos los niveles de las subestaciones y cubre todas las necesidades de comunicación, se requieren protocolos adicionales para desplegar redes de comunicación confiables y eficientes. En particular, en esta sección, se describen algunos protocolos que son necesarios para proporcionar interoperatividad, sincronización, redundancia y gestión.

2.4.1. Estándar *IEC 61158* - Buses de campo

Como se ha comentado, *IEC 61850* es el estándar que cumple con todos los requerimientos actuales de comunicaciones en una subestación, y además posee la flexibilidad suficiente para adaptarse a los requerimientos del futuro. Pero antes de la aparición del estándar *IEC 61850* existieron otras soluciones que permitieron resolver el problema de la interconexión. Una de estas soluciones son los denominados buses de campo, que son propias del ámbito industrial, pero que resolvieron satisfactoriamente los problemas de integración de datos en las subestaciones. A pesar de la proliferación de dispositivos que funciona bajo el estándar *IEC 61850*, en la actualidad todavía existe equipamiento con sistemas de comunicaciones basados en buses de campo estandarizados. Por lo que es necesario incluir este estándar entre los utilizados en el entorno energético.

El estándar *IEC 61158 "Industrial communication networks - Fieldbus specifications"*, define el concepto de bus de campo *(Fieldbuses)* y la forma como se implementarán. En la mayoría de las redes industriales la transferencia de bits de información se realiza de manera serial, la transferencia de datos en serie tiene la ventaja de requerir solo un número limitado de cables para el intercambio de información entre dispositivos. Con menos cables, la sincronización es más fácil y podemos enviar información a través de grandes distancias. Además de la infraestructura física de una red industrial, es necesario disponer de protocolos de comunicación, que asegure que el sistema envíe la información de forma fiable, segura y sin errores entre los nodos de la red. El conjunto de infraestructura física y el protocolo de comunicaciones se denomina bus de campo. En el ámbito de una subestación, un bus de campo *Ethernet* conecta los equipos que se encuentran ubicados en el nivel de proceso con el nivel de bahía.

Las diferentes partes del estándar *IEC 61158* especifican varios tipos de bus de

campo. Cada tipo de protocolo está diseñado para permitir la comunicación de múltiples dispositivos de medición y control. Generalmente un bus de campo ofrece capacidades de operar en un sistema en tiempo real, y a menudo se implementan únicamente las capas 1, 2 y 7 del modelo de referencia *OSI*. La estructura del estándar es muy compleja, en la actualidad consta de 82 documentos agrupadas en seis partes. La Tabla 2.5, presenta la estructura simplificada del estándar *IEC 61158*.

Tabla 2.5: Estructura simplificada del estándar *IEC 61158*

Parte Nº	**Título**
1	Overview and guidance for the *IEC-61158* and *IEC-61784* series
2	Physical layer specification and service definition
3	Data-link layer service definition
4	Data-link layer protocol specification
5	Application layer service definition
6	Application layer protocol specification

La primera parte *(IEC 61158-1)* del estándar define el concepto de bus de campo, presenta una visión general y una guía de la norma, describe la estructura de la norma, su relación con el modelo *OSI* y presenta brevemente los diferentes capítulos. El término "servicio" se utiliza a lo largo de toda la estructura del estándar, y se refiere a la capacidad abstracta de una capa del modelo *OSI* para dar soporte a los requerimientos de capa inmediatamente superior. La segunda parte *(IEC 61158-2)* está relacionada con la capa 1 del modelo *OSI*, se describen los servicios, modelos y las especificaciones eléctricas y mecánicas de la capa física. La tercera *(IEC 61158-3)* y cuarta *(IEC 61158-4)* parte se relacionan con la Capa 2 del modelo *OSI* (Enlace de Datos). En la tercera parte especifican las características de los servicios necesarios para dar soporte a los requerimientos de la capa de aplicación. La cuarta parte especifica las características propias del protocolo de comunicaciones, como se implementa y la estructura que debe tener. Los protocolos definidos en la cuarta parte *(IEC 61158-4)* proporcionan servicios a la capa de enlace de datos, utilizando los servicios disponibles en la capa física *(IEC 61158-2)*. La quinta *(IEC 61158-5)* y sexta *(IEC 61158-6)* parte se relacionan con la capa 7 del modelo *OSI* (Capa de aplicación). En la quinta parte se definen las características de los servicios de capa de aplicación que cualquier protocolo de nivel superior puede explotar. En la sexta parte se especifican las características del protocolo que se encarga de relacionar los servicios definidos en la capa de aplicación *(IEC 61158-5)* con los servicios implementados en la capa de enlace de datos o física *(IEC 61158-3, IEC 61158-2)*. La Figura 2.6, muestra la estructura del estándar y su relación con las capas (1-2-7) del modelo *OSI*.

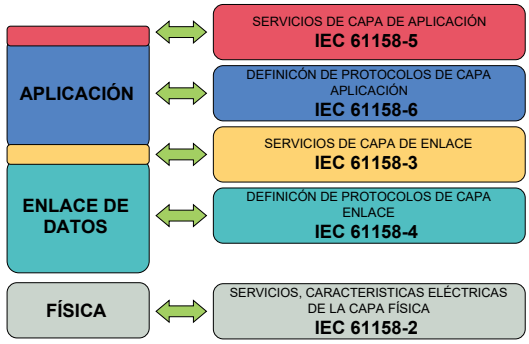


Figura 2.6: Estructura del estándar *IEC 61158* y su relación con modelo *OSI*

2.4.2. Estandar *IEC 61588* - Protocolos de sincronización

Los dispositivos que componen la red eléctrica necesitan colaborar y coordinar sus operaciones para realizar una tarea compleja. El problema de la sincronización en una red radica en conseguir que todos los elementos de la red tengan sus relojes internos ajustados en frecuencia y fase, para que en una comunicación, la transmisión y recepción de los datos se lleven a cabo sin deslizamientos en los intervalos de tiempo asignados.

En el pasado cuando se producía un corte en el suministro de energía, la identificación y alineación (que sean del mismo instante de tiempo) de los datos de falla de diferentes ubicaciones era una tarea muy compleja, en este sentido surgió la necesidad de incorporar mecanismos de sincronización de tiempo. El uso de una referencia única de tiempo en todos los *SAS* es muy importante para gestionar tareas complejas y monitorizar el comportamiento de la subestación (por ejemplo, alarmas, protecciones y sistemas de control).

Tradicionalmente, el método de sincronización más utilizado (especialmente en el sector eléctrico) era el definido por el *Inter-Range Instrumentation Group (IRIG-B)* [48, 49], pero necesitaba infraestructura dedicada, lo cual incrementaba considerablemente los costes. Además eran necesarios complejos procesos de calibración para compensar los retardos de propagación variables de las señales [50]. Para reducir los costos se desarrollaron protocolos de sincronización que se transmitían por la misma red de datos. Uno de los primeros protocolos

de sincronización que se planteó fue *Network Time Protocol (NTP)*, fue utilizado en aplicaciones donde la precisión necesaria estaba por debajo de los milisegundos [51]. En *NTP* la precisión depende de la extensión y complejidad de la red, en una red *LAN* la precisión esta en el orden de los milisegundos, mientras que en una *WAN* se puede incrementar hasta una decena de milisegundos [52]. Sin embargo, para redes industriales, la sincronización de relojes requiere una mayor precisión. En este sentido surgió el estándar *IEEE 1588 Precision Time Protocol (PTP)* [53], adoptado por primera vez en 2002 para aplicaciones de automatización y medición, establece un método para la sincronización de relojes con precisión de microsegundos. Posteriormente, en 2008, se ratificó la segunda versión del estándar para abordar las aplicaciones de automatización y de control de sistemas, la medición y la prueba automática de los sistemas, la generación de energía, los sistemas de transmisión y distribución y las telecomunicaciones.

2.4.3. Estándar *IEC 62439* - Redes de alta disponibilidad

El método básico para incrementar la disponibilidad frente a fallos es añadir redundancia en aquellos elementos en los que puedan darse. En este contexto, en el campo de las comunicaciones se ha elaborado el estándar para Redes *Ethernet* Industriales Fiables *IEC 62439*. El estándar esta compuesto por siete partes, la primera parte contiene un resumen general de la norma, describe el objetivo y presenta brevemente los diferentes capítulos. También proporciona información general sobre temas como: redes de alta disponibilidad, ejemplos de protocolos redundantes, topologías de red redundantes, etc. El resto de partes de la norma aborda las diferentes topologías y mecanismos que se mencionan en la primera parte del estándar. En la Tabla 2.6 se describen las diferentes partes del estándar.

Tabla 2.6: Estructura del estándar *IEC 62439*

Parte Nº	Título
1	General concepts and calculation methods
2	Media Redundancy Protocol *(MRP)*
3	Parallel Redundancy Protocol *(PRP)* and High-availability Seamless Redundancy *(HSR)*
4	Cross-network Redundancy Protocol *(CRP)*
5	Beacon Redundancy Protocol *(BRP)*
6	Distributed Redundancy Protocol *(DRP)*
7	Ring-based Redundancy Protocol *(RRP)*

Analizando las características de los diferentes mecanismos de redundancia que se especifican en el estándar *IEC 62439*, el grupo de trabajo *TC57 WG10* adoptó los protocolos especificados en la tercera parte del estándar *(IEC 62439-3)* como el método de redundancia para redes de automatización de subestaciones basadas en el estándar *IEC 61850*. Los dos estándares contemplados en el mismo son *Parallel Redundancy Protocol (PRP)* y *High-availability Seamless Redundancy (HSR)*. Para implementar estos protocolos es necesario que los dispositivos dispongan de dos interfaces *Ethernet*. *HSR* y *PRP* son protocolos *Zero Packet Loss (ZPL)* ya que al duplicarse todos los paquetes, en caso de pérdida de conexión entre dos nodos, el nodo destino recibirá siempre un paquete por alguna de las interfaces [54].

En el caso del protocolo *PRP*, cada puerto de un nodo está conectado a una red *Ethernet* independiente y convencional (LAN A y LAN B), la estructura de una red *PRP* se observa en la Figura 2.7(a). Los nodos *PRP* se denominan *Dual Attached Nodes (DANs)* [55]. Los dos puertos envían las tramas *Ethernet* redundadas por ambas redes, incluyendo la información necesaria para asegurar el adecuado tratamiento de las mismas (número de secuencia, identificadores de redundancia, etc.). En recepción, el nodo destino recibirá la misma trama por ambas interfaces, procesa la primera trama recibida y descarta el duplicado. En caso de un fallo en una de las redes, se sigue manteniendo la comunicación por la segunda. Las tramas *PRP* pueden circular por la infraestructura de una red *Ethernet* convencional. En redes *PRP* es posible conectar equipos con una sola interfaz *Ethernet* denominados como *Single Attached Nodes (SANs)* [55], como ordenadores, impresoras, etc. Para dotar de redundancia a los SANs es necesario utilizar un equipo de adaptación denominado *Redundancy Boxes (RedBox)* que nos permita duplicar el tráfico recibido del SAN hacia dos puertos, y gestiona los protocolos redundantes [54, 56]. Un equipo SAN conectado a una red LAN con acceso redundante mediante RedBox se denomina *Virtual-DAN (VDAN)*.

HSR, al igual que *PRP* ofrece redundancia enviando los paquetes por dos puertos *Ethernet*, pero a diferencia de *PRP*, los nodos *HSR* forman un anillo a través de sus dos puertos *Ethernet* [57, 58], en la Figura 2.7(b) se muestra la estructura de una red *HSR*. En este caso, los nodos se denominan *Doubly Attached Bridging Nodes HSR (DANHs)* [56,59]. Los nodos *DANHs* además de gestionar la eliminación de tramas redundadas y la emisión de tramas de supervisión, deben realizar el *forwarding* de las tramas que no están enviadas a ninguna de sus interfaces. Cada equipo se convierte en un *switch* de la red. Los tiempos de latencia fijados en el estándar para la retransmisión de tramas es muy exigente, por lo que es necesario utilizar dispositivos que realicen *"cut-through"*, es decir, retransmitir

un paquete antes de haberlo recibido por completo, unicamente identificando la dirección del nodo destino. Por este motivo, a diferencia de *PRP*, las tramas *HSR* no pueden ser retransmitidas por equipos *Ethernet* convencionales. *HSR* puede funcionar en modo *unicast* o modo *multicast*. Al igual que en *PRP*, si queremos conectar a una red *HSR* un nodo con una sola interfaz *Ethernet* tendremos que usar una RedBox que permita distribuir el paquete recibido del nodo a través de ambos puertos del anillo [56].

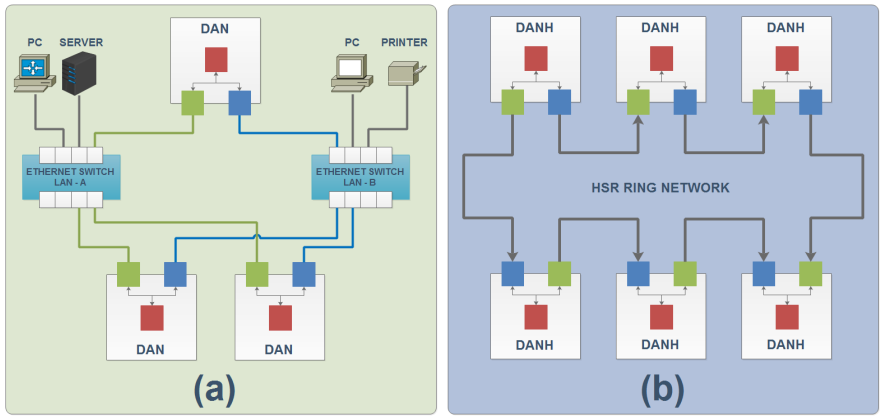


Figura 2.7: Redes de alta disponibilidad: (a) *PRP*. (b) *HSR*.

2.4.4. Estándar *IEC 61970/61968/62325* - Gestión de la energía

Mientras que el estándar *IEC 61850* se centra principalmente en la comunicación entre varios dispositivos del sistema eléctrico, la norma *IEC 61970* se concentra en la transmisión de información de los sistemas de gestión de energía. El objetivo de estas normas es proporcionar un modelo de datos para intercambiar datos complejos que representan información sobre muchos aspectos del sistema eléctrico, desde la topología, datos de activos hasta aspectos económicos. El estándar *IEC 61968* extiende las definiciones de la norma *IEC 61970* al ámbito de los sistema de distribución [60]. Y el estándar *IEC 62325* plantea un conjunto de normas para realizar intercambio de información entre los participantes y operadores del mercado energético.

El núcleo central de los estándares *IEC 61970/61968/62325* se basa en el uso de

modelos de información común *(Common Information Model, CIM)*. *CIM* fue desarrollado originalmente por el *Electric Power and Research Institute (EPRI)* a mediados de los 90, uno de los principales objetivos del *CIM* es proporcionar un modelo de datos independiente de la plataforma. Para la descripción de los modelos y sus clases se utiliza lenguaje *Unified Modeling Language (UML)* [61]. *UML* es un lenguaje para modelar sistemas a nivel de *software*, utiliza un entorno gráfico para visualizar, especificar, construir y documentar un sistema [62].

IEC 61970 "Energy Management System - Application Program Interface (EMS-API)". El extenso modelo de datos definido en *IEC 61970-301* representa la parte principal de este estándar e incluye la mayoría de los objetos requeridos para modelar redes eléctricas. Adicionalmente, el *IEC 61970* contiene las especificaciones de Interfaz de los componentes *(Component Interface Specifications, CIS)*, definiendo cómo se pueden utilizar los modelos de datos (independientes de la plataforma) y las interfaces genéricas en combinación con los estándares de comunicación.

IEC 61968 "Application integration at electric utilities - System interfaces for distribution management". A diferencia de la *IEC 61970*, el *IEC 61968* se centra más en los objetos virtuales necesarios para casos de uso comercial como la facturación, mercadeo o la planificación de la ampliación de la red. Los modelos de datos definidos en *IEC 61970-301*, se amplía con otros objetos especificados en *IEC 61968-11*.

IEC 62325 "Framework for Energy Market Communications". *IEC 62325* presenta un conjunto de normas que describen las directrices para las comunicaciones en el sector energético al nivel administrativo (productores, distribuidores y comercializadores de la energía eléctrica). Sus partes principales abarcan las comunicaciones entre los participantes y los operadores del negocio. Además, se especifican dos estilos de mercado: estilo europeo y estilo estadounidense.

2.5. Resumen

En este capítulo se ha presentado una visión general de la red eléctrica. Se ha identificado como diferencia entre la red tradicional y la denominada *Smart Grid* el hecho de que en la primera el flujo de energía y comunicaciones es unidireccional, es decir, de las fuentes de generación hacia los usuarios (carga). Por el

contrario, en la *Smart Grid* se establece un esquema de comunicación y flujo de energía bidireccional, por ejemplo, mediante contadores inteligentes, los usuarios pueden proporcionar datos en tiempo real sobre el consumo de energía, y también pueden suministrar energía a la red a través de pequeñas instalaciones de paneles solares o de aerogeneradores.

El elemento fundamental que permite la integración de la red de eléctrica y la red de comunicaciones es la subestación eléctrica. Para monitorizar y controlar del flujo de la energía en una subestación se implementan los sistemas de automatización de subestaciones *(SAS),* los cuales están compuestos por equipos de diferentes características distribuidos en tres niveles, denominados nivel de proceso, nivel de bahía y nivel de estación.

En los *SAS* y en la *Smart Grid* en general para asegurar la interoperabilidad entre dispositivos de diferentes fabricantes, es necesario emplear protocolos estandarizados como por ejemplo los definidos en la familia del estándar *IEC 61850*. Además de los modelos de comunicaciones descritos en el estándar *IEC 61850*, también es necesario contar con protocolos adicionales como el *IEEE 1588* para garantizar una sincronización precisa y la redundancia en las comunicaciones utilizando el estándar *IEC 62439-3*.

Cuando se introducen protocolos y *software* estandarizados, se presentan nuevas amenazas digitales que deben ser tratadas adecuadamente por los sistemas de comunicación. En este sentido, la norma *IEC 62351-6* especifica mecanismos de seguridad para proteger las comunicaciones definidas en el estándar *IEC 61850*. Sin embargo, las investigaciones han demostrado que los mecanismos de seguridad propuestos en la actual norma *IEC 62351-6 (ed.1)* son ineficientes, y se espera que la nueva versión del estándar, que se publicará en noviembre de 2019, corrija los errores detectados en la versión anterior.

Aunque el estándar *IEC 61850* se está incorporando a todos los niveles de las subestaciones cubriendo todas las necesidades de comunicación, todavía es necesario utilizar protocolos adicionales para el despliegue de redes de comunicaciones confiables y eficientes. Por ejemplo el estándar *IEC 61158*, debido a que en la actualidad todavía existe equipamiento antiguo con sistemas de comunicaciones basados en buses de campo.

Finalmente, los estándares *IEC 61970/61968/62325*, aunque no intervienen en las comunicaciones de una subestación, son indispensables para el intercambio de información entre participantes y operadores del mercado energético.

Capítulo 3

Requisitos de operación de los dispositivos para la *Smart Grid*

3.1. Introducción

Para poder diseñar un CPS Gateway para su utilización en la Smart Grid, es imprescindible que el sistema cumpla con los requisitos exigidos por los estándares. En este sentido, en la sección 3.2 de este capítulo se identifican los requisitos de operación de los dispositivos utilizados en una subestación eléctrica, para lo cual se analizan las estructuras de los datos que se transmiten por las redes de comunicaciones del sector eléctrico, que se definen en las partes 5, 8-1, 9-2 de la norma *IEC 61850*. También se analizan otros requerimientos que no están definidos en el estándar *IEC 61850*, pero que son necesarios considerar para garantizar la interoperabilidad, el procesamiento de eficiente de los datos y la seguridad de las comunicaciones.

3.2. Requerimientos

A fin de diseñar un CPS Gateway de alta capacidad para ser utilizado en la Smart Grid, es imprescindible que el sistema cumpla con las más altas exigencias en cuanto a los requerimientos de los estándares internacionales. Para determinar estos requerimientos es necesario identificar qué tipo de estructuras de datos son transmitidos por las redes de comunicación en el sector eléctrico. Para ello es necesario realizar un análisis de las partes *IEC 61850-5*, *IEC 61850-8-1* y *IEC 61850-9-2* del estándar *IEC 61850.*

Como se observa en la Figura 3.1, el estándar *IEC 61850* define una estructura en la que se puede identificar cinco perfiles de comunicación. Estos son: *SV, GOOSE, GSSE, PTP y MMS*. Estos a su vez se agrupan en perfiles con requerimientos de operación en tiempo real: *SV, GOOSE, GSSE* y *PPT*. Estos servicios de comunicación utilizan protocolos que no son *IP* para lograr la latencia necesaria, para lo que realizan el mapeo de los datos a nivel de Capa 2 en tramas *Ethernet*. *MMS* son servicios con tiempos de operación más permisivos y utilizan los protocolos *TCP/IP* para enviar la información.

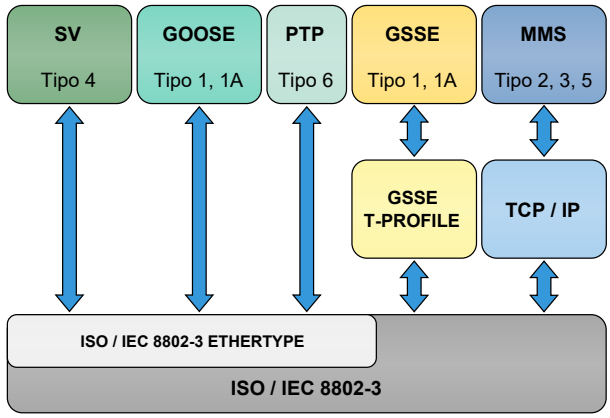


Figura 3.1: Tipo de mensaje y la clase de rendimiento definidos en *IEC 61850-5*

Los perfiles mencionados anteriormente se utilizan para encapsular los distintos mensajes que se generan en la subestación. Los mensajes se clasifican en siete tipos, desde el tipo 1 que cubre los mensajes rápidos hasta el tipo 7 para los

mensajes de comando y control de acceso. En la Tabla 3.1 se resumen los tipos de mensajes, utilización y perfil de comunicación que se utiliza para transmitir. Por ejemplo los mensajes tipo 1(A/B) y tipo 4 son los que necesitan menores tiempos de ejecución, normalmente contienen datos binarios, comandos simples o datos sin procesar de corriente y voltaje, por lo que son mapeados en perfiles *GOOSE* y *SV*.

Tabla 3.1: Tipo de mensajes definidos en el estandar *IEC 61850*

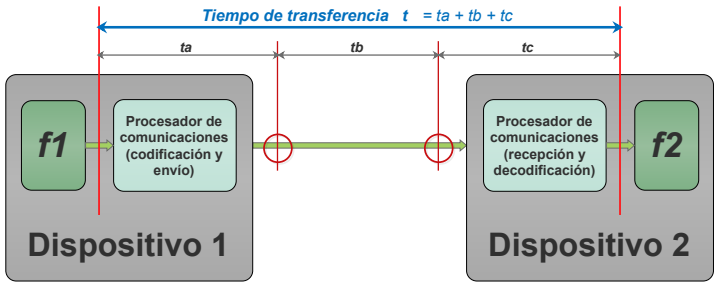
Tipo	Clase	Mensaje
1A	Mensajes de disparos (*trip*)	*GOOSE*
1B	Mensajes rápidos utilizados para protección	*GOOSE, GSSE*
2	Mensajes de rapidez media para el control (mandos, alarmas, etc)	*MMS*
3	Mensajes lentos para supervisión y configuración	*MMS*
4	Envío de valores instantáneos de señales analógicas	*SV*
5	Transferencia de ficheros	*MMS*
6A	Sincronización de tiempo (bus de estación)	*TimeSync (PTP)*
6B	Sincronización de tiempo (bus de proceso)	*TimeSync (PTP)*
7	Mensajes lentos para comandos de control desde *HMI*	*MMS*

Los diferentes tipos de mensajes identificados en la Tabla 3.1, deben satisfacer varios requisitos de rendimiento definidos por el estándar *IEC 61850*, entre los que podemos mencionar el tiempo de transferencia (latencia), la sincronización y el tiempo de restablecimiento.

Otros requisitos que están fuera del alcance del estándar pero son importantes considerar son la interoperatividad, la capacidad de reconfigurarse, la posibilidad de realizar análisis local de datos y la ciber-seguridad. A continuación se detallan cada uno de ellos.

3.2.1. Tiempo de transferencia (Latencia)

El concepto de Tiempo de Transferencia definido en la norma *IEC 61850-5* representa el tiempo necesario para la transmisión de un mensaje entre dos dispositivos. El tiempo de transferencia t como se muestra en la Figura 3.2, comienza desde el momento en que el emisor coloca el contenido de los datos de la aplicación sobre su pila de transmisión (codificación y envío), hasta el momento en que el receptor extrae los datos de su pila de transmisión (recepción y decodificación). El tiempo de transferencia t es la suma de los tiempos de procesamiento de los datos en ambos extremos (ta, tc) y el tiempo de transferencia de los datos sobre la red de comunicaciones (tb).



R

Figura 3.2: Definición del tiempo de transferencia según *IEC 61850-5*

Con el objetivo de especificar el tiempo de transferencia necesario para los diferentes tipos de mensajes, en el estándar *IEC 61850-5* se definen siete clases de tiempo de transferencia (TT0 a TT6) cada una con diferentes tiempos de transferencia. El rango de estos tiempos cubre desde 1000 ms máximos de latencia para la clase TT0 hasta 3 ms para la clase TT6. El tiempo máximo de transferencia para el intercambio de información se utiliza para clasificar los diferentes tipos de mensajes según clases de rendimiento. En el estándar se definen doce clases de rendimiento P. Por ejemplo para los mensajes de tipo 1A tenemos los tipos de clase de rendimiento: P1 y P2. P1 asocia los mensajes críticos 1A con mayor requerimiento temporal con la clase TT6 que define un tiempo de transferencia no mayor a 3 ms. P2 asocia los mensajes críticos 1A con menor requerimiento temporal con la clase TT5 que define un tiempo de transferencia menor de 10 ms. La Tabla 3.2 resumen los tipos de mensajes (Tabla 3.1), la clase de rendimiento a la que pertenece, su relación con las clases de tiempo de transferencia (TT) y máximo retardo permitido.

3.2.2. Sincronización

Una base de tiempo de alta precisión es importante en muchos sistemas distribuidos. Por ejemplo, en una subestación, los parámetros de corriente y voltaje se miden mediante sensores que se encuentran distribuidos por toda la subestación, y los datos de los diferentes sensores son transmitidos a un centro de control para su análisis. Para poder operar con esta información, los datos obtenidos por los sensores deben ser coincidentes en el tiempo, dado que un retraso producido en la transferencia de los datos desde un sensor provocaría un cálculo erróneo. Por

Tabla 3.2: Tipo de mensajes y clases de rendimiento

Tipo de mensaje	Clase de rendimiento	Tipo de mensaje	
		Clase	Tiempo *(ms)*
1A	P1	TT6	≤ 3
	P2	TT5	≤ 10
1B	P3	TT4	≤ 20
2	P4	TT3	≤ 100
3	P5	TT2	≤ 500
	P6	TT1	≤ 1000
4	P7[a]	TT6	≤ 3
	P8[b]	TT5	≤ 10
5	P9	TT0	≤ 10000
6	P10[c]	TT2	≤ 500
	P11[d]	TT1	≤ 1000
	P12[e]	TT0	≤ 10000
[a] *equivalente a P*1			
[b] *equivalente a P*2			
[c] *equivalente a P*5			
[d] *equivalente a P*6			
[e] *equivalente a P*9			

lo tanto, para obtener un cálculo correcto, es necesario incorporar un mecanismo que permita identificar el momento exacto de la captura de los datos.

El estándar *IEC 61850* no especifica ningún método de sincronización, lo que define el estándar son varias clases de rendimiento para la sincronización de *IEDs*. Las precisiones para estas clases van de $10\,ms$ a $1\,\mu s$, en la Tabla 3.3 se muestran las diferentes clases definidas para sincronización y la precisión necesaria.

Tabla 3.3: Clases de rendimiento para la sincronización de *IEDs*

Clase de sincronización	Precisión *(μs)*
TL	> 10000
T0	≤ 10000
T1	≤ 1000
T2	≤ 100
T3	≤ 25
T4	≤ 4
T5	≤ 1

Las primeras clases de sincronización de tiempo (TL, T0) se utilizan en aplicaciones con bajos requerimientos de sincronización. T1 y T2 se definen para eventos que no son de tiempo crítico, por ejemplo datos *SCADA*, datos de registradores de eventos y perturbaciones, entre otros. El resto (T3, T4 y T5) se definen para los eventos de tiempo críticos relacionados con las operaciones de protección y control, tales como el muestreo de señales, sincrofasores y disparos.

En este sentido, considerando los métodos de sincronización analizados en la sección 2.4.2, el estándar *IEEE 1588 PTP* es la mejor opción para ser considerada, ya que permite una sincronización de tiempo menor a $1\mu s$, necesaria para la clase T5. La funcionalidad *IEEE 1588 PTP* se puede implementar en *software*, *hardware* o mixtas, la elección dependerá de la precisión que requiere el sistema.

Para una implementación de *PTP* en *software* unicamente es necesario un *PS* con recursos básicos para ejecutar un sistema operativo, por ejemplo *Linux*, en la Figura 3.3, se muestra una implementación de *PTP* en *software*. Las librerías *PTP* acceden a los mensajes *PTP* y al valor del reloj en tiempo real *(Real Time Clock, RTC)* a través del *driver* del controlador de red que se ejecuta en el *kernel (Linux)*. Como consecuencia, si los eventos producidos en el controlador de red son administrados por *software*, la precisión se deteriorará seriamente debido a las variaciones de tiempo en el acceso de lectura al *RTC* y la carga de trabajo del *software*. De manera similar, las señales *PPS* que se usan comúnmente para monitorear la precisión de la sincronización *PTP* no se pueden generar con precisión.

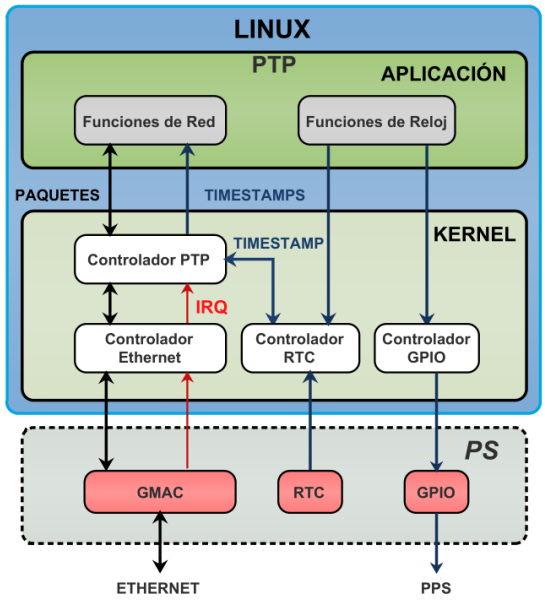


Figura 3.3: Estándar *IEEE 1588*: Implementación en *software*

En las soluciones mixtas, es necesario añadir *hardware* adicional que agilice el

proceso de captura de los datos que necesita el *software PTP* para realizar los cálculos. Por ejemplo un temporizador de mayor precisión y un módulo que permita capturar los mensajes *PTP*. En este sentido, es necesario plantear el uso de *SoC* modernos que integren en el mismo silicio la unidad de procesamiento y el área reconfigurable (*FPGA*). En la Figura 3.4 se muestra una implementación de *PTP* mixta. En la *CPU* además del *software PTP*, es necesario añadir controladores para acceder al *hardware* adicional, lo que dificulta su implementación. Por el contrario, las señales *PPS* que se usan para monitorear la precisión de la sincronización *PTP* son fáciles de implementar en el *hardware* adicional.

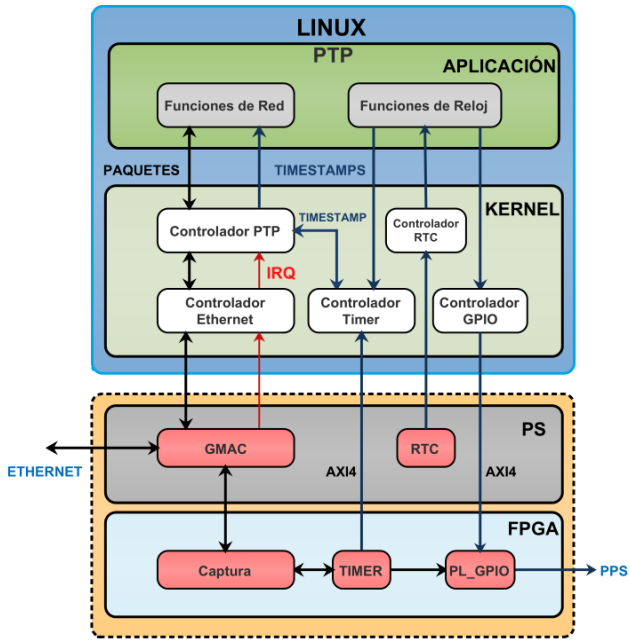


Figura 3.4: Estándar *IEEE 1588*: Implementación mixta

Finalmente, en la Figura 3.5 se plantea una arquitectura con un enfoque más preciso para *IEEE 1588*. En esta configuración la funcionalidad, la lógica de cálculo y marcado de tiempo se implementa directamente en *hardware*. En la actualidad existen controladores de tarjetas de interfaz de red *(Network Interface Cards, NIC)* que pueden realizar el *timestamp IEEE 1588* a nivel de *hardware*. Otro método es utilizando *IP cores* para ser utilizadas en implementaciones basadas en *FPGAs*.

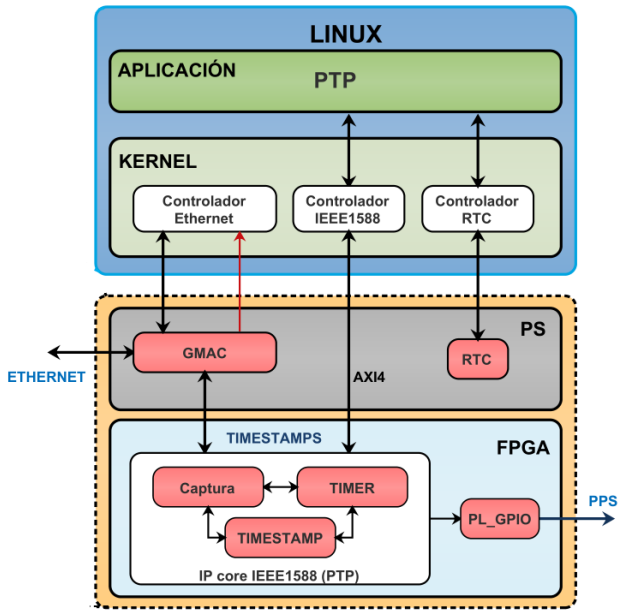


Figura 3.5: Estándar *IEEE 1588*: Implementación en *hardware*

3.2.3. Tiempo de restablecimiento: Alta disponibilidad

El tiempo durante el cual la subestación tolera una interrupción de un servicio se denomina "tiempo de gracia", y el tiempo de recuperación de la red debe ser inferior al tiempo de gracia. Por ejemplo cuando por el bus de estación unicamente se transporta información de comandos *(SCADA)*, según el estándar *IEC 61850-5* se toleran retrasos de unos 100 ms. Sin embargo, únicamente tolera un retardo de 4 ms cuando se transmiten señales de enclavamiento, disparo y bloqueo. A nivel de proceso, los servicios que se ejecutan para la operación de la subestación son críticos, por lo que el servicio debe diseñarse de manera que no se produzca ninguna falla. Los tiempos de recuperación que se especifican en el estándar *IEC 61850-5* se resumen en la Tabla 3.4.

Para implementar una red de alta disponibilidad para las comunicaciones especificadas en las partes *IEC 61850-8-1 (MMS)* e *IEC 61850-9-2 (SV, GOOSE)*, en el estándar *IEC 61850*-90-4 se plantea el uso de los protocolos de comunicación redundante *Rapid Spanning Tree Protocol (RSTP), PRP* y *HSR*.

Tabla 3.4: Tiempos de recuperación según el estándar *IEC 61850-5*

Comunicaciones	**Tiempos de recuperación Aplicación** *(ms)*	**Tiempo de recuperación Comunicación** *(ms)*
Cliente-Servidor Centro de control - IED	800	400
NTP, SNMP, PTP	500	300
IED - IED Bloqueo inverso, enclavamineto	12	4
Mensaje *GOOSE* de disparo (*trip*)	8	4
Protecciones (*Bus - Bar*)	< 1	Ininterrumpida
SV	<2	0

RSTP no asegura cero segundos de tiempo de recuperación, pero es lo suficientemente rápido para la mayoría de las aplicaciones que se ejecutan a nivel de estación. Por otra parte, el nivel más alto de disponibilidad se logra utilizando *HSR* y *PRP*. Estos protocolos son los únicos que ofrecen un tiempo de recuperación cero frente a un fallo en la red, esto supone que no existirá pérdida de información [54, 56, 58, 63]. Además permiten realizar la conexión en caliente *(hot-plugging)*, que implica poder conectar y desconectar dispositivos sin afectar al funcionamiento del resto de la red. Ambos ofrecen un robusto mecanismo de monitorización y control de la red, que está gestionado de forma automática por todos los nodos. Estas tres características hacen que los protocolos *HSR* y *PRP* sean excelentes candidatos para ser utilizados como mecanismos de redundancia.

HSR o *PRP* pueden ser implementado en *software,* para lo cual es necesario tomar como base el sistema operativo *Linux*, en especial el *kernel*. En el *kernel* de *Linux* a partir de la versión v3.12 en el subsistema de red se ha agregado un controlador para dar soporte al protocolo *HSRv0 (IEC 62439-3: 2010)*, y en la versión v4.6 se ha añadido soporte para *HSRv1 (IEC 62439-3: 2012)*. Este controlador le permite al usuario crear un dispositivo de red *HSR* utilizando un par de interfaces *Ethernet* estándar (A y B). A nivel de aplicación con el paquete *iproute2* (comando *ip link*), el usuario puede configurar una interfaz *HSR* que empareje las dos interfaces *Ethernet* para crear un nodo *HSR* o *DAN-H* [64]. En cuanto al protocolo *PRP* no hay ningún controlador nativo de *Linux*, sin embargo, existen librerías desarrolladas por investigadores o empresas que permiten la implementación del protocolo *PRP*. Por ejemplo, *Texas Instument* ha mejorado el controlador *Linux HSR* y el comando *ip link* en el paquete *iproute2* para hacerlo compatible con *PRP* en sus procesadores [64]. Con este cambio, es posible crear un nodo *PRP (DAN-P)* utilizando interfaces *Ethernet* estándar. El Instituto de Sistemas Embebidos *(Institute of Embedded Systems, InES)* de la Universidad de Ciencias Aplicadas de Zurich también proporciona un controlador *PRP* para ser agregado al *kernel* de *Linux* [65].

En la Figura 3.6 se observa una arquitectura con capacidad de manejar comunicaciones de alta disponibilidad implementada en *software*. El controlador *Linux HSR / PRP* proporciona una interfaz *Ethernet* estándar para la capa de aplicación y oculta los detalles del protocolo de redundancia que se implementa debajo de la capa del controlador. En resumen en el controlador *HSR/PRP* (*kernel* de *Linux*) se implementa la entidad de redundancia de enlace *(Link Redundancy Entity, LRE)*. Para el caso de *PRP*, la *LRE* contiene la lógica que permite duplicar las tramas *Ethernet* que se transmiten y a su vez descartar una de las tramas que se reciben [66]. Para *HSR* debido a que los nodos se conectan en una configuración en anillo es necesario añadir una funcionalidad extra para reenvío de tramas, es decir la *LRE* actúa como puente para las tramas que no interviene como origen o destino.

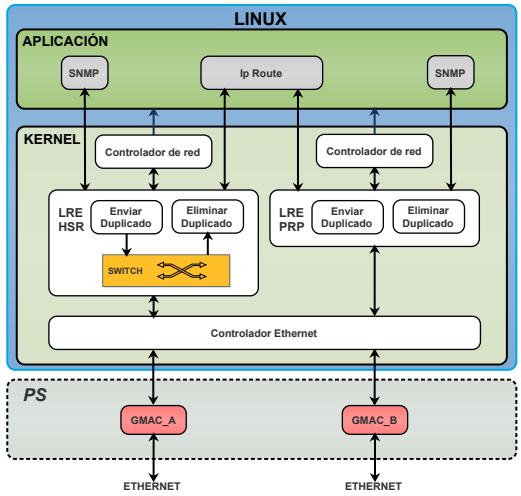


Figura 3.6: Implementación en *software* de *HSR* y *PRP*

La implementación en *software* de *HSR* o *PRP* es rápida y sencilla, pero tiene la desventaja de que sobrecargan al procesador a la hora ejecutar el proceso *(LRE)* que gestiona la duplicidad, el descarte (*PRP* y *HSR*) y el reenvío (*HSR*) de las tramas. A medida que aumenta la velocidad de transmisión, también aumenta la sobrecarga del procesador, de modo que los procesos secundarios que se están ejecutando experimentarán una reducción del rendimiento. Para el caso de *HSR*, no es posible realizar una implementación en *software* que sea *"cut-through"*, es decir, retransmitir un paquete antes de haberlo recibido por completo, por lo que provoca grandes latencias variables.

En la literatura también se han identificado varias soluciones que permiten implementar los protocolos *HSR* y *PRP* a nivel de *hardware*. Un ejemplo es el *SoC AM572x Sitara* fabricado por *Texas Instruments* [67], es un dispositivo multinúcleo que posee un procesador de propósito general *(ARM-A15)* y una unidad especializada *(PRU-ICSS)* para el procesado de tramas *Ethernet* y la ejecución de los protocolos *HSR* y *PRP*. Esta configuración resuelve el problema de la sobrecarga en el procesador de propósito general, pero las funcionalidades adicionales que se puedan implementar esta limitada a los recursos que integre el dispositivo. En este contexto, un enfoque que permite mayor flexibilidad en el diseño de dispositivos es el uso de *IP cores* en dispositivos reconfigurables.

3.2.4. Interoperabilidad

Las organizaciones internacionales centran sus esfuerzos en establecer estándares con el objetivo de unificar la arquitectura de los sistemas de comunicaciones Por ejemplo, en el sector eléctrico se han planteado el estándar *IEC 61850* [24], en el industrial el estándar *IEC 60870* [43], entre otros. En un entorno industrial, la interfaz y el protocolo de comunicación se denominan buses de campo *(fieldbus)*. Se pueden identificar dos tipos de bus de campo: el bus de campo basado en *Ethernet* y el bus de campo serie.

Un bus de campo basado en *Ethernet* se utiliza en entornos donde se requiere mayor ancho de banda, mayor rendimiento, operación en tiempo real y la capacidad de conectar más nodos a la red en distancias más largas. Ejemplos de buses de campo *Ethernet* son los siguientes: *Profinet, EtherCAT, Modbus/TCP, Ethernet/IP, Powerlink Ethernet, Sercos III, CC-Link IE,* entre otros [68].

Por el contrario, en aplicaciones de baja velocidad de datos y redes con pocos dispositivos, se utiliza el bus de campo serie. Los estándares *RS-485* y *CAN* son la base para la implementación de este tipo de bus de campo. Ejemplos de bus de campo en serie son: *Modbus, Profibus, Interbus, DeviceNet, CC-link*, entre otros [68].

Para garantizar la interoperatividad es necesario que los *IEDs* integren varias interfaces de comunicaciones seriales y *Ethernet* que den soporte a los diferentes buses de campo, así como una unidad de procesamiento que ejecute las aplicaciones de los diferentes protocolos de comunicaciones. También es necesario un puerto *Ethernet* dedicado para que los *IEDs,* puedan establecer comunicación

con servicios compatibles con las tecnologías de la información *(TI)* (por ejemplo, *HTTP, UDP, SNMP, FTP, TCP/IP*). La Figura 3.7 muestra un diagrama de bloques de la arquitectura de un dispositivo para que de soporte la interoperatividad con otros dispositivos.

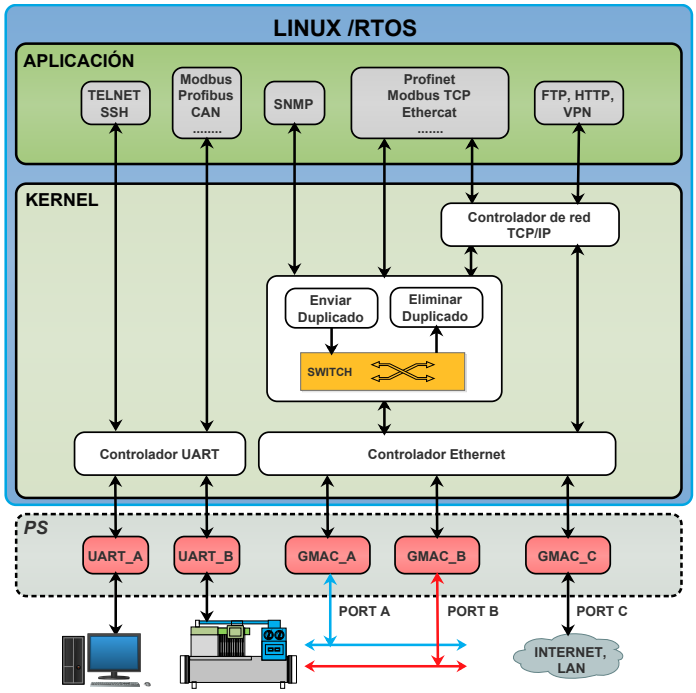


Figura 3.7: Arquitectura para dar soporte a la interoperatividad

En algunos casos los buses de campo basados en *Ethernet* para agregarles funcionalidades operativas en tiempo real es necesario agregar *hardware* adicional, que pueden ser módulos externos que integran las interfaces y pila del protocolo o *IP cores* para ser usados en implementaciones basadas en *FPGAs*.

Al día de hoy, la interoperabilidad obtenida en el sector industrial es limitada, siendo mayoritaria la parecencia de soluciones propietarias. Sin embargo, la necesidad de disponer de soluciones redundantes de distintos fabricantes en la infraestructura eléctrica ha forzado a que el nivel de interoperabilidad en el sector sea alto y que la aplicación de IEC 61850 sea una realidad.

3.2.5. Reconfigurabilidad

A medida que se agrega más funcionalidad a los dispositivos que están conectados a la red, la configuración remota de los diferentes elementos *(IP cores)* que se implementan en la *FPGA* es una opción cada vez más atractiva. En este caso, una alternativa es incluir en el diseño un módulo que permita la administración de la comunicación remota. Entre otros, en [69–72] proponen implementar en la *FPGA* un módulo de comunicaciones basado en microprocesador (por ejemplo un *Microblaze*) para administrar las tramas *Ethernet* de configuración. El procesador proporciona tanto la interfaz como las rutinas de *software* para transferir los datos de configuración, Figura 3.8. Esta solución tiene la desventaja de que es costosa en términos de recursos físicos y potencia requerida. Además, la complejidad del diseño aumenta y la velocidad de configuración disminuye.

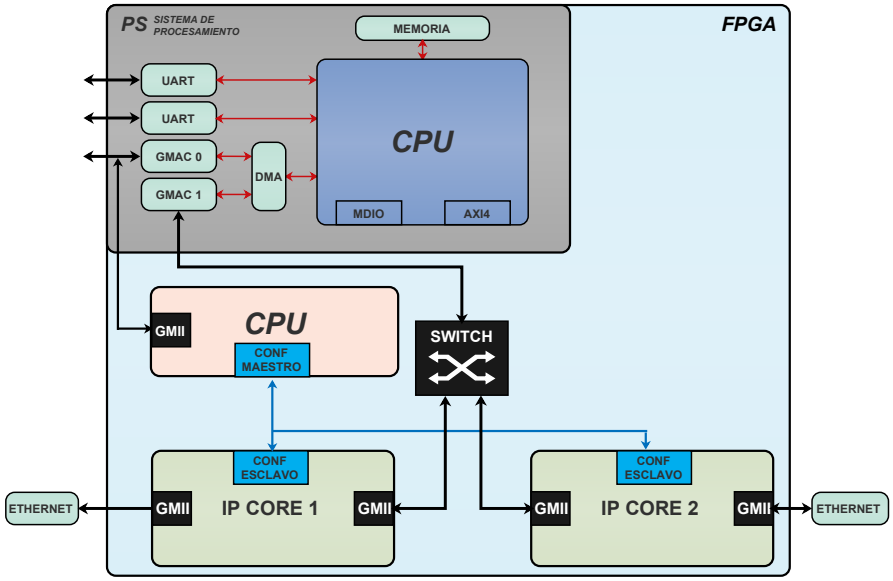


Figura 3.8: Configuración remota *(FPGA)* empleando un microprocesador

3.2.6. Ciber-seguridad

Con la implementación del estándar *IEC 61850,* las operaciones de control y supervisión de los *SAS* se están digitalizando. En este contexto, la interconexión de los equipos de control está evolucionando de soluciones propietarias a redes de comunicación estandarizadas. Esta evolución reduce los costes de implementación, pero también abre nuevas vulnerabilidades digitales, ya que los paquetes de información pueden ser fácilmente capturados, modificados, almacenados y reproducidos. Por ejemplo, en 2003 una planta nuclear en Ohio sufrió una vulnerabilidad en su sistema de control de la red al quedar infectado por el gusano *Slammer*. Este ataque desactivó el sistema de vigilancia de seguridad durante casi cinco horas [73]. Más recientemente, un ataque serio a la red eléctrica de Ucrania ha puesto de manifiesto la importancia de hacer cumplir las políticas de seguridad en las infraestructuras críticas [74]. El *North American Equipment Council (NREC)* también publicó sobre los ataques del gusano *Slammer*, detallando la manera en que los sistemas fueron infectados y los efectos que causó sobre los servicios públicos de energía en América del Norte. Por ejemplo, "el gusano migró a través de una conexión *VPN* a la red corporativa de una empresa hasta que finalmente llegó al área crítica de la red de control, supervisión y adquisición de datos *(SCADA)*" [75, 76].

Con estas premisas, la ciber-seguridad en las infraestructuras críticas se ha convertido en una prioridad para muchos gobiernos. En este sentido, se han puesto en marcha programas específicos de seguridad para definir métodos de protección de los datos que se utilizan para gestionar la operación de la red eléctrica.

Para avanzar en soluciones de seguridad interoperables, el grupo de trabajo 15 del *IEC TC57* trabaja en el desarrollo de estándares de ciber-seguridad para las comunicaciones de sistemas eléctricos, con el objetivo de cubrir tanto la infraestructura de la información como la seguridad de las comunicaciones. Uno de esos estándares es el *IEC 62351* que en su parte 6 plantea los mecanismos de seguridad para las comunicaciones bajo el estándar *IEC 61850*.

Los mecanismos de seguridad que plantea el estándar *IEC 62351-6* es en general un buen punto de partida para ayudar a asegurar las comunicaciones *IEC 61850*. Sin embargo, en la primera edición del estándar existen algunas deficiencias. La principal dificultad es alcanzar el rendimiento definido en *IEC 61850* para los datos *GOOSE* y *SV* añadiendo mecanismos de seguridad. En primer lugar, el estándar sugiere que el cifrado no debe utilizarse para la confidencialidad de los

datos en aplicaciones que utilizan mensajes *GOOSE/SV* que requieren tiempos de transferencia menores de 3 *ms*. Según el estándar sostiene que, puesto que la comunicación *GOOSE/SV* se limita a una *LAN* interna, el proceso de selección de la ruta de comunicación garantizaría la confidencialidad de los datos. Sin embargo, sugiere mecanismos para garantizar la integridad de los datos. Todos los demás métodos de comunicación *(MMS)* que no tengan las restricciones de tiempo implementarán tanto la integridad de los datos como las medidas de confidencialidad de los mismos.

El estándar *IEC 62351-6* para garantizar la integridad de los datos de las tramas *GOOSE* y *SV* plantea el uso sistema de encriptación de clave publica *Rivest, Shamir y Adleman (RSA)* para el intercambio de claves, *Advanced Encryption Standard (AES)* para el cifrado de datos y *Hash-based Message Authentication Code (HMAC)* para la autenticación de mensajes.

Sin embargo, el método de firma digital *RSA* tienen tiempos de ejecución largos que no permiten satisfacer los requisitos de tiempo que establece el estándar *IEC 61850* [25]. A pesar de que en las pruebas para validar el método de encriptación RSA se hayan utilizado procesadores *ARM* de gama alta con un *crypto* acelerador *(IP cores, ASIC)*, la firma *RSA* con claves de 1024 bits en algunos casos no puede ser calculada y verificada dentro de los 3 *ms*, que es el tiempo máximo de transferencia requerida por algunos mensajes *GOOSE* [26]. En otros casos aunque se logra realizar los cálculos cumpliendo las restricciones de tiempo, esta soluciones son poco viables especialmente por el costo. En la Tabla 3.5 se muestra un resumen del rendimiento de algunos dispositivos.

Tabla 3.5: Rendimiento en la ejecución del algoritmo *RSA(1024-bits)* sobre dispositivos basados en *IP cores*, *ASIC* y *SoC*

IP Core	Dispositivo	Tiempo (*ms*)
HelionRSA STD256 [77]	*FPGA* Xilinx UltraScale	14.9
Tiempo SecureTPKA [78]	*ASIC*	18.4
SILEX BA414EP [79]	*ASIC*	0.11
Athena TeraFire F5200 [80]	*FPGA* Xilinx Kintex 7	11.4
SafeXcel 1840 [81]	Co-Procesador	0.98
SafeXcel 3141 [82]	SoC	2.37

En la nueva versión del estándar *IEC 62351-6* que se publicará en su versión final el 05-marzo-2019, [41] se propone la utilización de criptografía simétrica en lugar de firmas digitales. El uso de criptografía simétrica tiene como finalidad minimizar el impacto negativo que tiene las medidas de seguridad sobre el rendimiento de los dispositivos de campo.

3.3. Resumen

Disponer de un PS que procese los mensajes de forma rápida no garantiza el funcionamiento de un sistema en tiempo real. También es necesario establecer algún mecanismo para sincronizar la información que se procesa y transmite con el resto de los dispositivos de la red. En este sentido, el estándar *IEEE 1588 Precision Time Protocol (PTP)* es la mejor opción, ya que ofrece una precisión inferior a $1\,\mu s$ y existe la posibilidad de ser implementada en *software*, *hardware* o mixta. Para una implementación de *PTP* en *software* sólo es necesario un *PS* con recursos suficientes para ejecutar la librerías *(standalone)* o un sistema operativo que sirva como base para ejecutar el software *PTP*, pero tiene la desventaja de que a medida que se añaden más funcionalidades al *PS*, la sincronización se degrada. Por otra parte, la implementación en *hardware* se puede hacer a través de interfaces *NIC* que integran las funcionalidades *PTP* o a través de *IP cores* para ser utilizados en *FPGAs*.

En los *SAS*, además de una sincronización de tiempo inferior a $1\,\mu s$, existen otros problemas críticos, como la necesidad de una red de alta disponibilidad que evite la pérdida de información en caso de falla del enlace de comunicaciones (tiempo de restablecimiento, sección 3.2.3). En este sentido, el mayor nivel de disponibilidad se consigue utilizando los protocolos *HSR* y *PRP* definidos en la norma *IEC 62439-3*, que se han descrito en el apartado 2.4.3. Estos protocolos ofrecen un tiempo de recuperación cero ante un fallo de la red, lo que significa que no habrá pérdida de información. Similar a la implementación de *PTP*, esta funcionalidad puede ser implementada a nivel de *software* y *hardware*.

Para garantizar la interoperatividad, es necesario que la arquitectura *CPS Gateway* integre varias interfaces de comunicaciones seriales y *Ethernet* para dar soporte a los diferentes buses de campo.

Los mecanismos de seguridad proporcionados por el estándar *IEC 62351-6* son generalmente un buen punto de partida para ayudar a proteger las comunicaciones *IEC 61850*. Sin embargo, hay algunas falencias en la primera edición del estándar. Por ejemplo, para garantizar la integridad de los datos en las tramas *GOOSE* y *SV*, el estándar propone el uso de códigos de autenticación de mensajes *(MAC)* utilizando el algoritmo computacional *SHA*, los cuales son firmados digitalmente mediante *RSA* para proporcionar la autenticacíon de origen. Sin embargo, el método de firma digital *RSA* tiene largos tiempos de ejecución que no cumplen con los requisitos de tiempo *(3 ms)* establecidos en el estándar *IEC*

61850. Por lo tanto, se propone el uso de criptografía simétrica para garantizar la seguridad de las comunicaciones de Capa 2 *(GOOSE y SV)*.

Capítulo 4

Cyber Physical Systems

4.1. Introducción

En este capítulo se realiza una introducción al estado del arte de los Sistemas ciber-físicos. Se resumen algunas definiciones planteadas por varios investigadores y organismos referentes en la temática, se presenta un esquema general de un *CPS* y se realiza una descripción de sus partes. También, en la sección 4.3 se describen las características que deben tener los *CPS* y las áreas de conocimiento que pueden ayudar en el desarrollo de los *CPS*. En la sección 4.4 se describen los campos de aplicación en los que se están desarrollando los *CPS*. En la sección 4.5 se describen varias arquitecturas de dispositivos que pueden emplearse en la *Smart Grid* y se realiza una comparativa entre estas arquitecturas sobre la base de los requisitos de operación definidos en el capítulo 3.

4.2. *Cyber Physical System*

Los sistemas computacionales los podemos encontrar en todas las aplicaciones en donde es necesario realizar procesamiento de información. Por ejemplo, en el automóvil la presencia de estos sistemas es sorprendente. Los podemos encontrar

en el sistema de control del motor, en los frenos *ABS*, bolsas de aire (*airbag*), en el control de tracción y en los sistemas de audio y vídeo. En el hogar estos sistemas controlan el microondas, el refrigerador, el lavavajillas, y la televisión. Estos sistemas menos visibles se los denomina sistemas empotrados, y el *software* que en ellos se ejecuta es llamado *software* empotrado [83].

Los avances de la electrónica digital y las redes de comunicaciones, han permitido que los sistemas empotrados se encuentren interconectados, generando un nuevo grupo de sistemas que se encargan de actividades más complejas, a estos sistemas se los denomina Sistemas ciber-físicos. El término "*Cyber Physical System*" (*CPS*) tiene su origen en el año 2006, y fue acuñado por Helen Gill perteneciente a la *"National Science Foundation"* con sede en los Estados Unidos, el cual hace referencia a la integración de la computación con los procesos físicos [83,84]. Una diferencia clave entre un sistema empotrado puro y un *CPS* es la existencia de una red de comunicación que interconectan los sistemas de computación. Por lo tanto, un *CPS* puede generalmente ser definido como sistema de sistemas.

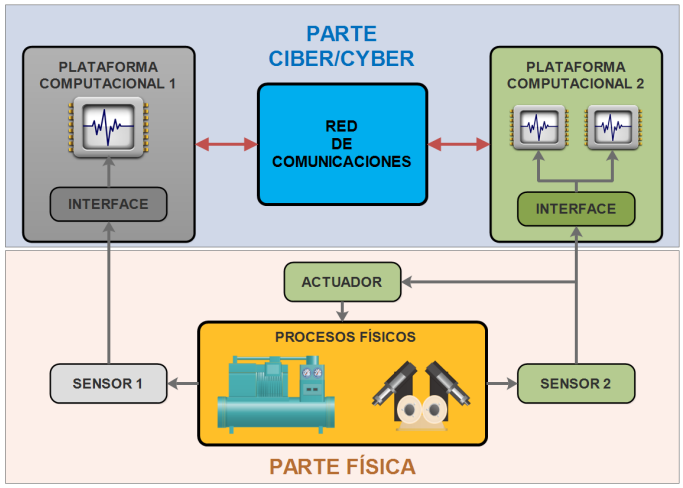


Figura 4.1: Arquitectura de un *CPS*

Existen muchas definiciones sobre los *CPS*, por ejemplo Lee [85], describe a los *CPS* como sistemas que integran la computación y los procesos físicos. Rajkumar [4] define a los *CPS* como sistemas físicos y de ingeniería cuyas operaciones son supervisadas, coordinadas, controladas e integradas en un núcleo de computación y comunicación. Jifeng [86] define a los *CPS* como sistemas *"3C"* que integran Computación, Comunicación, y Control. Baheti [87] describe a los *CPS*

como a una nueva generación de sistemas que integran capacidades computacionales y físicas, que pueden interactuar con los seres humanos a través de nuevas modalidades.

Resumiendo, los *CPS* son sistemas complejos, multidisciplinarios, se caracterizan por integrar componentes físicos (sensores, actuadores), procesamiento (control, análisis de datos) y comunicaciones, que pueden interactuar con otros sistemas y con los seres humanos [88, 89]. En la Figura 4.1 se muestra un diagrama de la arquitectura típica de un *CPS*, en este esquema podemos identificar tres partes principales.

En primer lugar, la parte "física", se refiere a los fenómenos físicos que requieren supervisión o control, es decir la planta de un sistema, que puede incluir elementos mecánicos, procesos biológicos o químicos u operadores humanos. En segundo lugar, tenemos la parte de las plataformas de procesamiento computacional, que constan de sensores y actuadores (conectores entre la parte "Física" y *"Ciber/Cyber"*), uno o más microprocesadores, y dependiendo de la complejidad uno o más sistemas operativos. En tercer lugar, hay una estructura de comunicaciones, que proporciona los mecanismos para que las diferentes plataformas de procesamiento intercambien información. En conjunto, las plataformas de procesamiento y la estructura de comunicaciones, conforman la parte *"Ciber/Cyber"* de un *CPS*. [83, 84]

Cabe señalar que los *CPS* son más frecuentes y relevantes hoy en día, en gran parte debido a la necesidad de utilizar más y más información en el control de los procesos industriales. Además, los nuevos avances tecnológicos en el campo de la electrónica, las tecnologías de la información y la informática hacen posible la construcción de este tipo de sistemas [90]. Esta nueva generación de dispositivos inteligentes se está introduciendo tanto para aplicaciones domésticas como industriales, también conocido como *Smart Connected Products (SCP)*. Desde un punto de vista tecnológico, los *SCP* domésticos e industriales comparten sus principales características (*hardware*, *software* y conectividad), lo que los diferencia es la cantidad de datos que pueden procesar y los tipos de interfaces de comunicación que integran [91]. Estos dispositivos necesitan nuevas infraestructuras tecnológicas, como redes para conectarse a la nube, protocolos de comunicación para intercambiar datos entre máquinas y con usuarios, herramientas de *Cloud* para generar nuevas aplicaciones para el fabricante o su ecosistema, lo que permite el desarrollo de nuevos modelos de negocio. Las plataformas de desarrollo deberían facilitar el desarrollo de aplicaciones de una manera más rápida, acelerando el *"Time to Market"*, reduciendo el riesgo del proyecto y facilitando la evolución

de las aplicaciones en entornos cambiantes como el sector industrial [92].

Además, la mayoría de los *CPS* requieren la integración de diferentes tecnologías y metodologías de diseño utilizadas en diversos dominios, como electrónica, comunicación, control, redes inalámbricas, informática distribuida, sistemas en tiempo real, seguridad e ingeniería orientada a modelos [93]. Algunos de los conceptos y tecnologías, como las redes inalámbricas y las comunicaciones basadas en *Ethernet*, que se utilizan en otros entornos *Smart* (por ejemplo, *Smart-Cities*, *Smart Grid*) se pueden usar directamente en el sector industrial. Sin embargo, es imprescindible tener en cuenta los problemas y necesidades particulares del sector para ofrecer soluciones que sean plenamente aceptadas por los agentes involucrados. Estas claves son los requisitos en tiempo real *(RT)*, alta disponibilidad, interoperabilidad, estandarización, flexibilidad, seguridad y ciber-seguridad. [94, 95]. En este sentido, un *CPS* puede cumplir estos requisitos, asegurando un procesamiento determinista, flexibilidad, comunicaciones y operaciones de alta disponibilidad.

Los *CPS* son necesarios para el desarrollo del Internet industrial de las cosas (*Industrial Internet of Things, IIoT)* [96]. En la *IIoT* a de más de las características de operación en tiempo real, alta disponibilidad, interoperabilidad y ciber-seguridad que los *CPS* proporcionan, el IIoT también aprovecha la comunicación máquina a máquina *(Machine-to-Machine, M2M)* y la enorme cantidad de datos que los dispositivos generan para incorporar el aprendizaje automático y las tecnologías *Big Data* en el sistema de producción [97]. La filosofía detrás de *IIoT* es que las máquinas, con un alto nivel de inteligencia, son mejores que los humanos para capturar y comunicar datos de forma precisa y consistente. Con estos datos, las empresas pueden detectar más rápidamente las fallas y los problemas, incluso antes de que ocurran, ahorrando tiempo y dinero. Particularmente, en la industria energética, el *IIoT* permitirá un mejor control del suministro de la energía, el sistema será más eficiente, con el objetivo de lograr una producción de electricidad sostenible y ecológica.

4.3. Características de un *Cyber Physical System*

Para diseñar y operar *CPS* que puedan ofrecer nuevas funcionalidades, se necesitan conocimientos y tecnologías de una amplia variedad de campos, muchas de estas tecnologías no se han desarrollado especialmente para los *CPS* por lo que

hay que adaptarlas a los requerimientos de los *CPS*, por otra parte, también hay tecnologías que tienen que ser desarrolladas específicamente para los sistemas ciber-físicos.

Esta sección resume las características que deben tener los *CPS* y las áreas de conocimiento que pueden ayudar en el desarrollo de nuevas capacidades para los *CPS* [98]. Se hace hincapié expresamente en los principios y fundamentos básicos, lo que demuestra el difícil reto que supone reunir el conocimiento necesario para abarcar tanto aspectos físicos como computacionales. Por ejemplo, hay grandes posibilidades de desarrollo en:

- Tecnologías de actuadores y sensores. La capacidad para detectar y reconocer objetos y variables físicas, es clave en los *CPS*. Con la información proporcionada es posible realizar un análisis de la situación actual de todos los actores que intervienen en la ejecución de la aplicación, incluso considerando su propio estado, y tomar decisiones para garantizar que el sistema esté funcionando al máximo de sus capacidades [99]. Debido a que los sensores son un vínculo de hardware entre el mundo físico y el mundo cibernético, es importante entender las propiedades de los sensores y su comportamiento en el mundo real, así como las técnicas para procesar las señales que producen [100]. Un concepto clave en el diseño de un *CPS* es la fusión de sensores, que se refiere a la combinación de varios sensores con la finalidad de obtener mediciones más precisas o datos de orden superior. Esta técnica se utiliza para detectar y corregir mediciones erróneas que se pueden producir al utilizar sensores individuales, así como para hacer inferencias sobre el estado del sistema que solo serán posibles utilizando varios sensores. Por ejemplo proyección, deducción y predicción [99].

- Redes de comunicaciones. Para el diseño de un *CPS* es necesario una comprensión detallada de todos los niveles por donde se establece el intercambio de la información, desde los principios de la capa física, hasta los protocolos con restricciones de tiempo que se ejecutan en la Capa 2 del modelo *OSI*. En un *CPS* y especialmente en los de gran escala, los sistemas estarán compuestos por componentes de diferentes proveedores, por lo que es necesario desarrollar estándares de comunicaciones y definir arquitecturas comunes, con el objetivo de garantizar la interoperatividad entre componentes y sistemas heterogéneos [84, 101].

- Ciber-seguridad. Todos los sistemas basados en tecnología de la información están sujetos a ataques cibernéticos. Muchos *CPS* son especialmente

vulnerables, ya sea porque están ubicados en entornos abiertos, las comunicaciones no son cifradas o utilizan comunicaciones fácilmente interceptables como las inalámbricas. Los *CPS* son vulnerables a fallas y ataques tanto en el lado físico como en el cibernético, debido a su escalabilidad, complejidad y naturaleza dinámica. El uso de una red a gran escala (como Internet), la adopción de protocolos de comunicación inseguros, el uso intensivo de sistemas heredados o la rápida adopción de tecnologías comerciales son otros factores que hacen que los *CPS* estén fácilmente expuestos a las amenazas a la seguridad [84]. Asegurar que los diseñadores de los *CPS* estén familiarizados con los riesgos de la seguridad, privacidad, y las técnicas para protegerlos serán cruciales [100].

- Fiabilidad y estabilidad. Muchos *CPS* serán parte de nuestra vida diaria, y su utilidad requerirá una alta confiabilidad y estabilidad. Los mejores sistemas son aquellos diseñados desde el principio teniendo en cuenta la confiabilidad y no como algo que se deba arreglar cuando los fallos se presenten. Los *CPS* también deberán ser robustos ante los problemas (incertidumbre) que pueden ser difícil de cuantificar en la fase de diseño. Para garantizar la estabilidad de los *CPS* es necesario considerar en la fase de diseño factores tales como la linealidad o no linealidad del sistema, el ancho de banda de los sistemas, la tasa de muestreo, los polos y los ceros del sistema (modelo de planta), el modelado del ruido y las limitaciones de los sensores y actuadores [100].

- Procesamiento en tiempo real. Las complejidades del mundo real a menudo conducen a situaciones que no son abordadas por el *software* y a menudo resultan en fallas, existiendo la necesidad de herramientas informáticas de desarrollo que incorporen en sus algoritmos las limitaciones de tiempo del mundo físico. Por otra parte, muchos dispositivos *CPS* tienen una capacidad de computación, memoria y energía limitadas, por lo que los diseños e implementaciones de *software* deben ser conscientes de estas limitaciones, y es necesario una comprensión de temas como programación en tiempo real, la semántica temporal en los programas y la sincronización del reloj en las redes de comunicaciones. El conocimiento por parte de los diseñadores sobre los principios del *software* embebido, los muchos lenguajes de programación, algoritmos, diseño de *software*, métodos formales y plataformas (arquitecturas y sistemas operativos) son necesarios para permitir el desarrollo de *CPS* confiables y de alta calidad [102].

- Control distribuido. Los *CPS* deben cooperar entre sí para cumplir sus objetivos, esta capacidad de cooperar y negociar permite que los *CPS* proporcionen servicios distribuidos y comportamientos colectivos coordinados,

de esta manera se puede formular estrategias para resolver problemas de manera distribuida, coordinada y en tiempo real [98]. Las tecnologías necesarias que se tienen que desarrollar están relacionadas con el área de la interoperabilidad entre sub-sistemas, para lo cual es necesario disponer de interfaces y protocolos de comunicaciones, robustos y fiables, que puedan soportar el gran flujo de datos y en tiempo real.

- Interacción humana. Uno de los beneficios que ofrecen los *CPS* radica en su capacidad para apoyar en las acciones e intenciones de los seres humanos y realizar tareas en su nombre. Al mismo tiempo, también puede tomar algunas decisiones y realizan algunas acciones automáticamente, por consiguiente ejercen una influencia en la conducta humana y en procesos sociales [103, 104]. Aunque estas capacidades ofrezcan ventajas significativas, todavía hay una cantidad enorme de trabajo por realizar para poder dominar de manera aceptable la interacción hombre-maquina. Por ejemplo, en el diseño de *CPS* es necesario considerar el comportamiento de los humanos, considerar como interfieren los factores humanos en el lazo de control. Un problema importante es hacer que un *CPS* sea fácil de operar, controlar y mantener por el usuario [102].

- Análisis de datos. Los *CPS* deben adaptar su comportamiento basándose en a los requerimientos de su contexto actual. Para ello deben tener la capacidad de adquirir y construir conocimiento, utilizando por ejemplo experiencias exitosas de ejecuciones anteriores. Los *CPS* tienen que ser capaces de actuar autónomamente para cumplir con los objetivos que normalmente son establecidos por sus usuarios o se derivan de su situación actual. Un ejemplo de este comportamiento podría ser el de un *CPS* que pueda reconfigurarse automáticamente a las condiciones de su entorno, ajustar sus algoritmos de control de forma dinámica para compensar el desgaste de sus piezas y estimar cuando realizar su mantenimiento [104]. Tecnologías del campo de aprendizaje de máquinas por ejemplo redes neuronales, algoritmos genéticos y *Big Data* podrían ser utilizadas para dotar de capacidades de aprendizaje y adaptación a los *CPS* [98]. Por otra parte, estos métodos utilizan los grandes volúmenes de datos, obtenidos de los sensores y de los sistemas de cooperación para ejecutar tareas especificas, por lo que es necesario disponer de dispositivos electrónicos que permitan almacenar una gran cantidad de datos, y los más importante sistemas de comunicación que puedan transportar dicha información.

4.4. Aplicaciones

Con la evolución de la tecnología hacia una nueva revolución industrial denominada *Industry 4.0* [92,105–108] y el *IIoT*, estamos observando una gran variedad de investigaciones sobre los *CPS* en sectores tan diversos como la industria del transporte (automoción, aire, marítimo y aeroespacial) [109–111], control del tráfico aéreo, edificios inteligentes [112], red de energía eléctrica [113, 114], dispositivos médicos [115], ciberdefensa [116] y la industria.

Esta sección, resume una serie de iniciativas de investigación que abordan algunos de estos sectores desde la perspectiva de los *CPS*, por ejemplo en el transporte, la salud, la fabricación inteligente y las infraestructuras críticas como la red eléctrica. La Tabla 4.1 proporciona una visión general del uso de *CPS* en estos sectores y las posibles funcionalidades a implementar.

Tabla 4.1: Sectores de aplicación de los *CPS*

Aplicación	Funcionalidad
Transporte	*CPS* de mediana y gran escala. Objetivos: ▪ Mejorar la seguridad mediante sistemas de asistencia a la conducción (conducción autónoma). ▪ Coordinar los servicios de gestión del tráfico para optimizar el uso de recursos.
Salud	*CPS* de pequeña y mediana escala. Objetivos: ▪ Implementar sistemas de monitoreo de signos vitales y su integración con los centros de salud. ▪ Mejorar la calidad de vida de las personas con discapacidades mediante la implementación de prótesis automatizadas.
Infraestructura	*CPS* de mediana y gran escala. Objetivos: ▪ Administrar, controlar y monitorizar las diferentes infraestructuras criticas (Agua, luz, transporte público, etc) de una ciudad. ▪ Mejorar la eficiencia y el rendimiento de los sistemas de iluminación, calefacción y agua potable de los edificios.
Energía Eléctrica	*CPS* de mediana y gran escala. Objetivos: ▪ Aumentar la eficiencia y fiabilidad de los sistemas de producción, transmisión y distribución de energía eléctrica.

4.4.1. Transporte

En la actualidad un vehículo con funcionalidades estándar contiene alrededor de 25 a 35 microcontroladores, y algunos vehículos de lujo contiene aproximadamente entre 60 y 70 [117]. Entre los sistemas que se encargan de controlar estos microcontroladores tenemos: sistema de *airbags*, sistema antibloqueo de frenos, caja negra, control de velocidad crucero, sistema de radio satélite, la telemetría, el sistema de control de emisiones, control de tracción, aparcamiento automático, sistemas de entretenimiento, visión nocturna, sensores para prevenir colisiones, sistemas de navegación, monitor de presión de los neumáticos, sistema de climatizado, etc. Estos microcontroladores se encuentran conectados en red y disponen capacidades de procesamiento en tiempo real, por lo que podemos considerar a estos sistemas como *CPS*.

Los desarrollos que se están realizando en los automóviles en su gran mayoría están orientados a facilitar la conducción, brindar confort y seguridad, pero no están contribuyendo a atacar problemas "macro", como por ejemplo la congestión en las grandes ciudades. Por lo tanto será necesario diseñar *CPS*, que consideren el comportamiento del peatón, conductor, vehículo y la infraestructura por donde circulan los vehículos. Esto permitirá desarrollar sistemas de transporte con capacidades avanzadas de detección, comunicación, computación y mecanismos de control.

Existen muchos desarrollos de *CPS* en el sector del transporte, por ejemplo el proyecto *eCall* [118], una iniciativa de la Comisión Europea, que tiene como objetivo desarrollar un sistema de auxilio inmediato, en caso de colisión el sistema establecerá comunicación con el puestos de auxilio más cercano, si los ocupantes no pueden hablar, el sistema envía una trama de datos, en la que se incluye la hora y punto exacto donde ocurrió la colisión. El 28 de abril 2015, el Parlamento Europeo votó a favor de la regulación de *eCall*, por lo que a partir de abril 2018 todos los vehículos de nueva homologación fabricados en la Unión Europea están obligados a incorporar el sistema de llamada de emergencia *eCall*.

En los *CPS* para el trasporte el uso del sistema de posicionamiento global *(GPS)* es muy importante. Por ejemplo en [119] se plantea un sistema de comunicaciones entre vehículos de corto alcance asistido por *GPS*. El sistema dispone de dos transceptores. Uno de ellos está conectado a Internet, por el que se envía datos de la posición, la ruta que recorrerá el vehículo, y de los canales de comunicación que está ocupando el otro tranceptor. El segundo transceptor es el encargado de

realizar las comunicaciones con restricciones de tiempo. Utiliza los datos que se están transmitiendo por Internet para determinar la posición de los otros vehículos y el uso del espectro. De esta manera el sistema consigue estimar los canales de comunicación que pueden utilizar en su recorrido, evitando interferencias que puede reducir el rendimiento. En [120] plantean el uso del *GPS* del móvil para enviar datos sobre la posición, la velocidad y las condiciones del tráfico, de esta manera se puede establecer rutas alternativas para reducir la congestión.

El uso de sensores tales como acelerómetros y giroscopios han permitido desarrollar sistemas como el que se plantea en [121]. En este trabajo se realizaron simulaciones sobre perfiles de aceleración de vehículos en varios tipos de carretera, como resultado presenta un algoritmo que permite detectar anomalías en la superficie de la carretera. Esta información puede ser útil en los sistemas de trasporte inteligentes, ya que podrían comunicar a los vehículos que están por circular por esa ubicación sobre el estado de la vía. Los acelerómetros también han permitido desarrollar sistemas de seguridad en caso de colisión, como los sistemas de accionamiento de los *airbags* [122, 123], pretensores de los cinturones de seguridad [124] y también los encontramos en los sistemas de suspensión que se adaptan en función del peralte de la vía [125].

Los avances realizados en los *CPS* para el sector del trasporte no se basan únicamente en el uso de sensores de última tecnología o en el desarrollo de sistemas empotrados. También cabe destacar que los sistemas de comunicación, basados en tecnologías satelitales, celulares y *wireless*, son una pieza clave para el desarrollo de los sistemas de comunicaciones vehículo a vehículo *(V2V)*, vehículo a peatones *(V2P)*, y del vehículo a infraestructura *(V2I)*. Este tipo de comunicaciones permitirán gestionar mejor el flujo de tráfico, garantizar la seguridad y mejorar los sistemas de análisis y conocimiento del estado actual de todos los actores que intervienen en los sistemas de trasporte.

Con la integración de los *CPS* en vehículos, carreteras y las infraestructuras de control, gestión y monitorización del trafico, los sistemas de trasporte inteligente pueden realizar tareas de prevención de colisiones, mejores sistemas de asistencia a la conducción, notificaciones en caso de averías o accidentes, reducción de la congestión y reducir el tiempo de viaje sin temor a retrasos inesperados [84].

4.4.2. Salud

Los dispositivos médicos producidos para esta industria son diversos, que van desde simples dispositivos digitales de medición de temperatura y presión, hasta dispositivos más complejos como los equipos de asistencia vital, el instrumental quirúrgico, equipos de radiología, oftalmología, electroterapia y diagnostico por imagen, entre otros.

La investigación de *CPS* está revelando numerosas oportunidades y desafíos para la medicina y la ingeniería biomédica. Entre los retos planteados en este campo, cabe destacar hospitales y quirófanos inteligentes, cirugía guiada por imagen, control de flujo en la administración de medicina intravenosa, y el desarrollo de prótesis físicas y neuronales. El equipamiento médico para el cuidado de la salud se basa cada vez más en dispositivos que están conectados en red. Por ejemplo, dispositivos tales como bombas de infusión para la sedación, ventiladores y sistemas de suministro de oxígeno para el apoyo a la respiración, y una variedad de sensores para el control de la condición del paciente que se utilizan en muchas salas de operaciones. A menudo, estos dispositivos deben ser configurados dependiendo de las necesidades y condiciones de un paciente específico. Por lo tanto, se necesitan dispositivos y sistemas médicos que se reconfiguran dinámicamente, y que puedan interactuar con los pacientes y operadores en entornos complejos. El desafío es desarrollar metodologías de diseño que permitan obtener sistemas debidamente certificados, seguros y confiables. En el informe planteado por la *National Information Technology Research and Development (NITRD)* [126], recomienda realizar las investigaciones para alcanzar los siguientes objetivos:

- Sistemas médicos interoperables y con arquitecturas abiertas.
- Monitoreo y control distribuido, y comunicaciones inalámbricas en tiempo real para el área de cuidados intensivos de un hospital.
- Métodos de certificación para dispositivos médicos de monitorización y asistencia, tanto en *software* como *hardware*.
- Modelado basado en *frameworks* para el diseño, y utilización de modelos específicos de pacientes para pruebas y certificación.

Otra área para la investigación de los *CPS* en el sector de la salud está en el desarrollo de interfaces entre el cerebro y las maquinas, para el desarrollo de aparatos ortopédicos, prótesis robóticas y exoesqueletos [127, 128].

En [103] se nos presenta un análisis del estado del arte sobre el desarrollo de *CPS* orientados a la salud y la medicina. Por ejemplo en [129], se presenta una investigación centrada en la obtención de señales de signos vitales de manera automatizada, e integrarlos en las historias clínicas electrónicas de los pacientes, denominados *Electronic Medical Records (EMR)*. Los autores de [130] proponen una arquitectura para análisis de datos utilizando *Big Data* tomando como ejemplo un sistema de monitorización remota, que utiliza varios sensores como medidor de presión, ritmo cardíaco, nivel de oxigeno y ECG. Estos datos son enviados a un servidor remoto que se encarga de procesar los datos y determinar el estado de salud del paciente. Wang y sus colaboradores [131] presentan una arquitectura que han denominado *CPeSC3*, que se compone de tres componentes principales, 1) núcleo de comunicación, 2) núcleo de cálculo, y 3) núcleo de planificación y gestión de recursos. Realizan un análisis de diferentes temas desde el punto de vista de la arquitectura propuesta, entre los que podemos mencionar: computación en la nube, programación en tiempo real, seguridad y redes de sensores.

En [132] proponen un mecanismo de seguridad para ser utilizados en los sistemas de monitorización de la salud. Los autores toman como enfoque que estos sistemas de monitorización pueden considerarse un *CPS*, que debe ser protegido. Esta propuesta denominada *Cyber Physical Security (CYPSec)* usa un método de generación de claves utilizando como fuente de aleatoriedad las señales fisiológicas del cuerpo. También proponen un control de acceso, que tiene la capacidad de proporcionar el acceso en caso de emergencias. *LiveNet* [133] es capaz de supervisar continuamente una amplia variedad de señales fisiológicas juntos con actividad del usuario, se basa una arquitectura modular con comunicaciones inalámbricas, con procesamiento de datos en tiempo real. Es utilizada para supervisar las actividades de personas que sufren enfermedades del corazón, parkinson, epilepsia, etc.

4.4.3. Infraestructura

Las ciudades inteligentes es otro ámbito prioritario de investigación que se ha planteado en la agenda de la Unión Europea (UE) en los últimos años. Propuesta por la UE la definición misma de ciudades inteligentes "*Smart Cities*" es universalmente reconocida como la referencia para este campo. "Una ciudad inteligente es un lugar donde las redes y los servicios tradicionales son más eficientes, en beneficio de sus habitantes y las empresas" [134]. Con esta visión, la

UE está invirtiendo en la investigación y la innovación de las *TIC* y en el desarrollo de políticas para mejorar la calidad de vida de los ciudadanos, para que las ciudades y sus infraestructuras sean más sostenibles y estén alineados con los objetivos 20-20-20 planteados por la UE [135]. Éstos establecen: (1) reducir las emisiones de gases de efecto invernadero en un 20 % con respecto a las cifras de 1990, (2) ahorrar el 20 % del consumo de energía mediante una mayor eficiencia energética, y (3) obtener al menos el 20 % del consumo energético a partir de fuentes renovables [136].

Las ciudades inteligentes deben gestionar una serie de subsistemas complejos como: redes de trasportaste inteligentes, sistemas de gestión de energía, servicios públicos (agua, electricidad, gas), manejo de residuos, etc. Estos subsistemas se caracterizan por la interacción directa con el mundo físico, por lo tanto una ciudad inteligente es un sistema formado por varios *CPS*. Los *CPS* pueden ser considerados como una de las tecnologías clave para el desarrollo de las ciudades inteligentes, ya que, antes de poder administrar, controlar y monitorizar las diferentes infraestructuras que conforman una ciudad inteligente, es indispensable contar con dispositivos que permitan recoger, procesar y transmitir un gran cantidad de información, a través de una gran variedad de sensores, por ejemplo: cámaras, sensores químicos, sensores de movimiento, sensores de infrarrojos, acelerómetros, *GPS* y sensores médicos entre otros.

Para cumplir la visión de los conceptos de red inteligente y ciudades inteligentes, es necesario construir edificios inteligentes, en donde el edificio puede producir tanto o más energía que la que consume. En un edificio podemos encontrar sistemas automatizados para el control de calefacción, ventilación y aire acondicionado, iluminación, control de acceso, control de ascensores y alarma contra incendios entre otros. Tradicionalmente, estos sistemas están diseñados para trabajar de manera independiente, es decir el comportamiento de un sistema no influye en los demás. Sin embargo, en la actualidad estos sistemas están siendo estrechamente integrados, con la finalidad de mejorar el control de los edificios, la seguridad, optimizar el uso de la energía, reducir los costos de operación y mantenimiento [137].

Las investigaciones en *CPS* no solo ofrece oportunidades de mejorar la eficiencia y el rendimiento en edificios comerciales y residenciales, también es aplicable en otras infraestructuras, como puentes, plantas de energía nuclear, estructuras *offshore*, represas, diques, refinerías y otras plantas químicas. En estas infraestructuras hay una necesidad crítica de desarrollar una tecnología que puede evaluar con precisión la integridad estructural, con la finalidad de estimar los periodos

de mantenimiento de la infraestructura para prolongar su vida útil y reducir los accidentes.

La interconexión global de estos subsistemas también genera retos relacionados con la seguridad. Tradicionalmente para conectar los diferentes dispositivos, existen dos redes de comunicación, una red de control y otra red troncal. En la red de control están conectados los sensores y actuadores, por los que se transmiten datos de los sensores, o para recibir parámetros de configuración. La red troncal es la encargada de interconectar varias redes de control, con el sistema central de procesamiento. La red troncal también esta conectada a redes externas como Internet, para dotar de servicios como control y monitorización remota. Por desgracia, este aumento de la conectividad y la posibilidad de controlar el sistema de manera remota, hacen que los edificios inteligentes sean vulnerables a los ataques de hackers [138].

En [139] presentan un análisis sobre los métodos que se están utilizando para dotar de mayor seguridad a los sistemas de comunicaciones utilizados en los diferentes sistemas que conforman un edificio inteligente. Las vulnerabilidades de seguridad en los protocolos de comunicación utilizados en los sistemas de automatización, por ejemplo *BACNet*, *LonWorks/LonTalk* y *KNX/EIB*, pueden ser utilizadas para comprometer el normal funcionamiento de un dispositivo de la edificación. Un dispositivo comprometido puede ser utilizado para interrumpir otros dispositivos o subsistemas, por medio de métodos de suplantación de identidad o el lanzamiento de ataques de denegación de servicio [140].

4.5. *Cyber Physical Systems* en la *Smart Grid*

4.5.1. Introducción

Muchos aspectos de nuestras vidas dependen de una producción, transmisión y distribución de energía eléctrica de manera fiable. Por ejemplo, las operaciones financieras, el transporte, los servicios de emergencia y las fabricas son dependientes de la disponibilidad del suministro eléctrico. La Academia Nacional de Ingenieros de los Estados Unidos ha calificado al desarrollo de la electrificación, como el mayor logro de la ingeniería del siglo XX. Sin embargo, a pesar de que la infraestructura eléctrica ha funcionado varias décadas sin mayores inconve-

nientes, necesita ser modernizada con la finalidad de aumentar su eficiencia y fiabilidad [102, 141].

La generación de energía eléctrica en gran medida proviene del uso de carbón, de la energía nuclear y de los combustibles fósiles. En la actualidad, debido a la preocupación por el calentamiento global, la generación eléctrica sé está enfocando en el uso de energías renovables. Por ejemplo,mediante parques eólicos, granjas de paneles solares, generadores marinos y sistemas de generación basado en geotermia. Por lo tanto, los sistemas energéticos del futuro serán cada vez más heterogéneos en los tipos de fuentes de energía y su localización [142]. Por otra parte, los volúmenes de energía producidos por estas instalaciones son extremadamente aleatorios, y no están en relación con el consumo requerido. Por lo tanto, con una cantidad cada vez mayor de sistemas de generación eléctrica con base en fuentes energías renovables, el esquema centralizado actual de gestión de la red eléctrica es cada vez más inadecuado. En este contexto, surge la necesidad de implementar un enfoque más distribuido. En este caso, un *CPS* en una red eléctrica inteligente, ofrece una solución a esta necesidad, ya que permite la coordinación descentralizada y cooperativa entre los diferentes procesos técnicos y administrativos. Por ejemplo, un *CPS* podría disponer de la información necesaria para controlar la generación de energía de una instalación fotovoltaica, en función de la comercialización y la facturación realizada [143].

La utilización de la tecnología de los *CPS* en la red eléctrica, hará que los sistemas de generación, trasmisión y distribución sean más inteligentes, eficientes y fiables. En el momento de diseñar *CPS* para el sector eléctrico es necesario considerar el comportamiento de los consumidores, ya que la cantidad de energía que se genera es impulsada por la demanda. Los productores de energía no disponen de mecanismos que sean económicos para almacenar el exceso de la energía producida, para lo cual se están realizando investigaciones sobre el almacenamiento de energía en estado estacionario, entre los que podemos mencionar baterías [144], volantes de inercia [145,146], superconductor de almacenamiento de energía magnética [147,148], almacenamiento de energía mediante de aire comprimido [149] y supercondensadores [150]. Hasta que no se disponga de elementos económicos de almacenamiento de energía, es necesario desarrollar mecanismos con bajos tiempos de respuesta, que permitan activar o desactivar los sistemas de generación en función de la demanda de energía. Un avance significativo en este campo es la utilización de contadores inteligentes para saber la cantidad de energía que se requiere en un cierto instante de tiempo. Esta capacidad junto con una predicción fiable de la producción de energías renovables, permite determinar en menor tiempo el punto de equilibrio entre la oferta y demanda de energía [136].

4.5.2. Arquitecturas y sistemas existentes

El continuo aumento en la integración y flexibilidad de los sistema embebidos, y la reducción de los costes tanto en el diseño como en la producción, representan ventajas significativas para la utilización de estos dispositivos en la *Smart Grid*. Sin embargo, a nivel comercial sigue siendo difícil encontrar dispositivos que cubra todos los aspectos discutidos en la sección 3.2 en un solo dispositivo. A continuación se describen varias arquitecturas de dispositivos que se pueden integrar en la *Smart Grid*.

En [151] proponen una arquitectura de un *Gateway* inteligente para medir el consumo de energía en el hogar, que puede obtener los datos de consumo de energía en tiempo real de cada dispositivo. Por ejemplo, puede medir el consumo de energía de aire acondicionado, calentadores de agua, ventiladores y otros equipos eléctricos. Los datos recogidos pueden ser utilizados para detectar el punto de falla por la sobrecarga de la red y realizar la desconexión en caso de emergencia. La arquitectura se compone principalmente de cinco partes, un módulo de procesamiento *(CPU)*, un módulo de entradas analógicas, un módulo de entradas/salidas digitales, un módulo de comunicación y un módulo de suministro de energía. La Figura 4.2 muestra un diagrama de bloques de la arquitectura del *Gateway* que han planteado los autores de este trabajo.

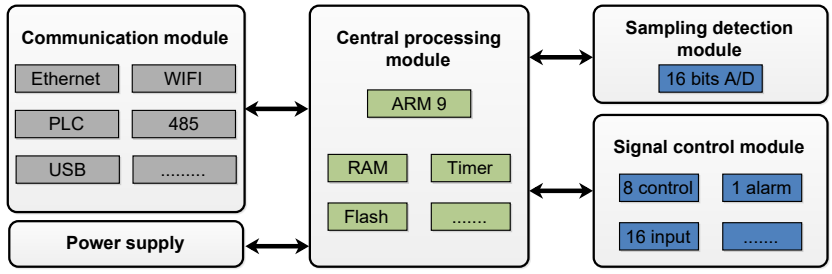


Figura 4.2: Arquitectura de un *Gateway* *Gateway* inteligente para medir el consumo de energía [151]

En [152] plantean una infraestructura de comunicaciones entre dos niveles, el de campo y el de estación, con especial atención en las aplicaciones para la medición, el control, la supervisión y la adquisición de datos. Para la integración

de los dos niveles proponen una combinación de dos soluciones, un *Gateway* y un *tunneling* que permite una conexión semitransparente de extremo a extremo entre los servidores de aplicaciones y los nodos de campo. Plantean como medio de comunicación el cableado de la red eléctrica *(PowerLine Communication)* y el uso de protocolos industriales definidos en el estándar *IEC 60870* (buses de campo). La Figura 4.3 muestra un esquema de la arquitectura de comunicaciones y los dispositivos que proponen los autores de este trabajo.

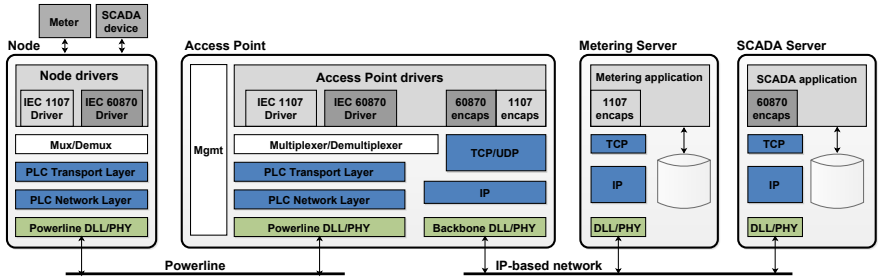


Figura 4.3: Esquema de la pila de protocolos implementados en el *Gateway* de comunicaciones. [152]

En [153] los autores abordan los problemas de seguridad, interoperabilidad e integración entre los dispositivos fasores heredados y los de última generación mediante el diseño de un *Gateway* que incorpora mecanismo de seguridad. El *Gateway* también proporciona funcionalidades de conversión de protocolos desde *IEEE C37.118.2* a *IEC 61850-90-5* y viceversa. La seguridad esta basadas en *Group Domain of Interpretation (GDOI)*, que es un mecanismo de seguridad recomendado por el *IEC 61850-90-5* para los sistemas basados en sincrofasores, debido a que proporciona autenticación, refresco periódico de claves y confidencialidad. La arquitectura del *Gateway* esta basado en un procesador *ARM* de bajo consumo y de bajo costo. La Figura 4.4 muestra un esquema de la arquitectura del *software* para establecer comunicaciones seguras basadas en *GDOI.*

En [154] plantean una arquitectura de un *Gateway* multi-puerto *Ethernet* que permite concentrar en un dispositivo varias *Merging Units (MU)*. La arquitectura esta basa en *FPGA* por lo que tiene la capacidad de recibir y procesar las tramas *GOOSE* y *SV* provenientes de las *MU* en tiempo real. En la *FPGA* también se implementa un *soft-procesor NIOS II*, este procesador tiene como objetivo

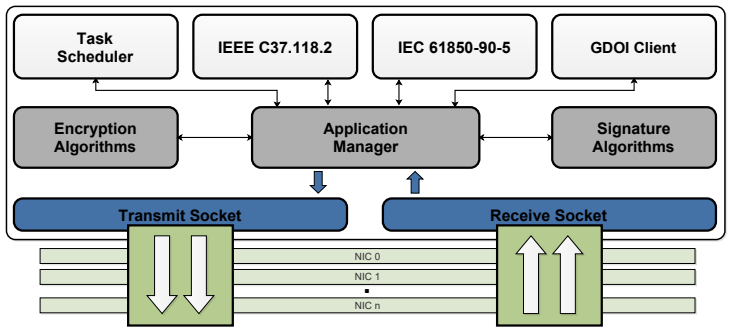


Figura 4.4: Arquitectura a nivel de software del *Gateway* de seguridad [153]

ejecutar un *software* que permite gestionar la configuración de la *FPGA* para realizar el procesamiento de las tramas. La Figura 4.5 muestra un diagrama de bloques de la arquitectura del *Gateway* multi-puerto *Ethernet* que proponen los autores de este trabajo.

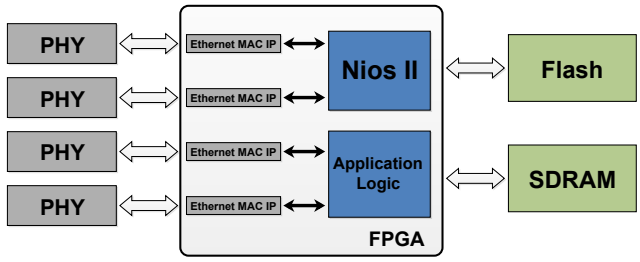


Figura 4.5: *Gateway* multi-puerto *Ethernet* [154]

En [155] describen la arquitectura y la implementación de una *Merging Unit (MU)* basada en *FPGA*. Esta arquitectura aprovecha al máximo las características de velocidad de procesamiento, la multitarea y la modularidad de la *FPGA* para conseguir que las *MUs* procesen y transmitan las tramas de datos según la norma *IEC61850-9-2*. La arquitectura incorpora dos interfaces *Ethernet* para realizar comunicaciones a través de cable y fibra óptica. La Figura 4.6 muestra un diagrama de bloques de la arquitectura de la *MU* propuesta por los autores de este trabajo.

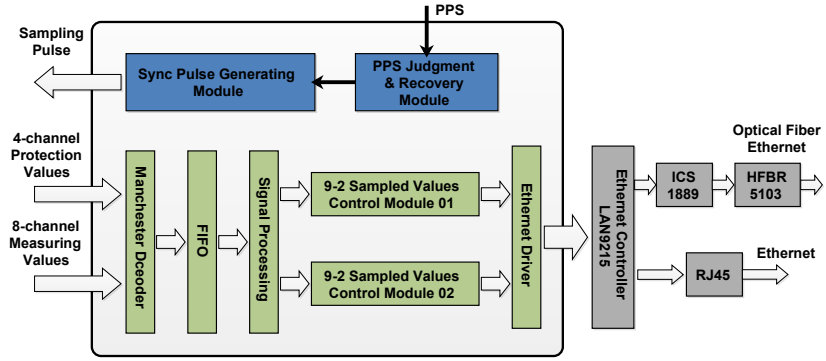


Figura 4.6: Diagrama de bloques de la arquitectura de un *Merging Unit* [155]

En [**156**], han desarrollado un dispositivo de protección *(relay)* basado en *FPGA*. Para conseguir un procesamiento en tiempo real, en la *FPGA* se han implementado la lógica que se encarga de procesar las tramas *SV* y *GOOSE* conforme a la norma *IEC61850*. Además, el dispositivo dispone de conversores analógico-digitales para realizar la medición de señales de tensión y corriente. También, se implementan las funciones para realizar la comunicación y las funciones de protección. En la Figura 4.7 se muestra el esquema general del relé de protección basado en *FPGA* propuesto por los autores. El esquema consta de tres partes: un módulo para la gestión del dispositivo, un módulo donde se implementan todas la funcionalidades del dispositivo y un módulo de las interfaces externas.

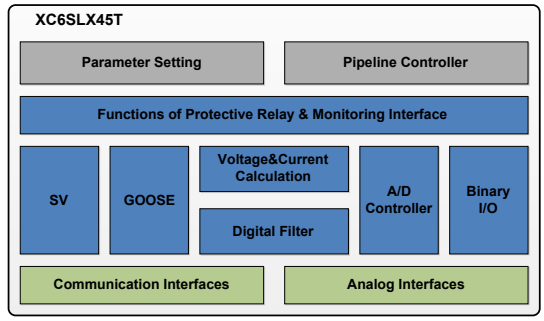


Figura 4.7: Esquema general del relé de protección basado en *FPGA* [156]

Como se mencionó en la sección 3.2, las redes de automatización de subestaciones requieren una alta disponibilidad y una sincronización precisa de tiempo. En este sentido, en [157] los autores describen la implementación basada en *FPGA* de los protocolos *HSR* y *PRP* definidos en el *IEC 62439-3* y del protocolo *IEEE 1588 PTP*. Los autores demuestran que toda las funciones de los protocolos *HSR, PRP* y *PTP* pueden ser implementadas en la *FPGA*. Esta implementación es completamente en *hardware* y permite liberar de carga al procesador. El diseño planteado es modular y permite construir nodos con un gran número de puertos, adicionalmente el diseño es independiente de la tecnología y del sistema operativo. En la Figura 4.8 se muestra un diagrama de bloques de la arquitectura del dispositivo propuesto por los autores.

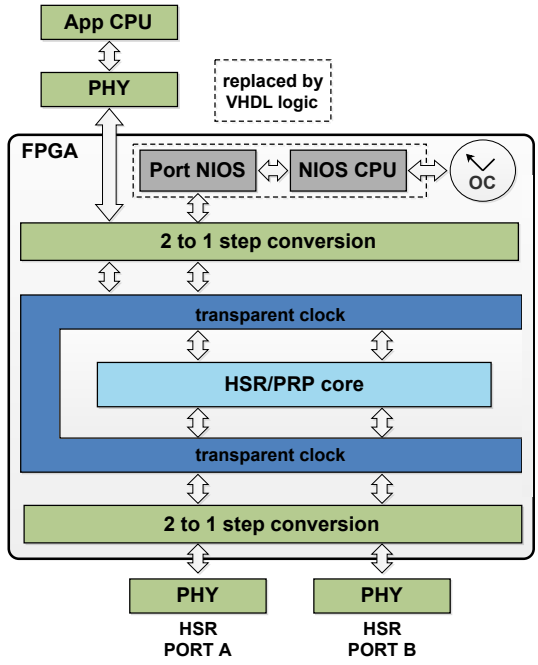


Figura 4.8: Diagrama de bloques de la arquitectura con funcionalidad *HSR, PRP* y *PTP* [157]

En [158], para conseguir un procesamiento en tiempo real han desarrollado un dispositivo de protección *(relay)* basado en *FPGA*. Como se muestra en la Figura

4.9, los algoritmos implementados en la *FPGA* se agrupan en tres módulos. El primer módulo es la interfaz de comunicaciones en la que implementa la lógica que se encarga de procesar las tramas *SV* y *GOOSE* conforme a la norma *IEC61850*. Los paquetes *SV* recibidos son procesados por el segundo módulo, que se encarga de calcular en base a las medidas de corriente y tensión la impedancia de falla y determina la lógica de disparo. Las señales de disparo también se generan en este módulo y son empaquetadas en formato *GOOSE*. El tercer módulo implementa la máquina de estados que controla todo el diseño.

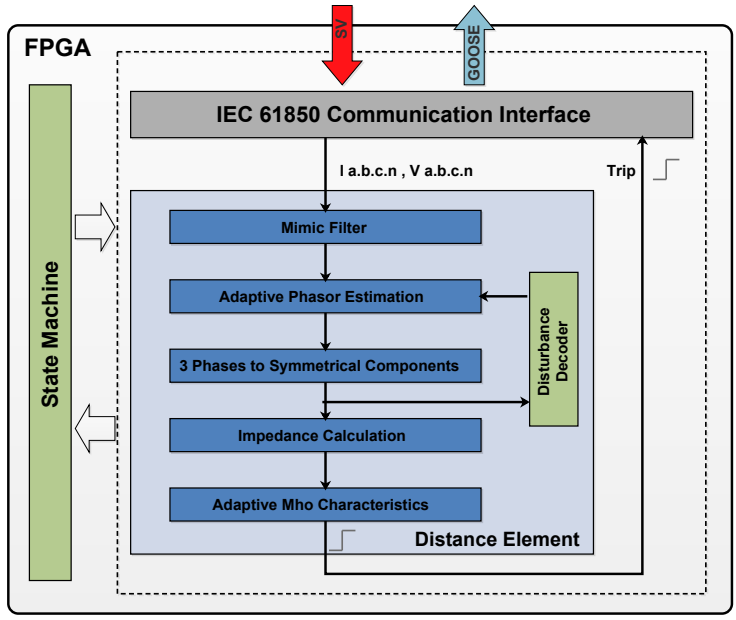


Figura 4.9: Diagrama de bloques de un relé de protección basado en *FPGA* [158]

El trabajo realizado por [159], se describe la implementación de un dispositivo que permite la sincronización de tiempo *(Time Gateway)* entre los dispositivos antiguos con los modernos basados en el protocolo *IEEE 1588 PTP*. Los dispositivos antiguos suelen obtener la información de tiempo mediante el conocido protocolo *SNTP*, que utiliza representaciones de tiempo y estrategias de sincronización muy diferentes con respecto al protocolo *PTP*. Por lo tanto, el *Gateway*

permite interconectar dispositivos *SNTP* con un dominio de sincronización *PTP*. Los autores han demostrando su aplicabilidad por medio de un prototipo basado en *FPGA*. El diagrama de bloques del *Gateway* se muestra en la Figura 4.10, tiene dos interfaces de red: una para el dominio *PTP* y otra para el dominio *SNTP*. La interfaz *Ethernet (PHY+MAC)* para el dominio *SNTP* es externa y se conecta mediante un puerto de expansión *(Bridge)*. La lógica de control de la arquitectura del *Gateway* está implementada alrededor de un procesador *NIOS II*, que es un *soft-procesor* de 32 *bits* que dispone de periféricos especiales para procesar los datos procedentes de las interfaces de red.

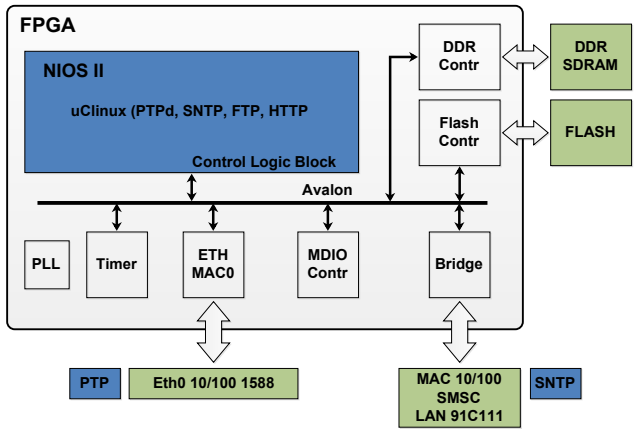


Figura 4.10: Diagrama de bloques del *Time Gateway* propuesto en [159]

En [160] plantean el uso de modernos sistemas embebidos *(SoC)* que integran procesadores implementados en silicio y una *FPGA*. En el documento se plantea la posibilidad de implementar varios tipos de dispositivos, como por ejemplo, un controlador de subestación, un relé digital, un *switch HSR/PRP*, etc. En la Figura 4.11 se observa la arquitectura de un controlador de subestación, el hardware utilizado es un *SoC* de *Intel* que incluye un procesador *ARM* implementado en silicio, mediante el uso de IP cores se puede añadir comunicación de alta disponibilidad, mecanismos de sincronización *IEEE 1588-2008 PTP* y protocolos de comunicaciones industriales *(fieldbus)*. El *software* que permite ejecutar esta arquitectura es muy variado, desde aplicaciones *standalone* hasta un sistema operativo en tiempo real.

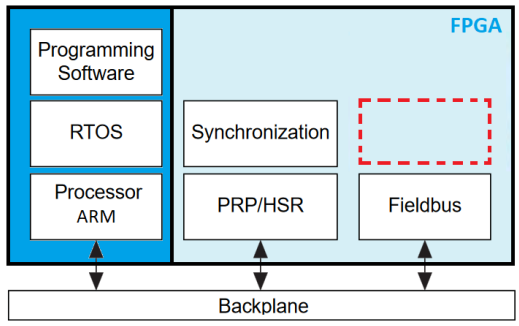


Figura 4.11: Arquitectura de un controlador de subestación basado en una plataforma SoC [160]

En [161], describen el diseño de un dispositivo terminal inteligente que cumple con las especificaciones del estándar *IEC61850*, la implementación ha sido realizada en una plataforma que integra un microprocesador *(ARM)* y una *FPGA*. El terminal tiene la capacidad de generar tramas *GOOSE* y *MMS*, puede realizar el control en modo fase o multifase y como método de sincronización utilizar *IRIG-B*. La estructura de la arquitectura del dispositivo terminal inteligente se muestra en la Figura 4.12, se puede identificar un módulo de procesamiento de datos basado en un procesador *ARM Cortex-M3*, un módulo de comunicación *GOOSE*, un módulo de entrada de señales de estado, un módulo de procesamiento de señales analógicas, un módulo de comunicación *MMS*, un módulo de sincronización, un módulo de alimentación y un módulo para comunicaciones seriales.

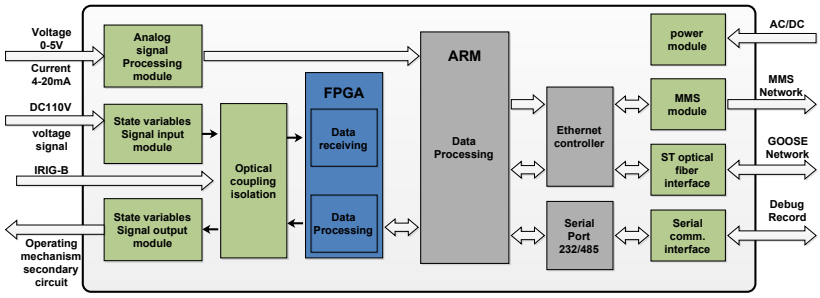


Figura 4.12: Arquitectura de un dispositivo terminal inteligente basado en una plataforma *SoC* [161]

4.5.3. Cumplimiento de los requisitos de la *Smart Grid* por parte de los sistemas identificados en el Estado del Arte

En esta sección se analizan las propuestas de arquitecturas de dispositivos para ser utilizadas en la *Smart Grid* considerando los requisitos analizados en la sección 3.2. Ademas de estos requerimientos también es necesario identificar la plataforma en la que se implementan las arquitecturas.

Las propuestas [151–153] presentan una arquitectura de un dispositivo basado en un microprocesador, que proporciona flexibilidad a nivel de *software* permitiendo añadir nuevas funcionalidades al dispositivo. El tipo de microprocesador y su capacidad de procesamiento determinará la cantidad y complejidad de las aplicaciones que se pueden añadir al dispositivo, tales como mecanismos de acceso a datos (*web, FTP*, entre otros.) y seguridad para tramas de capa tres *(VPN, IPSec)*. Por otro lado, las arquitecturas incorporan interfaces serie y Ethernet que permiten la interoperabilidad con otros dispositivos. Como puntos negativos, los diseños [151, 152] no incorporan mecanismos para garantizar la sincronización, alta disponibilidad y seguridad de las comunicaciones de Capa 2. La propuesta [153] centra su estudio únicamente en un mecanismo de seguridad para las comunicaciones del protocolo *IEC TR 61850-90-5*.

Con un enfoque diferente [154–159] proponen el uso de una *FPGA* como plataforma para implementar un dispositivo inteligente. El uso de *FPGAs* permite garantizar la ejecución de los procesos en tiempos exactos (determinismo) y con mayor rapidez. El punto negativo de este enfoque es que los diseños son específicos para una aplicación. Sin un microprocesador, es difícil cambiar el *software* y añadir nuevas funciones al dispositivo. De manera general estos trabajos se centran en el desarrollo de una arquitectura que da solución solamente a uno de los requerimientos identificados anteriormente. Por ejemplo [158, 159] realizan la implementación del protocolo de sincronización *IEEE 1588 PTP*. En [157] a más de la sincronización también implementan una solución de comunicación *HSR* y *PRP*. Por otra parte, [154–156] se centran unicamente en el procesamiento de las tramas *GOOSE* y *SV*.

Finalmente, en [160, 161] proponen el uso de una plataforma *SoC* que incorpora procesadores implementados en silicio y una *FPGA*. Esta configuración aprovecha la flexibilidad a nivel de *software* que permite una arquitectura con un microprocesador. A nivel de *hardware (FPGA)*, la elección de la arquitectura in-

terna permitirá ofrecer beneficios más significativos al dispositivo, por ejemplo, comunicaciones de alta disponibilidad *(IP Core HSR / PRP)*, sincronización *(IP Core 1588)*, entre otros. Como punto negativo, estas arquitecturas no incorporan todas las características requeridas por un dispositivo para *Smart Grid*. Por ejemplo, [161] incorpora un mecanismos de sincronización básico y no implementa comunicaciones de alta disponibilidad.

En la Tabla 4.2, se resumen las características identificadas en los trabajos analizados en la sección anterior. Es importante destacar que las arquitecturas propuestas en los trabajos mostrados en la Tabla 4.2, ninguna de ellas satisface completamente todos los requerimientos de operación que exige la *Smart Grid* y, en particular, no cuentan con un mecanismos para realizar la configuración remota. En cuanto ciber-seguridad para las tramas *Ethernet* de Capa 2 *(GOOSE y SV),* unicamente se identificó la propuesta de [153], pero al ser una implementación basada en microprocesador los resultados presentados concluyen que los tiempos obtenidos no satisfacen los requerimientos del estándar *IEC 61850*.

Atendiendo a este análisis, se plantea el reto de definir arquitecturas suficientemente generales como para dar solución a cualquiera de los escenarios de aplicación planteados en los trabajos presentados en la sección 4.5.2. Por otra parte las arquitecturas planteadas deben incorporar mecanismos que permitan satisfacer los requisitos de operación que plantea la *Smart Grid*.

Tabla 4.2: Comparación con trabajos relacionados que se centran en el diseño de dispositivos inteligentes para la *Smart Grid*.

Características	**[151]**	**[152]**	**[153]**	**[154]**	**[155]**	**[156]**	**[157]**	**[158]**	**[159]**	**[160]**	**[161]**
Plataforma											
Microprocessor	√	√	√	-	-	-	-	-	-	-	-
FPGA	-	-	-	√	√	√	√	√	√	-	-
Microprocessor + *FPGA*	-	-	-	-	-	-	-	-	-	√	√
Sincronización	*X*	*X*	*X*	*X*	*X*	*X*	√	√	√	√	*X*
Alta disponibilidad	*X*	*X*	*X*	*X*	*X*	*X*	√	*X*	*X*	√	*X*
Interoperabilidad											
Interfaces seriales	√	*X*	√	√	*X*	√	*X*	√	*X*	√	√
Interfaces Ethernet	√	*X*	√	√	√	√	√	√	√	√	√
Buses de Campo	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*
Reconfigurable	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*
Ciber-seguridad	*X*	*X*	*Layer 2*	*X*	*X*	*X*	*X*	*X*	*X*	*Layer 3*	*X*

4.6. Resumen

En este capítulo se ha presentado una visión general sobre los *CPS*. Se han identificado varias definiciones en la literatura, de las cuales se puede concluir que los *CPS* son sistemas electrónicos complejos y multidisciplinarios, caracterizados por la integración de componentes físicos (sensores, actuadores), de procesamiento (control, análisis de datos) y de comunicaciones, que pueden interactuar con otros sistemas y con los seres humanos.

Por otra parte se han identificado algunas características que deben tener los *CPS* entre las que podemos mencionar: la capacidad para detectar y reconocer objetos y variables físicas, comunicaciones estandarizadas para garantizar la interoperatividad entre componentes y sistemas heterogéneos, ciber-seguridad para evitar la captura de información o inyección de información no deseada en los sistemas, la fiabilidad y estabilidad para garantizar que los *CPS* puedan operar siempre de la misma forma (fiabilidad) ante perturbaciones externas o internas (estabilidad), procesamiento en tiempo real para ejecutar algoritmos de control distribuido y análisis de datos, entre otros.

También se han identificado varias iniciativas de investigación en diferentes campos de conocimiento desde la perspectiva de los *CPS*. Por ejemplo, en el área transporte, los automóviles modernos pueden reducir la velocidad automáticamente para evitar colisiones y con ayuda del *GPS* pueden enviar información relevante del tráfico y de las condiciones de la calzada a otros vehículos. En el área de la salud se están desarrollando dispositivos médicos para el monitoreo los signos vitales en tiempo real, que puedan adaptarse a los cambios y enviar la información a centros de control remotos. En el campo de las infraestructuras, los edificios inteligentes dispondrán de sistemas automatizados de control de calefacción, ventilación, climatización, iluminación, entre otros. Estos sistemas han sido tradicionalmente diseñados para funcionar de forma independiente, sin embargo, las investigaciones en *CPS* proponen mecanismos para la integración de los mismos. Esta integración permitirá mejorar la gestión y la seguridad de los edificios, optimizar el uso de la energía, reducir los costes de operación y de mantenimiento. La investigación de *CPS* no solamente ofrece oportunidades para mejorar la eficiencia y el rendimiento en edificios comerciales y residenciales, también es aplicable en otras infraestructuras de gran complejidad debido a su distribución geográfica, como por ejemplo en la red eléctrica. En este ámbito, es fundamental desarrollar una tecnología que pueda supervisar, evaluar y controlar con precisión la integridad de la red de generación, transmisión y distribución de

la energía. En este sentido, es necesario proponer arquitecturas de dispositivos electrónicos con capacidad para procesar datos en tiempo real con bajos tiempos de respuesta y que, además, integren sistemas de comunicaciones de alta disponibilidad, con un gran ancho de banda y baja latencia.

En general, las arquitecturas analizadas anteriormente se caracterizan por el uso de un único dispositivo ya sea un microprocesador, una *FPGA* o un *SoC*. En este dispositivo se implementan todas las funciones responsables tanto de la ejecución de las aplicaciones de control como de la gestión de los protocolos de comunicación. También, es importante destacar que de las arquitecturas descritas ninguna de ellas satisface completamente todos los requerimientos de operación que exige la *Smart Grid*. En este sentido, la arquitectura del *CPS Gateway* que se plantea en esta tesis pretende ser lo suficientemente flexible como para dar solución a todos los requisitos de operación identificados en la sección 3.2.

Capítulo 5

Propuesta de una arquitectura de un *Cyber Physical System Gateway*

5.1. Introducción

En la literatura, se identificaron varias arquitecturas de dispositivos para ser utilizados en la *Smart Grid*, pero ninguna de ellas satisface todos los requerimientos de operación que exige el sector, en particular la reconfiguración remota y la ciber-seguridad para las tramas *GOOSE* y *SV*. El objetivo principal de esta tesis es proponer una arquitectura de un *CPS Gateway* considerando los estándares de comunicación utilizados en la *Smart Grid* descritos el el capítulo 2, los requerimientos de operación identificados en el capítulo 3, así como, la arquitectura general y las características de un *CPS* analizadas en el capítulo 4.

En la sección 5.2 se presenta una visión general de la arquitectura del *CPS Gateway* propuesta y los requisitos de diseño que debe cumplir la nueva arquitectura *SoC*. En la sección 5.2.1 se propone una arquitectura que garantice el funcionamiento del sistema en tiempo real. En la Secciones 5.2.2, 5.2.3 y 5.2.4 se proponen

arquitecturas para garantizar la alta disponibilidad en las comunicaciones, la interoperabilidad y la reconfiguración de los módulos de la arquitectura del *CPS Gateway*, respectivamente. En la sección 5.2.5 se propone la arquitectura de un *CPS Gateway* incorporando mecanismos de ciber-seguridad en lo que respecta a las comunicaciones *Ethernet* de Capa 2. Finalmente, una vez analizados cada uno de los requerimientos y planteadas las arquitecturas, en la sección 5.3 se propone una arquitectura general que integra en un mismo dispositivo todos los componentes definidos en la sección 5.2.

5.2. Definición de la arquitectura base

Un *Gateway* desempeña un papel fundamental en una red de comunicaciones, se encarga de recoger los datos de los dispositivos, luego filtrarlos y reenviarlos a los dispositivos de niveles superior. El diseño de un *Gateway* de comunicaciones es un reto que implica no sólo el desarrollo de la arquitectura del *hardware*, sino también del *software*. Estas actividades toman una cantidad significativa de tiempo y requieren la participación de un gran equipo de diseño, por consiguiente los ciclos de diseño y comercialización del producto se incrementan. Diseñar un *Gateway* para cada aplicación desde cero no es una solución viable, en este sentido, en esta tesis se ha planteado el diseño de una arquitectura de un *Gateway* de comunicaciones lo suficientemente flexible como para dar solución a todos los requerimientos de comunicaciones que exige la *Smart Grid*. Para que la arquitectura sea personalizable a cada situación es necesario utilizar dispositivos *SoC* reconfigurables que integren en el mismo silicio la unidad de procesamiento y el área reconfigurable (*FPGA*).

Para el planteamiento de una arquitectura en primer lugar hay que determinar el entorno donde se utilizara el dispositivo y los requerimientos de operación. En este sentido, teniendo en cuenta los requerimientos recogidos en la sección 3.2, es necesario desarrollar y proponer una arquitectura de un dispositivo de comunicaciones *(Gateway)*, que permita:

- Realizar procesamiento de datos y operación en tiempo real (latencia, determinismo, sincronización).
- Establecer comunicaciones de alta disponibilidad.
- Establecer comunicación con todos los niveles de la subestación (proceso,

bahía, estación) y fuera de ella (mercados, operaciones, proveedor de servicios).

- Reconfigurarse.
- Comunicaciones con mecanismos de ciber-seguridad.

A continuación en cada una de las secciones se plantean arquitecturas que permiten dar solución a los requerimientos de operación descritos en la sección 3.2.

5.2.1. Operación en tiempo real

El concepto de "tiempo real" implica una respuesta a eventos o señales en un tiempo predecible después de su ocurrencia. Por ejemplo, en el sector eléctrico los lazos de control digital rápidos pueden exigir tiempos de reacción menores que $3\,ms$ (véase Tabla 3.2 mensajes tipo 1A y 4). En este sentido, para garantizar el funcionamiento de un sistema en tiempo real es necesario contar con dispositivos que puedan leer, procesar y transmitir los datos rápidamente. Para ello es necesario en primer lugar contar con un sistema de procesamiento *(Processing System, PS)* que permita ejecutar las diferentes aplicaciones, ya sea de manera directa *(standalone)* o mediante un sistema operativo. En la Figura 5.1 se muestra una arquitectura de un *PS* con los elementos básicos para implementar una aplicación de control, como por ejemplo puertos con entradas y salidas digitales, interfaces de comunicación seriales y *Ethernet*.

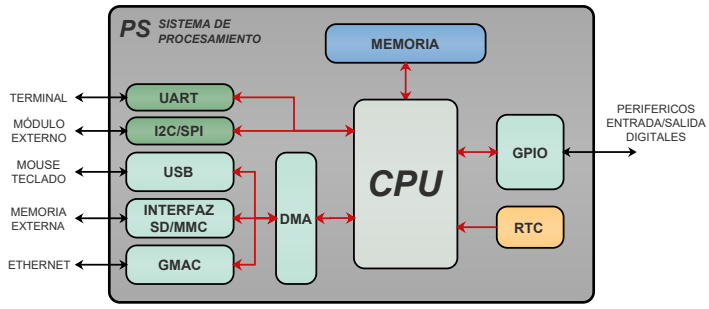


Figura 5.1: Arquitectura de un sistema de procesamiento

Como se mencionó anteriormente, muchos de los datos de medición de la red eléctrica deben tener una precisión absoluta de aproximadamente $1\,ms$. Una precisión de tiempo de $1\,ms$ es relativamente fácil de alcanzar. Sin embargo, las tareas de control actuales exigen una precisión menores a $1\,\mu s$. En este sentido como se analizó en la sección 3.2.2, el protocolo *IEEE 1588 PTP* es la mejor opción a considerar, ya que permite obtener una sincronización de tiempo en el rango de los nanosegundos.

De las diferentes formas de implementar el protocolo *IEEE 1588 PTP*, la implementación en *hardware* es la mejor solución, puesto que permite alcanzar el mayor rendimiento sin sobrecargar el sistema de procesamiento de la arquitectura. En este sentido, es necesario utilizar una *FPGA* para implementar las funciones *PTP* utilizando *IP cores* de uso comercial. En la Figura 5.2, se observa la arquitectura de un *CPS Gateway* añadiendo el mecanismo de sincronización *IEEE 1588 PTP* a través de un *IP core*.

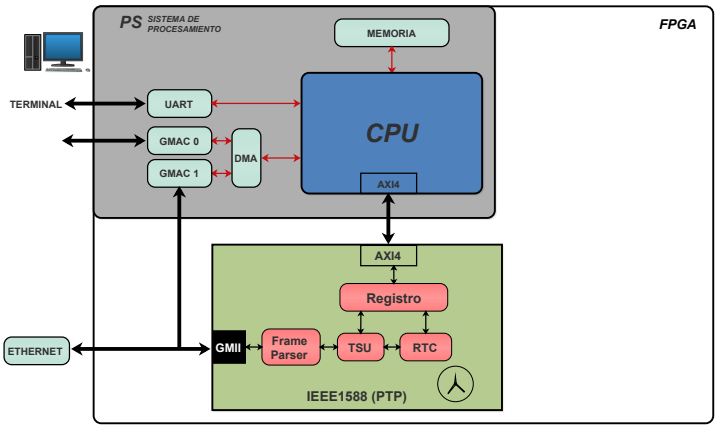


Figura 5.2: Arquitectura *CPS Gateway:* Sincronización *IEEE 1588 PTP* implementación en *hardware*

Una implementación *PTP* en *hardware* generalmente incorpora una unidad central de procesamiento *(CPU)* que se encarga de ejecutar la pila *PTP* y al menos, dos módulos de *hardware* específicos: un reloj en tiempo real *(RTC)* y una unidad que se encarga de realizar el *time stamping (TSU)*. El *RTC* es normalmente un contador de 64 *bits* en el cual se incorporan algunos registros que permiten configurarlo. Básicamente, estos registros controlan la frecuencia con la que se

actualiza el temporizador y la cantidad de nanosegundos que hay que añadirle en cada momento. La *TSU* es la responsable de almacenar información importante sobre la trama que se esta procesando, como por ejemplo, el *ID* de la secuencia, el tipo de mensaje o la información de *time stamping*.

Como se observa en la Figura 5.2, en una implementación *PTP* en *hardware* es necesario otros módulos adicionales, como es el caso del *Frame Parser* y la interfaz de registro. El *Frame Parser* se integra dentro del módulo *TSU* y se comunica con la interfaz *PHY* del dispositivo, analiza todas las tramas *Ethernet* y detecta las que son *IEEE 1588*. Dependiendo de la configuración del *hardware*, puede detectar tramas *IEEE 1588* etiquetadas a nivel de Capa 2 o 3.

La interfaz de registro permite comunicar la *CPU* y los módulos de *hardware PTP*. A través de estos registros, los controladores *software IEEE 1588* pueden leer la información sobre la trama y el *timestamping*, así como controlar la configuración del *RTC*. Estos controladores proporcionan una capa de abstracción y definen el conjunto de funciones que pueden emplear las aplicaciones de usuario para gestionar los módulos *TSU* y *RTC*.

Finalmente, otros elementos adicionales pueden ser necesarios para ejecutar la pila *PTP*, como temporizadores, controladores de memoria *(DDR)*, módulos de comunicación serial *(UART)*, etc. La configuración del sistema depende de las necesidades de la aplicación y del *software* que se ejecute en la *CPU*.

En la Tabla 5.1 se presenta la cantidad de recursos necesarios para implementar un *IP core IEEE 1588 PTP*. El *IP core* analizado corresponde al desarrollado por **SoC***e* [162]. En este sentido se tomó como base los recursos disponibles en una *FPGA Zynq XC7Z020* de *Xilinx*. Esta *FPGA* tiene 106400 *Slice Registers*, 53200 *Slice LUTs*, 140 *RAMB36*, 280 *RAMB18* y 220 bloques *DSP48E1*. Como se puede observar, para la implementación del *IP core* en el peor de los casos se requiere el 4,29 % de los recursos de la *FPGA* .

Tabla 5.1: Recursos de la *FPGA* utilizados en la implementación del módulo *IEEE 1588 PTP*

Recurso	Módulo *PTP*	Disponible	Utilización (%)
Slice LUTs	1408	53200	2,65
Slice Registers	2340	106400	2,21
Block RAM	6	140	4.29

5.2.2. Alta disponibilidad

En los *SAS*, además de precisión en la sincronización de tiempo en niveles inferiores a $1\,\mu s$, existen otros problemas críticos, como la necesidad de una red de alta disponibilidad. La tecnología elegida para esta red debe evitar la perdida de información en caso de falla del enlace de comunicaciones (tiempo de recuperación). Específicamente, los tiempos de recuperación son muy exigentes en el sector, llegando a fijarse cero segundos para redes con tráfico *GOOSE* y *SV*. Estas tramas son de nivel 2, sin mecanismos de notificación que confirme la recepción por parte del destinatario. Esta necesidad ha forzado el desarrollo de estándares para el sector que aseguran cero tramas perdidas en caso de un único fallo en la red. A este respecto, el estándar *IEC 61850-90-4* plantea el uso de los protocolos *HSR* y *PRP* como mecanismo de redundancia.

Para evitar la perdida de rendimiento en las aplicaciones que se estén ejecutando en el microprocesador se plantea la implementación de los protocolos *HSR* y *PRP* a nivel de *hardware*. Con este enfoque, se garantiza que la capacidad informática disponible para la ejecución de las aplicaciones de usuario sea independiente de los algoritmos utilizados para gestionar la redundancia de la red. Como se ha mencionado en la sección 2.4.3, para que un dispositivo pueda operar con los protocolos *HSR* o *PRP* es necesario que disponga al menos de dos interfaces *Ethernet*, y de capacidad de implementación de los algoritmos que procesen estos protocolos. En este sentido, debido a la flexibilidad a nivel *hardware* que debe tener la arquitectura del *CPS Gateway*, es necesario plantear la implementación de los protocolos *HSR* y *PRP* utilizando *IP cores*.

En la actualidad existen varias soluciones *HSR/PRP* basadas en *IP cores*, como por ejemplo las proporcionadas por *Flexibilis* [163], **SoC***e* [164], *NetTimeLogic* [165], *InES* [166], entre otros. En la Figura 5.3 se muestra un diagrama de bloques de una arquitectura del *CPS Gateway* en la que se ha agregado un *IP core* con una configuración genérica para dar soporte a comunicaciones *HSR/PRP*.

La estructura del *IP core* permite implementar una topología *RedBox*. Por lo tanto, la funcionalidad de los tres puertos *Ethernet* que se observa en esta configuración es:

- Con los puertos *Ethernet* (Puerto A y Puerto B) se implementan la funcionalidad *DAN* para el caso de *PRP* y *DANH* para *HSR*. Estos puertos permiten conectar al equipo a una red con topología en anillo y gestionar

las tramas redundantes *HSR* o *PRP*.

- Con el Puerto C se establece un enlace *Ethernet* convencional entre el *IP core* y la *CPU*.

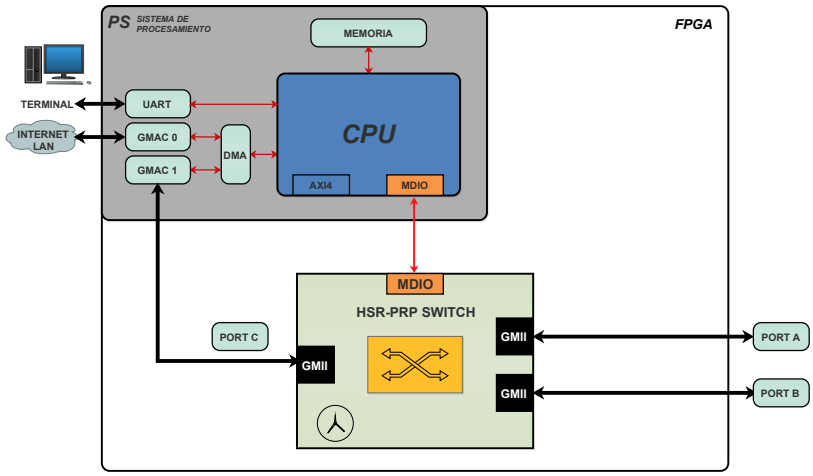


Figura 5.3: Arquitectura *CPS Gateway* para soportar alta disponibilidad: Implementación en *hardware* de *HSR* y *PRP*

En esta arquitectura, el Puerto C se conecta al puerto *Ethernet GMAC 1* dedicado del *PS*. Por lo tanto, la *CPU* es el nodo final para la información recogida por las interfaces *Ethernet* redundantes (Puerto A y Puerto B). El enfoque propuesto hace que la redundancia sea transparente para las aplicaciones que se ejecutan en el sistema de procesamiento. Por lo tanto, todos los protocolos de alto nivel como *TCP/IP* pueden ejecutarse sin ninguna modificación. A través de la interfaz *MDIO*, la *CPU* puede acceder al *IP core HSR-PRP* para su configuración y supervisión.

En la Tabla 5.2 se presenta la cantidad de recursos necesarios para implementar un *IP core HSR-PRP*. El *IP core* analizado corresponde al desarrollado por **SoC***e* [164]. Como se puede observar, para la implementación del *IP core* en el peor de los casos se requiere el 34,50 % de los recursos de la *FPGA*.

Tabla 5.2: Recursos de la *FPGA* utilizados en la implementación del módulo *HSR-PRP*

Recurso	Módulo *HSR-RPR*	Disponible	Utilización (%)
Slice LUTs	18356	53200	34,50
Slice Registers	17115	106400	16,08
Block RAM	32	140	22,85

5.2.3. Interoperabilidad

Como se ha mencionado en la sección 2.3.1, el estándar *IEC 61850* está siendo implementado en todos los niveles de una subestación, asegurando la interoperabilidad entre dispositivos y el cumplimiento de todos los requisitos de comunicaciones. Sin embargo, en la actualidad en una subestación todavía existen soluciones de automatización basadas en buses de campo. Por lo tanto, es necesario incorporar en la arquitectura del *CPS Gateway* mecanismos que permitan implementar los diferente buses de campo.

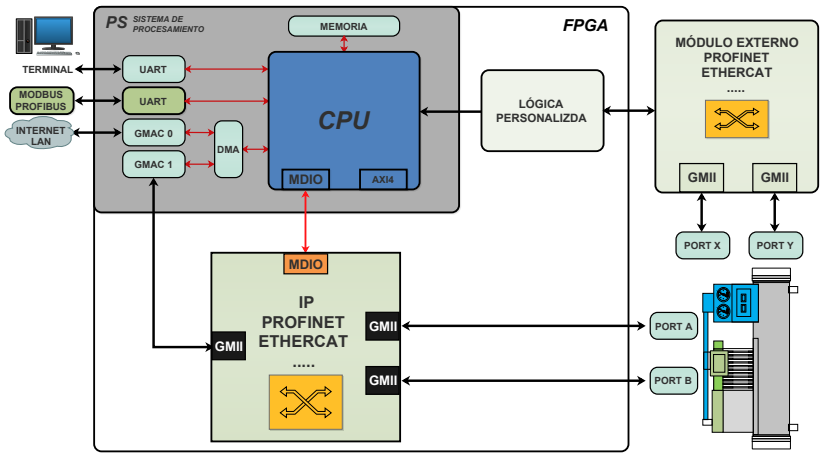


Figura 5.4: Arquitectura del *CPS Gateway* para dar soporte a la interoperatividad con requisitos de tiempo real

La Figura 5.4 muestra un diagrama de bloques de la arquitectura del *CPS Gateway* para que de soporte la interoperabilidad con otros dispositivos.

En primer lugar, el sistema de procesamiento *(PS)* permite ejecutar el *software* necesario para gestionar todos los componentes de la arquitectura y ejecutar las librerías específicas de un determinado protocolo de comunicación industrial. Para dar soporte a los buses de campo serie, como por ejemplo *Modbus* y *Profibus* la arquitectura debe incorporar interfaces seriales *(UART)*.

El rendimiento y los tiempos de ejecución que se consiguen con la implementación de los buses de campo serie a nivel de software es suficiente. Sin embargo, para el caso de los buses de campo Ethernet, que tienen que cumplir con ciertos requisitos específicos, como un comportamiento en tiempo real, ciclos de ejecución cortos y baja variación de tiempo *(jitter)*, soporte de varias topologías de red, incluyendo la redundancia mediante conexión en cadena o en anillo, es necesario incluir componentes de *hardware* específicos.

En este sentido, en la Figura 5.4 se pueden identificar dos enfoques de implementación diferentes. Mientras que en una implementación convencional se utiliza un módulo externo donde se implementan todas las funcionalidades del protocolo, en esta arquitectura se integran estas funcionalidades mediante la combinación de *IP cores* específicos.

Como puede observarse en la Figura 5.4, el intercambio de datos entre el *PS* y los *IP cores* se realiza mediante interfaces *GMII* a través de un *switch*. Para acceder a los registros de configuración de los *IP cores* utilizados en la arquitectura del *CPS Gateway* desde el *PS*, se utilizan las interfaces estandarizadas *AXI4* y *MDIO*.

En la Tabla 5.3 se presenta la cantidad de recursos necesarios para implementar un *IP core* para gestionar comunicaciones *Profinet*. El *IP core* analizado corresponde al desarrollado por **SoC***e* [167]. Como se puede observar, para la implementación del *IP core* en el peor de los casos se requiere el 25,71 % de los recursos de la *FPGA*.

Tabla 5.3: Recursos de la *FPGA* utilizados en la implementación del módulo *Profinet*

Recurso	Módulo *Profinet*	Disponible	Utilización (%)
Slice LUTs	3640	53200	6,84
Slice Registers	4196	106400	3,94
Block RAM	36	140	25,71

5.2.4. Reconfigurabilidad

Como se mencionó en la sección 3.2.5 una de las necesidades de los dispositivos modernos es la posibilidad de configuración remota y la actualización de *firmware* de los *IP cores* que contienen la *FPGA*. En este caso, una alternativa que se propone para la arquitectura del *CPS Gateway*, es utilizar un módulo *hardware* y un protocolo de comunicación que se transmite a través del mismo canal *Ethernet* de la infraestructura de red implementada en la *FPGA (HSR, IEEE 1588, etc.)*. La implementación del módulo de configuración en la *FPGA* no debe tener dependencia *software* o de *CPU* para asegurar la disponibilidad del servicio de configuración ante cualquier circunstancia. Debe ser compacta para no generar costos adicionales y debe proporcionar seguridad incorporando un módulo para la autenticación y cifrado de mensajes.

Arquitectura del *IP core COEsec*

COEsec tiene la ventaja de no requerir una *CPU* para su funcionamiento y usa pocos recursos de la *FPGA*, a su vez, garantiza la seguridad en el momento de acceder al dispositivo remoto por medio de encriptación y autenticación. El núcleo del *IP core* es un motor criptográfico que es capaz de encriptar, descifrar y autenticar las tramas *COE* mediante el uso del algoritmo *AES-GCM* implementado en el *hardware*.

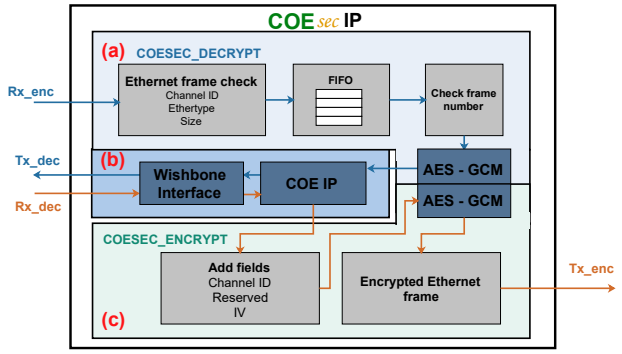


Figura 5.5: Diagrama de bloques de la arquitectura del *IP core COEsec*

La Figura 5.5 muestra el diagrama de bloques de la arquitectura del *IP core COEsec* propuesto. En la arquitectura se identifican tres bloques principales: a) Un módulo que implementa la lógica para de recepción, validación y descifrado de tramas *COEsec*. b) Un módulo para establecer la comunicación entre el *COEsec* y los *IP cores* a configurar, a través del bus *Wishbone* [168]. c) Un módulo para generar, cifrar y transmitir tramas *COEsec*. El principio de funcionamiento se describe a continuación:

Cuando una nueva trama *Ethernet* es recibida por *IP core COEsec*, primero se comprueba si la trama contiene un *Ethertype* válido, si pertenece a un canal válido *(Channel ID)*, y si tiene un tamaño de trama dentro de los límites establecidos en la configuración del *IP*. Si la validación es positiva, la trama se almacena en una memoria *FIFO* con capacidad para varias tramas. De lo contrario, la trama se descarta.

Las tramas almacenadas en la memoria *FIFO* se procesan una a una. Primero, se comprueba si el número de trama es válido para el canal con el que está asociado. Si el número de trama es incorrecto, se descarta la trama y se genera una trama de respuesta. Por el contrario, si el número de trama, el *Ethertype* y el *Channel ID* son válidos, la trama se envía a un módulo *AES-GCM*, que es responsable de descifrar y autenticar la trama. En este sentido, si la trama está correctamente autenticada, se envía al módulo *IP COE* sin los campos *Channel ID, Reserved* e *IV*. El *IP COE* es el encargado de interpretar el comando enviado en la trama (R o W) y transmitirlo al *IP core* a configurar. En cambio, si la trama no es autenticada, la trama se descarta. La misma operación se realiza con la siguiente trama almacenada en la memoria *FIFO*.

Una vez que una trama descifrada y autenticada es procesada por el *IP COE*, este envía una respuesta con la correspondiente información en función del comando utilizado (R o W). Cuando se recibe la respuesta, el *IP COEsec* genera una trama añadiendo los campos *Channel ID, Reserved* e *IV* y la envía al módulo *AES-GCM* para encriptarla y generar la etiqueta de autenticación. Finalmente, se construye la trama *Ethernet* con los datos cifrados y se transmite al dispositivo remoto.

Para describir el *hardware* de la arquitectura del *IP COEsec* mostrada en la Figura 5.5 se utilizó el lenguaje *VHDL*. Los recursos necesarios para la implementación del *IP COEsec* en una *FPGA Zynq (XC7Z020) Xilinx*, se presentan en la Tabla 5.4.

Tabla 5.4: Recursos de la *FPGA* utilizados para la implementación del módulo *COEsec*

Recurso	COE	*COEsec*	Disponible	Utilización *COE* (%)	Utilización *COEsec* (%)
Slice LUT	1318	4748	53200	2,48 %	8,92 %
Slice Registers	1215	4138	106400	1,14 %	3,89 %
Block RAMs	0.5	5,5	140	0,36 %	3,87 %

Como se puede observar, la implementación del módulo sin un mecanismo de seguridad en el peor de los casos requiere el 2,48 % de los recursos de la *FPGA*. Si se compara con la implementación del *IP* añadiendo un mecanismo de seguridad *(COEsec)*, se necesita un 6,44 % adicional de recursos. En resumen, la implementación propuesta del *IP COEsec* requiere en el peor de los casos un 8,92 % de los recursos totales disponibles en la *FPGA*.

En la Figura 5.6 se muestra la arquitectura del *CPS Gateway* incorporando el *IP core COEsec* y la funcionalidad de configuración remota de los *IP cores*. Este bloque incorpora una interfaz *Wishbone* maestra encargada de gobernar el acceso a los registros internos de configuración del resto de los *IPs* incluidos en la arquitectura.

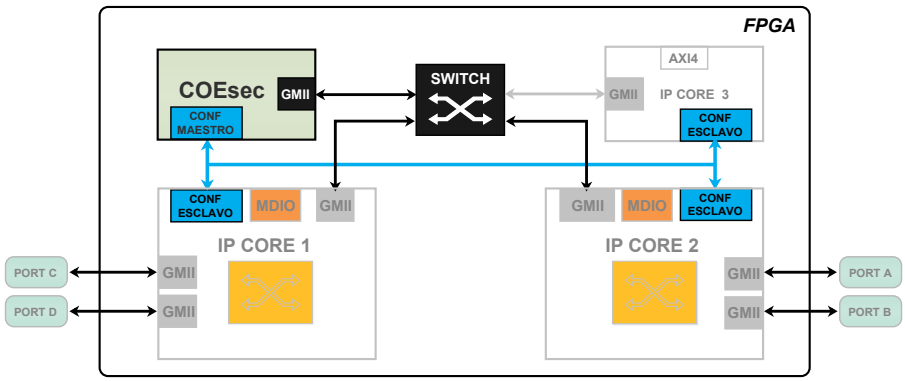


Figura 5.6: Arquitectura del *CPS Gateway:* Configuración remota *(FPGA)* incorporando la configuración remota de los *IP cores*

5.2.5. Ciber-seguridad

En esta sección se plantea una arquitectura de un *CPS Gateway* incorporando mecanismos de ciber-seguridad en lo referente a comunicaciones en Capa 2, que es donde se tramiten los mensajes de tiempo crítico *GOOSE*, *SV* y *PTP*.

En este sentido, se propone la implementación de un Módulo de Cifrado y Descifrado *(MCD)* que use criptografía simétrica para ofrecer seguridad a las comunicaciones de Capa 2. Tal y como se detalla en el capítulo 3, las latencias asumidas para el procesamiento de las tramas son excepcionalmente bajas debido a los estrictos requisitos de procesamiento en tiempo real. Esta restricción combinada con la necesidad de soportar un alto ancho de banda en la comunicación requerida en el caso de la agrupación de varios *streams* de *SVs* plantean un desafió técnico significativo. El procesamiento de las tramas se debe realizar a la velocidad de linea y con una arquitectura *hardware* dedicada que procese las tramas sin romper el flujo de transmisión y recepción de las mismas en el canal de comunicación. Para poder hacer frente a este desafió y poder plantear una arquitectura acorde a esta hipótesis, se requiere del uso de tecnología *FPGA*. Ésta ofrece la capacidad de cálculo requerida y la flexibilidad necesarias.

Como primer paso hay que definir y modelar la arquitectura *hardware* del *MCD*. En la Figura 5.7 se muestra el diagrama general de bloques de la arquitectura propuesta para ofrecer seguridad a las tramas *GOOSE* y *SV*.

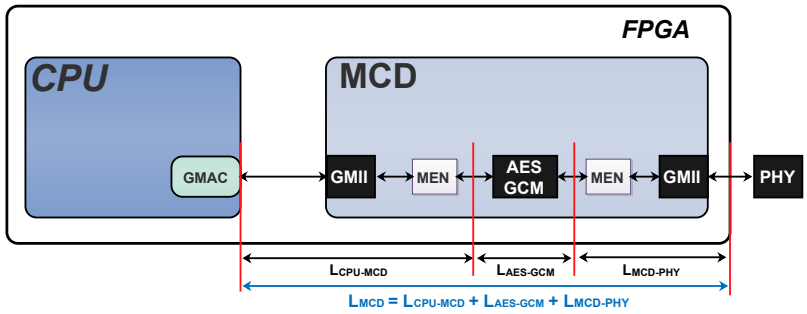


Figura 5.7: Latencia introducida por el MCD

Con el objetivo de verificar si la arquitectura propuesta cumple con los requeri-

mientos de operación que plantea el estándar *IEC 61850*, es necesario determinar el tiempo que necesita el *MCD* para procesar las tramas *GOOSE/SV*. Como se puede identificar en la Figura 5.7, para determinar la latencia introducida por el *MCD* no solo hay que considerar el retardo introducido por el proceso de cifrado (L_{AES_GCM}), también hay que considerar el tiempo necesario para la transmisión de la trama desde la *CPU* al *MCD* (L_{CPU_MCD}) y el tiempo necesario para la transmisión de la trama cifrada del *MCD* al *PHY* (L_{MCD_PHY}). Considerando estos retardos la latencia total introducida por el *MCD* se expresa con la Ecuación 5.1.

$$L_{MCD} = L_{CPU\_MCD} + L_{AES\_GCM} + L_{MCD\_PHY} \tag{5.1}$$

La latencia (L_{CPU_MCD}) introducida por el proceso de transmisión de la trama desde la *CPU* al *MCD* esta definida por la Ecuación 5.2.

$$L_{CPU\_MCD} = \frac{Tamaño\_trama\_bits}{Tasa\_de\_transmisión} \tag{5.2}$$

Según las especificaciones del estándar *IEC 61850-90-4* recomienda que el tamaño de las trama *SV* sean entre 160 y 180 *bytes*. Para el caso de las tramas *GOOSE*, las tramas deben ser entre 92 y 250 *bytes* [169].

Si se aplica la Ecuación 5.2 para una transmisión a $1\,Gbps$ y tramas *SV* de 180 *bytes*, se obtiene que el máximo retardo introducido por el proceso de transmisión para las trama *SV* es de $1{,}44\,\mu s$:

$$L_{CPU\_MCD_{SV\_max}} = \frac{Tamaño\_trama\_bits}{Tasa\_de\_transmisión} = \frac{180 * 8}{10^9} = 1,44\mu s \tag{5.3}$$

Para el caso de las tramas *GOOSE* el máximo retardo introducido es de $2{,}00\,\mu s$.

$$L_{CPU\_MCD_{GOOSE\_max}} = \frac{Tamaño\_trama\_bits}{Tasa\_de\_transmisión} = \frac{250 * 8}{10^9} = 2,00\,\mu s \tag{5.4}$$

Para determinar la latencia (L_{MCD_PHY}) introducida por el proceso de transmisión de la trama desde la el *MCD* al *PHY* también se utiliza la Ecuación 5.2, la diferencia radica en que el tamaño de la trama se ve incrementada porque se añaden los parámetros necesarios para realizar el descifrado de las tramas. Entre los que podemos mencionar el número de secuencia, el vector de inicialización *(IV)* , la etiqueta de autenticación *(Tag)* y si los datos a cifrar no son múltiplos de 128 se añaden *bits* de relleno *(padding)*. Si se considera los valores utilizados en la definición del protocolo para realizar la configuración remota analizado en la sección 5.2.4, el tamaño de la trama cifrada se incrementaría en 34 *bytes*. Si se aplica la Ecuación 5.2 considerando este incremento, el retardo introducido por el proceso de transmisión desde el *MCD* al *PHY* para las tramas *SV* es de 1,71 μs.

$$L_{MCD\_PHY_{SV\_max}} = \frac{Tamaño\_trama\_bits}{Tasa\_de\_transmisión} = \frac{(180 + 34) * 8}{10^9} = 1,71\,\mu s \qquad (5.5)$$

Para el caso de las tramas *GOOSE* el máximo retardo introducido es de 2,27 μs.

$$L_{MCD\_PHY_{GOOSE\_max}} = \frac{Tamaño\_trama\_bits}{Tasa\_de\_transmisión} = \frac{(250 + 34) * 8}{10^9} = 2,27\,\mu s \quad (5.6)$$

La arquitectura propuesta utiliza el *Advanced Encryption Standard (AES)* como algoritmo de cifrado. Hay cinco modos AES de operación recomendadas por el *National Institute of Standards and Technology (NIST)* [170]: *Electronic Code-Book (ECB)*, *Cipher-block chaining (CBC)*, *Output FeedBack (OFB)*, *Cipher FeedBack (CFB)* y *Counter (CTR)*. *OFB*, *CFB* junto con *CTR* utilizan un único componente de cifrado, tanto para cifrar como para descifrar los mensajes.

Los modos de funcionamiento *AES* ofrecen mecanismos de confidencialidad pero no dispone de medio de autenticación de la información. La forma más común de resolver esta carencia es combinar un algoritmo de cifrado con un *Message Authentication Code (MAC)*, como *HMAC* o *CBC-MAC*. Sin embargo, la integración correcta de estas primitivas genera una solución sub-óptima en términos de utilización de recursos y de la latencia total requerida para el procesamiento de las tramas [171].

Para resolver el problema de la confidencialidad y la autenticación, en la última década se han desarrollado los modos de operación *Authenticated Encryption (AE)*. En particular, aquellos algoritmos que permiten utilizar datos adicionales para autenticación *(Encryption Algorithm that allow Additional Data authentication , AEAD)*. Entre ellos, podemos mencionar *OCB*, *CCM* y *GCM*.

- *Offset Codebook Mode (OCB),* es un algoritmo muy rápido, que produce una baja sobrecarga en el algoritmo de cifrado utilizado, pero tiene el inconveniente de que utiliza tecnologías patentadas.
- *Counter with CBC-MAC (CCM)*, es considerablemente más lento que la *OCB*, pero esta disminución en el rendimiento es aceptable si se considera que no hay necesidad de licencias por las patentes.
- *Galois Counter Mode (GCM)*, puede aprovechar al máximo el procesamiento en paralelo reduciendo así la sobrecarga. Por lo tanto, es más eficiente que *OCB* y no tiene el inconveniente de patentes.

Teniendo en cuenta las opciones de *AEAD* disponibles y sus limitaciones, se ha elegido *AES-GCM* como la opción más adecuada para la arquitectura planteada. La implementación en *hardware* propuesto del algoritmo de cifrado *AES-GCM* toma como base el núcleo desarrollado por *Rudolf* [172], que requiere diez ciclos de reloj para realizar el cifrado y la autenticación, para un tamaño de clave de 128 *bits*. *AES-GCM* ha sido seleccionado debido a sus capacidades criptográficas, utilización de recursos y los rendimientos alcanzados, especialmente en implementaciones *hardware*.

La Figura 5.8 representa la arquitectura básica del algoritmo de cifrado *AES*, en donde se implementa una unidad de cifrado con lógica combinacional y se usa de manera iterativa para el número de rondas. Este método de implementación del cifrado por bloques *AES* se ha considerado para estimar la latencia del *MCD*. Concretamente, en la Ecuación 5.7 se define la latencia del bloque *AES-GCM*, llamada $L_{AES-GCM}$, donde N_r es el número de rondas y T_{MCD} es el período de reloj del *MCD*.

$$L_{AES-GCM} = N_r \cdot T_{MCD} \tag{5.7}$$

Los algoritmos criptográficos que se implementan en *hardware* también se caracterizan utilizando el *throughtput*, que se define como el número de *bits* cifra-

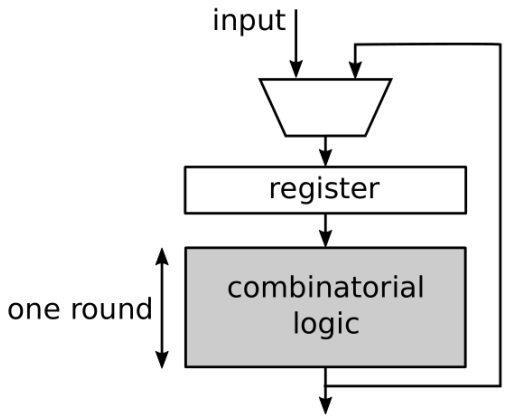


Figura 5.8: Arquitectura básica para cifrado por bloques

dos/descifrados en una unidad de tiempo y se relaciona con la latencia mediante la Ecuación 5.8 [173]. Donde Tb es el tamaño del bloque y Num_BPS es el número de bloques procesados simultáneamente.

$$Throughtput_{cifrado} = \frac{Tb \cdot Núm\_BPS}{Latencia_{cifrado}} \tag{5.8}$$

En la arquitectura *SoC* propuesta, el *MCD* utiliza el cifrado por bloques *AES-GCM* para calcular el texto cifrado y la etiqueta de autenticación de las tramas *Ethernet* a medida que se desplazan por la interfaz *GMII*. A diferencia de la encriptación del sistema de archivos, en la que se deben cifrar o descifrar grandes cantidades de datos, en la interfaz *GMII* para *Gigabit Ethernet* los datos se capturan en segmentos de 8 bits. Por lo tanto, no tiene sentido incluir en el cálculo la opción de procesar varios bloques simultáneamente. Por consiguiente, sólo se procesa un bloque cada vez con esta arquitectura y el rendimiento se calcula mediante la Ecuación 5.9.

$$Throughtput_{AES-GCM} = \frac{Tb}{L_{AES-GCM}} = \frac{128}{N_r \cdot T_{MCD}} \tag{5.9}$$

Como se indica en las Ecuaciones 5.10 y 5.11, la frecuencia mínima de operación

del *MCD* es la que garantiza un funcionamiento óptimo del sistema a un *Gigabit*. Si se aplican los datos del núcleo desarrollado por *Rudolf* [172] en la Ecuación 5.11, en donde el tamaño de bloque es ($Tb = 128$) y ($Nr = 10$) la frecuencia mínima de operación del *MCD* es de $78,125\,MHz$.

$$Throughtput_{AES-GCM} = \frac{128 \cdot f_{MCD_{min}}}{N_r} = {}^{1Gbit}/_{s} \tag{5.10}$$

$$f_{MCD_{min}} = \frac{10^9 \cdot N_r}{128} \tag{5.11}$$

Del mismo modo, la latencia máxima del algoritmo *AES-GCM* puede calcularse a partir de la frecuencia mínima, tal como se indica en la Ecuación 5.12, lo que da como resultado $128\,ns$ independientemente del tamaño de la clave.

$$L_{AES-GCM_{max}} = \frac{N_r}{f_{MCD_{min}}} = \frac{N_r \cdot 128}{N_r{,}10^9} = 128\,ns \tag{5.12}$$

Para determinar la latencia máxima introducida por el proceso de cifrado de toda la trama utilizando el algoritmo *AES_GCM*, se aplica la Ecuación 5.13. En este caso, al tamaño de la trama hay que restarle 22 *bytes* que corresponden: 6 a la dirección *MAC* origen, 6 a la dirección *MAC* destino, 2 al campo *Ethertype* y 8 al encabezado de la estructura de las tramas *(SV, GOOSE)*.

$$L_{AES-GCM_{TOTAL}} = L_{AES-GCM} \cdot \frac{Tamaño\_trama\_bits - 22}{Tb} \tag{5.13}$$

$$L_{AES-GCM} = L_{AES-GCM_{max}} = 128\,ns$$

$$L_{AES-GCM_{SV\_max}} = 128\,ns \cdot \frac{Tamaño\_trama\_bits - 22}{Tb}$$

Para el caso de las tramas *SV* con tamaño de trama de 180 *bytes* la latencia máxima es $1{,}26\,\mu s$.

$$L_{AES-GCM_{SV\_max}} = 128\,ns \cdot \frac{(180-22)\cdot 8}{128} = 1,26\,\mu s$$

Para el caso de las tramas *GOOSE* con tamaño de trama de 250 *bytes* la latencia máxima es 1,82 μs.

$$L_{AES-GCM_{SV\_max}} = 128\,ns \cdot \frac{(250-22)\cdot 8}{128} = 1,82\,\mu s$$

En resumen aplicando los resultados obtenidos en la Ecuación 5.1, el retardo máximo introducido por el *MCD* para las tramas *SV* y *GOOSE* es:

$$L_{MCD} = L_{CPU\_MCD} + L_{AES\_GCM} + L_{MCD\_PHY} \tag{5,1}$$

$$L_{MCD_{SV\_max}} = 1,44 + 1,26 + 1,71 = 4,41\,\mu s$$

$$L_{MCD_{GOOSE\_max}} = 2,00 + 1,82 + 2,27 = 6,09\,\mu s$$

Modelado

Para determinar la latencia y el ancho de banda necesario al utilizar el algoritmo de cifrado y autenticación *AES-GCM* en las tramas *GOOSE* y *SV*, se utilizó el *software VisualSim Architect* [174], que es un *software* de modelado y simulación de sistemas embebidos. Utilizando este entorno, es posible depurar, simular y analizar el rendimiento y el comportamiento de una arquitectura de un sistema eléctrico antes de que el sistema real sea implementado. En la simulación se puede incluir elementos del sistema tales como microprocesadores, memorias, buses, interfaces, entre otros. También incluye la posibilidad de realizar simulaciones incorporando elementos de red como *router* y *switch*.

Como primer paso hay que definir y modelar la arquitectura del *MCD* que se utilizará para ofrecer seguridad a las tramas *GOOSE* y *SV*. La arquitectura que

se utilizó se muestra en la Figura 5.9. El *MCD* consta de un bloque *ETH_ENC* que se encarga de recibir/transmitir y procesar las tramas *Ethernet* sin cifrado, una memoria *MEN* para almacenar las tramas *Ethernet*, un bloque *ETH_DEC* que se encarga de recibir/transmitir y procesar las tramas *Ethernet* con cifrado, y un bloque de cifrado y descifrado con el algoritmo *AES-GCM*.

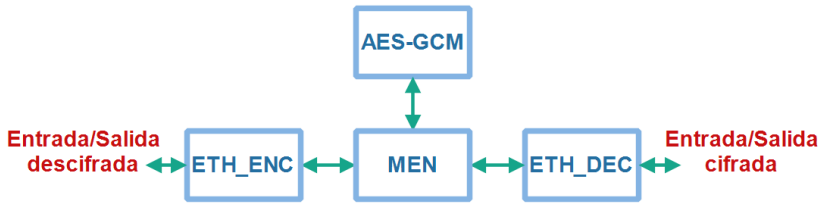


Figura 5.9: Arquitectura del modelo *MCD*

Los parámetros que se configuraron en los diferentes bloques de la arquitectura mostrada en la Figura 5.9 se detallan en la Tabla 5.5. La frecuencia de operación para los bloques del sistema es $125\,MHz$.

Tabla 5.5: Parámetros generales de configuración de la arquitectura.

Hardware	**Reloj (*MHz*)**	**Ancho del bus (*byte*)**
ETH_ENC	125	1
ETH_DEC	125	1
MEM	125	16
AES_GCM	125	16

Una vez definida la arquitectura del *hardware* el siguiente paso es definir el comportamiento de las tareas que se ejecutan en la arquitectura. En la Figura 5.10 se observan 2 tareas, en la primera se modela el flujo de los datos de las tramas *SV* y la segunda representa el comportamiento de las tramas *GOOSE*.

En donde:

1. Se generan los eventos *SV* o *GOOSE*, especificando cantidad de tramas que se generan por segundo y el tamaño. Para una frecuencia de red de $50\,Hz$, las tramas *SV* según las especificaciones del estándar *IEC 61850-90-4* se generan a razón de 4000 muestras/s, con tamaños de trama entre 160 y 180 *bytes*. Para el caso de las tramas GOOSE se generan entre 20 y 140 $tramas/s$, con tamaños de trama entre 92 y 250 *bytes* [169].

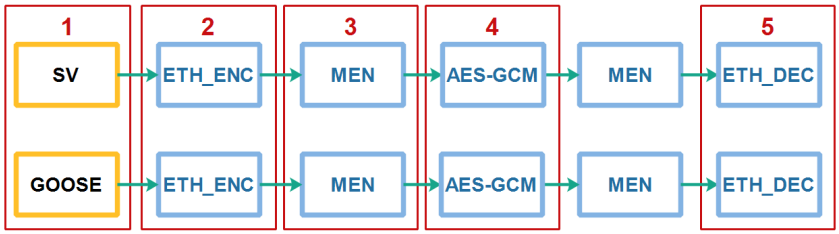


Figura 5.10: Modelo del flujo de procesamiento de las tramas *GOOSE*, *SV*

2. Se calcula el tiempo necesario para la recepción/transmisión de una trama *Ethernet* sin cifrado aplicando la Ecuación 5.14.

$$T_{ETH\_ENC} = \frac{Tamaño\_trama\_bits}{Ancho\_bus\_datos \cdot frecuancia\_de\_operación} \quad (5.14)$$

$$T_{ETH\_ENC} = \frac{Tamaño\_trama\_bits}{8 \cdot 125 \cdot 10^6}$$

3. Se calcula el tiempo necesario para pasar una trama de datos desde el módulo *ETH_ENC* al *AES_GCM* aplicando la Ecuación 5.15. Para el modelo se consideró una memoria *FIFO* en la que es necesario dos ciclos de reloj para transferir un dato de 128 *bits*.

$$T_{MEN} = \frac{Tamaño\_trama\_bits \cdot número\_ciclos}{Ancho\_bus\_datos \cdot frecuancia\_de\_operación} \quad (5.15)$$

$$T_{MEN} = \frac{Tamaño\_trama\_bits \cdot 2}{128 \cdot 125 \cdot 10^6}$$

4. Se calcula el tiempo para realizar el cifrado o descifrado de los datos enviados en las tramas *Ethernet* aplicando la Ecuación 5.13. *Tb* es el tamaño del bloque y es igual a 128 *bits*. *N* es el numero de ciclos para realizar el cifrado de un bloque de 128 *bits* y f_{MCD} es la frecuencia de operación del módulo *AES_GCM*.

$$L_{AES-GCM_{TOTAL}} = L_{AES-GCM} \cdot \frac{Tamaño\_trama\_bits - 22}{Tb} \quad (5.16)$$

$$L_{AES-GCM} = \frac{N}{f_{MCD}} = \frac{10}{125 \cdot 10^6} = 80\,ns$$

$$L_{AES-GCM} = 80\,ns \cdot \frac{Tamaño\_trama\_bits - 22}{128}$$

5. Se calcula el tiempo necesario para la recepción/transmisión de una trama *Ethernet* con cifrado aplicando la Ecuación 5.14.

$$T_{ETH\_DEC} = \frac{Tamaño\_trama\_bits + 34}{Ancho\_bus\_datos \cdot frecuancia\_de\_operación}$$

Los datos con los que se configuraron los diferentes bloques fueron seleccionados considerando una aplicación con una frecuencia de red eléctrica de $50\,Hz$, y un tiempo de simulación de $0{,}1\,s$. En la Tabla 5.6 se resumen los parámetros que se utilizaron para realizar la simulación, y las referencias utilizadas para la justificación de los valores.

Tabla 5.6: Parámetros de simulación

Parámetro	**Valor**	**Origen_información**
Frecuencia de Red	$50\,Hz$	
Frecuencia *SV*	4000 $muestras/s$	*IEC 61850-90-4* (Pag. 32) [169]
Tamaño de tramas *SV*	(160 -180) *bytes*	*IEC 61850-90-4* (Pag. 32)
Frecuencia *GOOSE*	(20 - 140) $eventos/s$	*IEC 61850-90-4* (Pag. 104)
Tamaño de tramas *GOOSE*	(92 - 250) *bytes*	*IEC 61850-90-4* (Pag. 104)
Velocidad *Ethernet*	$1\,Gbps$	
AES-GCM encrypt	10 ciclos (128 bits)	[172]

Los datos que se obtienen de las simulaciones son los referentes a la latencia *(μs)*, que es el tiempo que necesita el *MCD* para cifrar o descifrar una trama *Ethernet* (*GOOSE* o *SV*).

El tiempo de procesamiento necesarios para el cifrado de las tramas *SV* para velocidad de transmisión *Ethernet* de $1\,Gbps$ se muestran en la Figura 5.11. El tiempo necesario para realizar recepción, procesamiento y cifrado de una trama *Ethernet SV* de 160 *bytes* es de $3,87\,\mu s$ y para una trama de 180 *bytes* es de $4,33\,\mu s$.

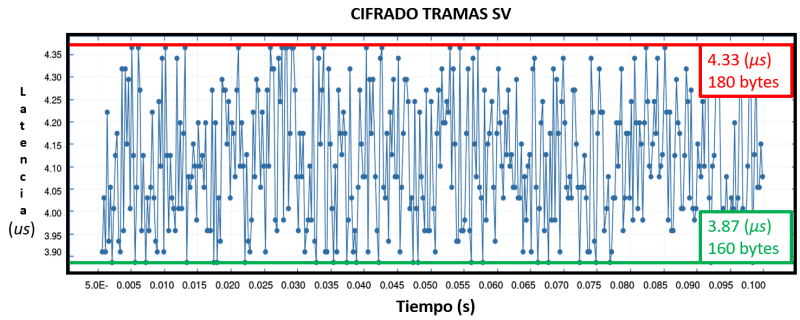


Figura 5.11: Tiempo para recepción, procesamiento y cifrado de tramas *Ethernet SV (160-180 bytes)*

El tiempo de procesamiento necesarios para el descifrado de las tramas *SV* para velocidad de transmisión *Ethernet* de $1\,Gbps$ se muestran en la Figura 5.12. El tiempo necesario para realizar recepción, procesamiento y descifrado de una trama *Ethernet SV* de 160 *bytes* es de $3,95\,\mu s$ y para una trama de 180 *bytes* es de $4,41\,\mu s$.

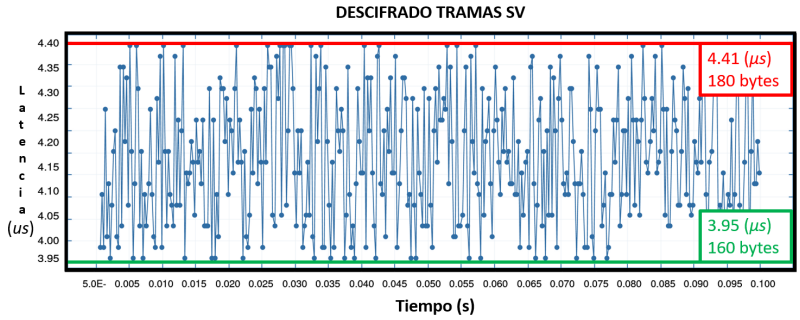


Figura 5.12: Tiempo para recepción, procesamiento y descifrado de tramas *Ethernet SV (160-180 bytes)*

El tiempo de procesamiento necesarios para el cifrado de las tramas *GOOSE* para velocidad de transmisión *Ethernet* de $1\,Gbps$ se muestran en la Figura 5.13. El tiempo necesario para realizar recepción, procesamiento y cifrado de una trama *Ethernet GOOSE* de 92 *bytes* es de $2,31\,\mu s$ y para una trama de 250 *bytes* es de $5,94\,\mu s$.

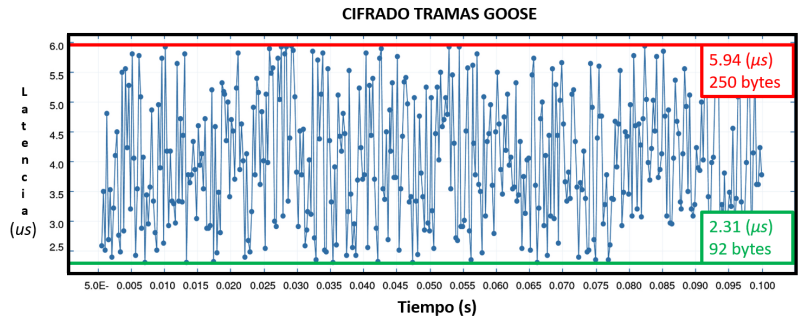


Figura 5.13: Tiempo para recepción, procesamiento y cifrado de tramas *Ethernet GOOSE (92-250 bytes)*

El tiempo de procesamiento necesarios para el descifrado de las tramas *GOOSE* para velocidad de transmisión *Ethernet* de $1\,Gbps$ se muestran en la Figura 5.14. El tiempo necesario para realizar recepción, procesamiento y descifrado de una trama *Ethernet GOOSE* de 92 *bytes* es de $2,39\,\mu s$ y para una trama de 250 *bytes* es de $6,02\,\mu s$.

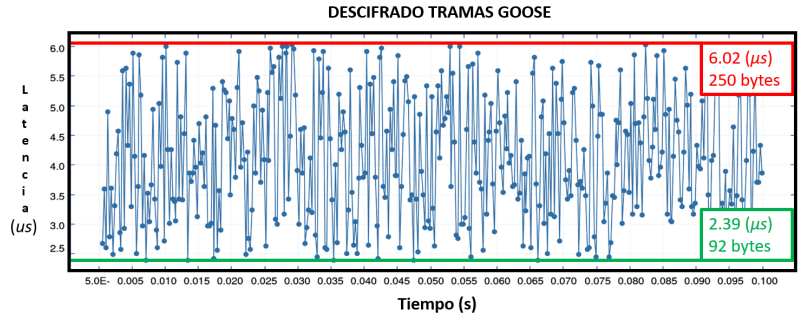


Figura 5.14: Tiempo para recepción, procesamiento y descifrado de tramas *Ethernet GOOSE (92-250 bytes)*

En la Tabla 5.7, se presentan los resultados obtenidos en las simulaciones para tramas *GOOSE* y *SV*, con y sin seguridad (cifrado), a velocidad *Ethernet* de $1\,Gbps$. *Lat*_SV_160 representa el tiempo que *MCD* necesita para la recepción, procesado y cifrado/descifrado de una trama *SV* con un tamaño de 160 *bytes*, y *Lat*_SV_180 representa el tiempo para un tamaño de trama *SV* de 180 *bytes*.

*Lat*_GOOSE_92 y *Lat*_GOOSE_250 representan el tiempo que MCD necesita para la recepción, procesado y cifrado/descifrado de una trama *GOOSE* con un tamaño de 92 y 250 *bytes* respectivamente.

Tabla 5.7: Resultados de simulaciones para cifrado y descifrado de tramas *SV* y *GOOSE*

Tramas (1 *Gbps*)	**Cifrado**	**Descifrado**
Lat_*SV*_160 (μs)	3,87	3,95
Lat_*SV*_180 (μs)	4,33	4,41
Lat_*GOOSE*_92 (μs)	2,31	2,39
Lat_*GOOSE*_250 (μs)	5,94	6,02

Diseño del *IP core MCD*

El diagrama de bloques de la arquitectura del *hardware* del *IP core MCD* propuesto se observa en la Figura 5.15. El principio de funcionamiento es el siguiente:

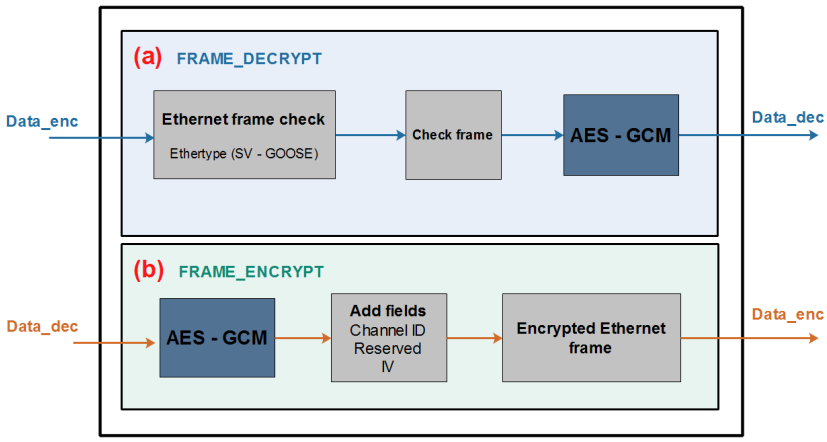


Figura 5.15: Diagrama de bloques de la arquitectura del *hardware* para cifrado y descifrado de las tramas *Ethernet* (*SV* y *GOOSE*)

Cuando el *MCD* recibe una nueva trama *Ethernet*, se comprueba si esta contiene un *Ethertype* válido y tiene un tamaño de trama dentro de los límites establecidos en la configuración del *IP*. Si es así, la trama es almacenada en una *FIFO* de entrada con capacidad para varias tramas. Las tramas que han sido almacenadas

en la *FIFO* son tratadas una a una, y son enviadas al módulo *AES-GCM* que se encargará de descifrar y autenticar la trama. Finalmente la trama *Ethernet* se reconstruye con los datos descifrados y se envía a las aplicaciones. Por lo tanto, a nivel de aplicación el uso del cifrado es transparente, por lo que no es necesario modificar los programas para trabajar con esta funcionalidad. Para el caso de realizar el cifrado, primero se obtienen los datos *(payload)* de la trama *Ethernet* original y se realiza el cifrado, luego se reconstruye la trama *Ethernet* con en *payload* cifrado y los campos necesarios para descifrar y autenticar la trama *Ethernet*, y por último la trama es transmitida.

Se realizaron simulaciones utilizando *Vivado Simulator*, con el propósito de verificar el funcionamiento y determinar los tiempos de latencia que el *MCD* introduce en un diseño. Se realizaron simulaciones para tramas *Ethernet* cifradas de un tamaño de 160 *bytes*. A 100 $Mbps$ el *MCD* descifra la trama *Ethernet* en $9,29\,\mu s$ y a $1\,Gbps$ la trama es descifrada en $3,42\,\mu s$.

La Figura 5.16 muestra un diagrama de bloques de la implementación de un *CPS Gateway* agregando el módulo para cifrado de las tramas de Capa 2 *(GOOSE* y *SV)*.

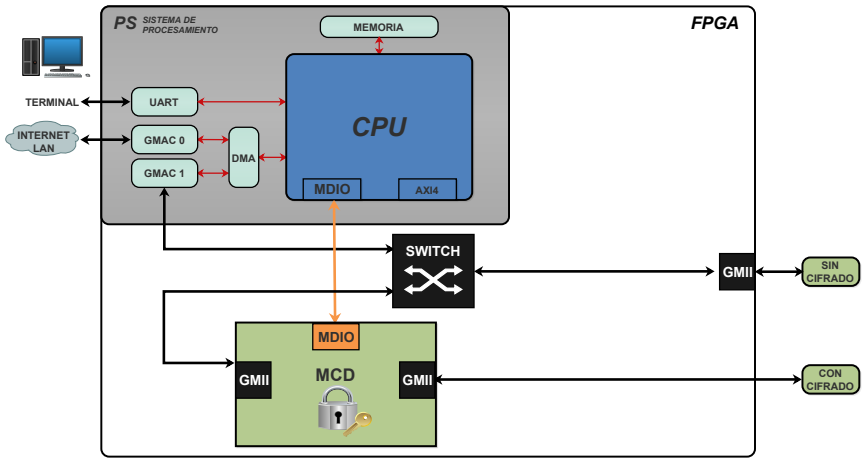


Figura 5.16: Arquitectura del *CPS Gateway* segura

La cantidad de recursos de la *FPGA* necesarios para implementar el módulo *MCD* descrito en la Figura 5.15 se presenta en la Tabla 5.8. Como se observa, para la implementación del módulo *MCD* se requiere en el peor de los casos 62,60 % del

total de recursos disponibles de la *FPGA*.

Tabla 5.8: Recursos de la *FPGA* utilizados en la implementación del módulo *MCD*

Recurso	Módulo *MDC*	Disponible	Utilización (%)
Slice LUTs	33304	53200	62,60
Slice Registers	27185	106400	25,55
Block RAM	85,5	140	61,07

5.3. Arquitectura de un *Cyber Physical System Gateway*

Una vez analizados cada uno de los requerimientos operativos exigidos por la *Smart Grid* y planteadas las arquitecturas para dar solución a cada uno de ellos, en esta sección se plantea una arquitectura general que integra en un mismo dispositivo todos los componentes definidos en la sección 5.2. En este sentido, en la Figura 5.17 se muestra un diagrama de bloques de la arquitectura del *CPS Gateway* propuesto para la *Smart Grid*. En la arquitectura están integradas las distintas soluciones expuestas en la sección anterior y se pueden identificar seis módulos, los cuales se presentan a continuación:

1. El módulo de procesamiento *(PS)* permite ejecutar el *software* necesario para gestionar todos los componentes de la arquitectura, ejecutar librerías específicas de un protocolo de comunicaciones y realizar procesamiento y análisis de datos. El *PS* dispone de varias interfaces *Ethernet* y serie *(RS-232, I^2C)* para establecer comunicación con el exterior o con los módulos internos *(HSR/PRP, IEEE 1588,* etc.*)*. En el módulo de procesamiento se puede identificar un puerto *Ethernet* de 1 *Gbps (GMAC0)*. Este puerto se utilizará como interfaz para acceder a una red local o a *Internet*, para proporcionar al sistema acceso a servicios tales como *web, FTP,* base de datos, *cloud*, entre otros. Adicionalmente, la *GMAC1* se utiliza para interconectar el *PS* con los *IP cores* que se implementan en el *PL*. Las interfaces serie *(UARTx)* se utilizan para implementar comunicaciones industriales a través de buses de campo serie *(Modbus, Profibus)*, o como terminal para monitorizar, configurar y controlar el *CPS Gateway*.

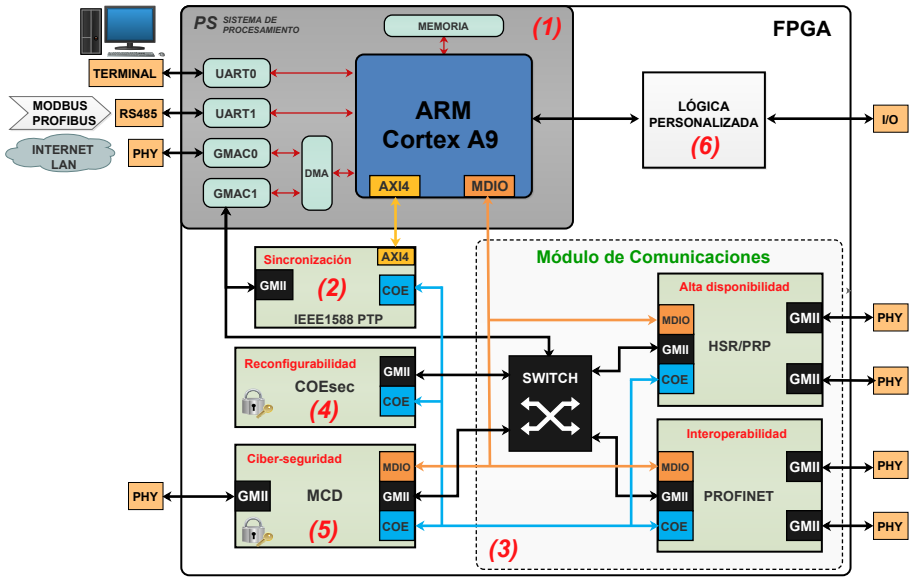


Figura 5.17: Diagrama de bloques de la arquitectura de un *Cyber Physical System Gateway* para la *Smart Grid*

2. El módulo *IEEE 1588 PTP* se utiliza para soportar aplicaciones con bajos tiempos de sincronización. Este módulo proporciona un mecanismo de sincronización en el rango de nanosegundos. El módulo *IEEE 1588 PTP* se utiliza para realizar el *timestamping* y para dar soporte al protocolo *HSR*.

3. El módulo de comunicaciones integra un módulo que gestiona las comunicaciones de alta disponibilidad, un módulo de comunicaciones industriales y un *switch Ethernet* para interconectar los diferentes módulos. El módulo de comunicaciones consta de cuatro interfaces *Ethernet*, dos de las cuales se utilizan para dar soporte a los protocolos de comunicación de alta disponibilidad *(HSR / PRP)*, y las otras dos se utilizan como puertos para comunicaciones industriales basadas en *Ethernet (Profinet, EtherCAT, EtherNet / IP*, entre otros*)*

4. El Módulo *COEsec* permite realizar la reconfiguración de los *IP cores* utilizados en la arquitectura del *CPS Gateway*. El módulo propuesto es compacto y no necesita una *CPU* para gestionar las tramas de configuración. Las datos de configuración se transmiten a nivel de Capa 2 e incorpora un mecanismo de seguridad para la autenticación y el cifrado de las tramas. Las

funcionalidades de este módulo pueden ser reemplazadas si la arquitectura incorpora un *PS*.

5. El módulo para ciber-seguridad para los datos críticos de comunicación en la *Smart Grid* es un sistema criptográfico capaz de cifrar, descifrar y autenticar las tramas *Ethernet* de Capa 2 *(GOOSE y SV)* utilizando algoritmos *AES* implementados en hardware. Este módulo utiliza criptografía de clave simétrica, que se basa en el algoritmo *AES-GCM* de 128 *bits* para proporcionar cifrado y autenticación de datos.

6. En la arquitectura también es ha considerado la posibilidad de incorpora módulos genéricos para añadir más funcionalidades al CPS Gateway, como por ejemplo interfaces para la lectura de sensores utilizando comunicaciones I^2C o convertidores analógico-digitales *(A/D)*.

Finalmente, en la Tabla 5.9 se resume los recursos necesarios para implementar la arquitectura del *CPS Gateway* incluyendo todos los módulos analizados. Como se puede observar para la implementación es necesario un *SoC* que incluya una *FPGA* con capacidad mínima de 68353 *Slice LUTs,* 63295 *Slice Registers* y 196 *Block RAM (RAMB36).* Esta cantidad de recursos sobrepasa la capacidad de la *FPGA Zynq XC7Z020* que se tomo como base para la estimación de recursos de los diferentes módulos.

Tabla 5.9: Recursos de la *FPGA* utilizados en la implementación del módulo *MCD*

Recurso	Módulo *PTP*	Módulo *HSR-RPR*	Ethernet *Switch*	Módulo *Profinet*	Módulo *COEsec*	Módulo *MDC*	Total
Slice LUTs	1408	18356	6897	3640	4748	33304	68353
Slice Registers	2340	17115	8321	4196	4138	27185	63295
Block RAM	6	32	31	36	5.5	85,5	196

Con el fin de superar el problema de los recursos limitados, se podría elegir una *FPGA* con más recursos. Por ejemplo, en lugar del dispositivo *Zynq XC7Z020*, se puede emplear una *FPGA* superior dentro de la familia *Zynq-7000*, por ejemplo el dispositivo *Zynq XC7Z030* que contiene una *PL* equivalente a la de la familia *Kintex-7*, esta *FPGA* tienen una capacidad de 78600 *Slice LUTs,* 157200 *Slice Registers* y 265 *Block RAM (RAMB36)* [175].

5.4. Resumen

Debido a las necesidades de computación y a la flexibilidad a nivel de *hardware* que debe tener la arquitectura, en este trabajo se ha propuesto la utilización de una plataforma *SoC* reconfigurable para la implementación de un *CPS Gateway* para la *Smart Grid*. En la arquitectura propuesta se ha considerado la implementación de una serie de medidas que permitan afrontar todos los requerimientos operativos de los dispositivos para la Smart Grid identificados en el capítulo 3.

En este sentido, la arquitectura del *CPS Gateway* propuesto incorpora un Sistema de Procesamiento *(PS)* que permite la ejecución simultánea de diferentes aplicaciones, el software de gestión de la arquitectura y para realizar el procesamiento y análisis de datos.

El estándar *IEEE 1588 Precision Time Protocol (PTP)* es la mejor opción a ser considerada en la arquitectura propuesta de *CPS Gateway,* ya que garantiza una sincronización en el orden de los nanosegundos. En este sentido, se ha considerado utilizar un *IP core PTP* para realizar la sincronización.

En cuanto a la alta disponibilidad, como se menciono en la sección 3.2.3 la implementación en *software* de *HSR* o *PRP* es rápida y sencilla, pero tienen la desventaja que sobrecargan al procesador. Para evitar la pérdida de rendimiento en la arquitectura del *CPS Gateway*, se ha considerado la implementación de *HSR* y *PRP* en *hardware (FPGA)* mediante el uso de *IP cores*.

Para los buses de campo basados en *Ethernet* que requieren funcionalidades operativas en tiempo real es necesario considerar el uso de módulos externos que integran las interfaces y la pila del protocolo o *IP cores* para ser utilizados en las implementaciones basadas en *FPGA*.

Para la configuración remota de los *IP cores* implementados en la *PL* se propone la utilización de un *IP core* y un protocolo de comunicaciones que se transmite a través del mismo canal *Ethernet* de la infraestructura de red implementada en la *FPGA (HSR, IEEE 1588, etc.)*. El *IP core* propuesto es compacto y no necesita una *CPU* para gestionar las tramas de configuración.

El uso de la criptografía de clave simétrica se ha considerado como un mecanismo de ciber-seguridad para tramas de Capa 2 como *GOOSE* y *SV*. En este senti-

do, se propone implementar una solución basada en *hardware* que proporcione un equilibrio entre rendimiento y recursos. Este enfoque minimiza la latencia y aumenta la eficiencia en comparación con otras soluciones criptográficas basadas en *software*. El módulo de ciber-seguridad propuesto se basa en el algoritmo de encriptación *AES-GCM* de 128 *bits*, este algoritmo ha sido seleccionado por sus capacidades criptográficas, la utilización de recursos y el rendimiento alcanzado, especialmente en implementaciones de hardware, aunque con pequeños cambios podría soportar casi cualquier algoritmo criptográfico.

Capítulo 6

Validación del *Cyber Physical System Gateway* sobre plataformas *System-on-Chip*

6.1. Introducción

Considerando las características a nivel de *hardware* que deben tener las arquitecturas definidas en el capítulo 5. En este capítulo se describen las pruebas experimentales que se realizaron para validar las arquitecturas propuestas.

Para la implementación y validación experimental es necesario utilizar un dispositivo donde la unidad de procesamiento y sus periféricos junto con la lógica reconfigurable están integrados en un solo circuito integrado. El dispositivo *Zynq* de *Xilinx* es un buen ejemplo, ya que permite desarrollar potentes sistemas embebidos con características específicas como la que se ha definido a lo largo del capítulo 5. En la sección 6.2 se describen las características del *SoC Zynq* de *Xilinx* y del *hardware* utilizado para realizar la experimentación y validación de

la arquitectura propuesta.

Para demostrar la aplicabilidad de la arquitectura del *CPS Gateway* en entornos industriales como el eléctrico, se implementaron tres arquitecturas. En la sección 6.3, se describe la implementación de una arquitectura para validar el funcionamiento del *CPS Gateway* en cuanto a interoperabilidad, alta disponibilidad y sincronización.En la sección 6.4, se describe un diseño que permite la configuración remota de los *IP cores* utilizados en la arquitectura. Finalmente, en la sección 6.5, se plantea una arquitectura para demostrar la encriptación de tramas *Ethernet (GOOSE y SV)* de Capa 2, con el objetivo de cuantificar el retardo que se añade al utilizar el cifrado.

6.2. Descripción del *hardware* de ensayos

La flexibilidad a nivel de *hardware* y *software* que debe tener la arquitectura del *CPS Gateway* planteada hace que las *FPGAs* y los dispositivos reconfigurables como los *SoC* de última tecnología sean los mejores candidatos al momento de realizar su implantación. Las *FPGAs* son dispositivos versátiles que pueden configurarse para implementar cualquier sistema digital, incluyendo procesadores embebidos, si fueran necesario. Las *FPGAs* también pueden ser reconfiguradas tantas veces como se desee, ofreciendo así una plataforma más flexible que los circuitos integrados para aplicaciones específicas *(Application Specific Integrated Circuit, ASIC)*.

En este trabajo se plantea la implementación de un *CPS Gateway* utilizando las *FPGAs* y las herramientas de desarrollo de *Xilinx* y complejos *IP cores* desarrollados por terceros. La elección de la *FPGA* ha estado condicionada por unos requisitos mínimos impuestos por los escenarios de pruebas planteados. Entre las condiciones mínimas que la *FPGA* debe cumplir, en lo que al escenario se refiere, se encuentran, principalmente la necesidad de una unidad de procesamiento con capacidad de ejecutar el sistema operativo *Linux* y que disponga de interfaces de comunicación *RS-232* y *Ethernet* para monitorizar y controlar el funcionamiento del sistema. Por otro lado, la *FPGA* debe tener una sección de lógica programable *(Programmable Logic, PL)* de suficiente capacidad como para implementar las comunicaciones y funcionalidades específicas del entorno de *Smart Grid* discutido en el capítulo 3.

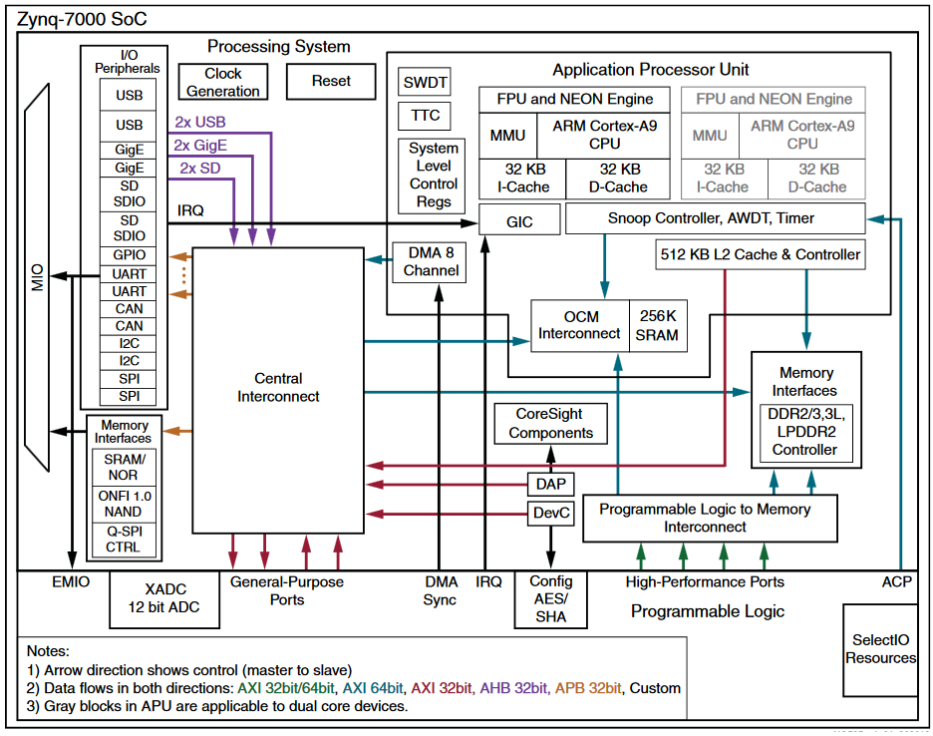
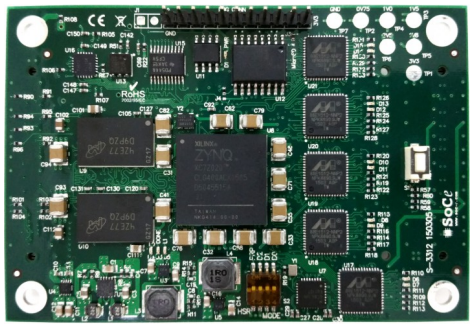


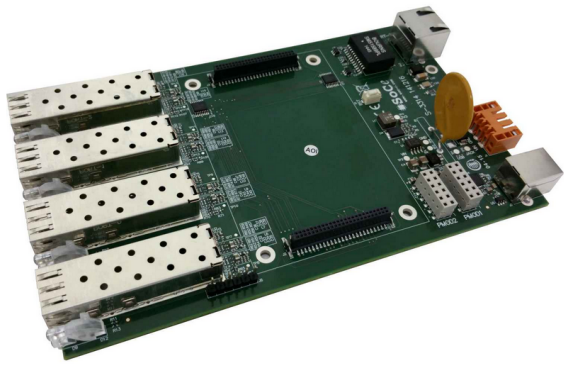
Figura 6.1: Zynq diagrama de bloques [176].

En este sentido, para realizar la validación de la arquitectura se utilizó un *All Programmable SoC (AP-SoC)* de *Xilinx*, en concreto de la familia *Xilinx Zynq-7000*. Este *SoC* consta de dos partes principales: un sistema de procesamiento *(Processing System, PS)* que contiene un procesador *ARM Cortex-A9* de doble núcleo y una *PL* que es equivalente a una *FPGA* de última generación de 28nm. También cuenta con memoria integrada, y una amplia gama de potentes periféricos e interfaces de comunicaciones de alta velocidad. Entre otros, controladores *Gigabit Ethernet*, controladores de memoria, bus *CAN*, etc. La sección *PL* es ideal para implementar subsistemas lógicos, aritméticos y de procesamiento de datos de alta velocidad, mientras que el *PS* es capaz de ejecutar rutinas de *software* incluyendo la integración de sistemas operativos. Esta versatilidad permite que las funcionalidades de cualquier sistema diseñado puedan separarse de forma adecuada entre el *hardware* y el *software*. La comunicación entre el *PS* y la *PL* se realiza utilizando *Advanced eXtensible Interface (AXI)*, que es un bus de comu-

nicaciones estandarizado de 32/64 bits. El diagrama de bloques de un dispositivo *Zynq-7000 AP SoC* se representa en la Figura 6.1 [176]. Cada dispositivo de la serie *Zynq-7000* contiene el mismo *PS*, pero los recursos *PL* y *Input/Output (IO)* varían de un dispositivo a otro para adaptarse a los diferentes requisitos de las aplicaciones.



(a)



(b)

Figura 6.2: *Hardware* de desarrollo: a) Smart-zynq. b) Smart-zynq carrier

Como sistema de desarrollo electrónico *hardware* se utilizó la placa *Smart-zynq* fabricada por la empresa **SoC***e* , representada en la Figura 6.2. Esta dispone de una *FPGA* Zynq (XC7Z020) de *Xilinx* y todas las unidades de memoria y periféricos necesarios para ejecutar un sistema *Linux* completo. También se utilizó la placa de expansión de periféricos *Smart-zynq carrier* [177], que proporciona una serie de conectores para ampliar las posibilidades de implementación de una

gran variedad de soluciones de comunicaciones de alta disponibilidad, ya que dispone de 5 puertos *Ethernet*, que soportan velocidades de 10, 100 y 1000 $Mbps$.

Para la gestión de todos los recursos de *hardware*, a nivel de *software* en el módulo de procesamiento se instaló el sistema operativo *Linaro (linaro-vivid-developer-20150914-710)*, que es una versión de *Linux* para sistemas embebidos. El *software* necesario para cargar el *Bitstream*, el árbol de dispositivos y la *FSBL* se desarrolla en *Xilinx SDK*. Finalmente, se compila la infraestructura del sistema con la instalación de las herramientas *software* necesarias, como, por ejemplo *Python, GCC*.

Para demostrar la aplicabilidad de la arquitectura del *CPS Gateway* en entornos industriales como el eléctrico, se implementaron tres topologías *SoC* para el dispositivo *Zynq*. En la primera, se verifica el funcionamiento del *CPS Gateway* en cuanto a interoperabilidad, alta disponibilidad y sincronización. En la segunda, se plantea un mecanismo de configuración remota de los *IP cores* utilizados en arquitectura. Finalmente, se implementó un entorno de pruebas para demostrar la encriptación de tramas *Ethernet* (*GOOSE* y *SV*) de Capa 2 y determinar el retardo que esto añade a las comunicaciones.

6.3. Interoperabilidad, alta disponibilidad y sincronización

Los cambios en la estructura, la organización y la incorporación de las *TIC* en red eléctrica, aumentan las necesidades de comunicación entre los diferentes elementos que conforman la red, y crece también la complejidad de los dispositivos, por tal motivo el *software* junto con los sistemas de comunicaciones se están convirtiendo en los elementos más importantes a la hora de diseñar *CPS*, ya que deben ser fiables, seguros, eficientes y con procesamiento en tiempo real [178]. Por lo tanto, es necesario plantear nuevos dispositivos de comunicación con interfaces y protocolos de comunicación estandarizados, que permitan la interoperabilidad entre dispositivos de diferentes fabricantes. Estos nuevos dispositivos, denominados CPS Gateway deben permitir una interoperatividad, una integración más eficiente y fácil entre los niveles de proceso, bahía y estación de la subestación y fuera de ella, permitiendo a su vez el procesamiento en tiempo real, necesario en ciertas secciones y operaciones de *Smart Grid*.

6.3.1. Implementación

La Figura 6.3 muestra un diagrama de bloques de la arquitectura de un *CPS Gateway* que puede establecer comunicación con los diferentes elementos que se pueden encontrar en un entorno industrial. En la arquitectura se pueden identificar cuatro módulos que juntos dan soporte a la interoperabilidad, la alta disponibilidad y la operación en tiempo real (sincronización), los cuales se describen a continuación:

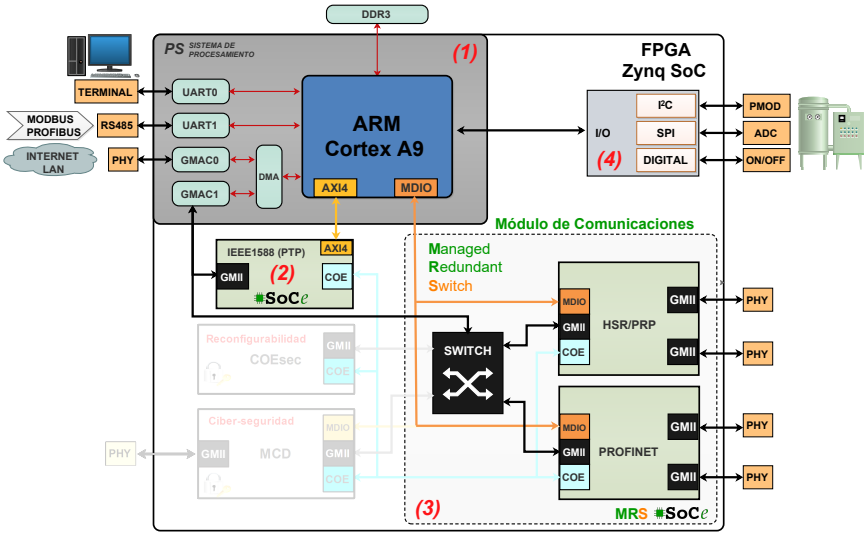


Figura 6.3: Diagrama de bloques de la implementación de la arquitectura *SoC* de un *CPS Gateway*

1. El módulo de procesamiento *(PS)*, permite ejecutar el *software* necesario para gestionar todos los componentes de la arquitectura, ejecutar librerías específicas de un protocolo de comunicaciones y realizar procesamiento y análisis de datos. El *PS* dispone de interfaces *Ethernet GMACx* y serie *UARTx* para comunicarse con el exterior o con los módulos internos *(HSR/PRP, IEEE 1588, etc.)*. En este sentido, se puede observar en la Figura 6.3 dos interfaces serie. La primera *UART0,* se utiliza como terminal para supervisión, configuración y control de la plataforma. La segunda *UART1* es utilizada para implementar comunicaciones industriales serie

(Modbus, Profibus). El *GMAC1* se utiliza para interconectar el *PS* con los *IP* que se implementan en la PL. El puerto *Ethernet* de alta velocidad *GMAC0*, se utilizara como interfaz de acceso a una red local o a *Internet*, con la finalidad de dotar al sistema acceso a servicios tales como *web, ftp*, base de datos, *cloud*, entre otros.

2. El módulo *IEEE 1588*, se utiliza para soportar aplicaciones con exigentes requisitos de sincronización. Esta *IP* proporciona un mecanismo para obtener una precisión en la sincronización del tiempo en el rango de nanosegundos.

3. El módulo de comunicaciones, es un *IP core Managed Redundant Switch (MRS)* que permite gestionar comunicaciones de alta disponibilidad *HSR/ PRP* y comunicaciones industriales de bus de campo. El *IP core* soporta la última versión de los estándares de alta disponibilidad *HSR* y *PRP* en combinación con el *IP IEEE 1588-2008*. En este módulo podemos identificar cuatro puertos *Ethernet*, dos son utilizados como puertos para realizar comunicaciones industriales basadas en *Ethernet* y los otros dos para implementar los protocolos de *HSR/PRP*.

4. El módulo de entrada/salida, se utiliza para añadir mayor funcionalidad al sistema, dispone de puertos digitales de entrada/salida, así como interfaces para la lectura de sensores utilizando comunicaciones I^2C o convertidores analógicos/digitales *(A/D)*.

Para la integración y configuración de los *IP cores* en la arquitectura se utilizaron las siguientes herramientas de desarrollo *software*:

- *Vivado*: Usado para el diseño del *hardware*, interconectar las interfaces de la *PS* con los periféricos implementados en la *PL* [179].

- *Software Development Kit (SDK)*: Usado para definir el *devicetree*, *First Stage Boot Loader (FSBL)* y generar el *zynq boot image*, necesarios para ejecutar el sistema operativo *Linux* en el *SoC* [180].

- Stack *Profinet*. Librerías desarrolladas en C por la empresa port.de [181], en las que se incluye todas las funciones necesarias para gestionar todos los requerimientos de un dispositivo *Profinet PN-IO* esclavo.

- *Python*. Utilizado para programar, compilar y ejecutar las aplicaciones *Modbus* y *Profibus*.

- Librerías *Modbus*. Contienen todas las funciones necesarias para gestionar los requerimientos de una comunicación *Modbus* serie. Estas librerías se las puede descargar del repositorio de aplicaciones de *python* [182].
- Librerías *Profibus*. Contiene las funciones básicas para gestionar las comunicaciones en las capas 1, 2 y 7 del estándar *Profibus*. Como aportación personal en estas librerías, se desarrollaron nuevas funciones de Capa 1 (física), que permitan gestionar la comunicación por *RS-485* [183].

El diseño del *hardware* de la arquitectura, como se ha explicado anteriormente, se realizó en *Vivado*, los recursos de la *FPGA* necesarios para esta configuración se resumen en la tabla 6.1.

Tabla 6.1: Recursos de la *FPGA* utilizados en la implementación

Recurso	Módulo ***MRS***	Módulo ***IEEE 1588***	Diseño Completo	Disponible	Utilización (%)
Slice LUTs	42780	1408	44188	53200	83,06
Slice Registers	28493	2350	30843	106400	28,99
F7 Muxes	2231	65	2296	26600	8,63
F8 Muxes	874	31	905	13300	6,80
Block RAM	46	6	52	140	37,14

Como se puede observar, los recursos utilizados en el diseño ocupan el 83.06 % de los recursos disponibles en la *FPGA (XC7Z020)*. Por lo tanto, hay posibilidades de implementar lógica adicional que permita dar mayor funcionalidad al sistema, por ejemplo puertos de comunicaciones I^2C, *SPI, CAN*, interfaces gráficas con *HDMI, VGA*, algoritmos de cifrado, etc.

La comunicación *Modbus* y *Profibus* se realiza a través de una interfaz *RS-485* que dispone la placa de desarrollo. El transceptor de bus que se utiliza es el *ISO35*, que es un transceptor *full duplex* de baja potencia de $3,3V$. Cumple con los requisitos del protocolo serial y las especificaciones eléctricas de *RS-485*. Su tasa de transferencia de datos es de hasta $1\,Mbps$. La dirección de comunicación es controlada por el pin *RTS* de la *UART*. Cuando *RTS* esta en bajo (0), el sistema puede recibir datos desde el bus. Cuando *RTS* es alto (1), el sistema puede enviar datos al bus. En esta aplicación se utiliza una comunicación *half duplex*, el diagrama de conexión se muestra en la Figura 6.4.

Con el entorno *Xilinx SDK*, basado en Eclipse, se desarrolló el *software* necesario para cargar el *bitstream*, crear el *devicetree* y el *FSBL* que se ejecutará en la plataforma anteriormente definida. En el sistema operativo *Linux (Linaro)* se

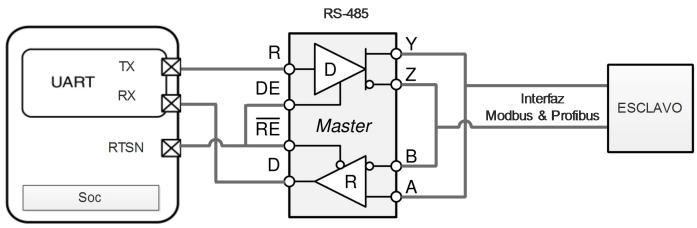


Figura 6.4: Interfaz *RS-485*: diagrama de conexión

instalaron todos los paquetes necesarios para compilar y ejecutar las aplicaciones *Profinet (esclavo), Profibus (maestro) y Modbus (maestro).*

Implementación *Modbus*

Para implementar la comunicación *Modbus* se utilizó la librería *minimalModbus* [182], que esta desarrollada en lenguaje *python*. Las comunicaciones que se pueden implementar con esta librería son: *Modbus-ASCII* y *Modbus-RTU*.

```
1  import minimalMODBUS
2  instrument = minimalMODBUS.Instrument('/dev/ttyUSB1', 1, modo)
3  instrument.serial.baudrate = 19200,
   instrument.serial.bytesize=8,
   instrument.serial.parity = PARITY_NONE,
   instrument.serial.stopbits = 1.
4  valor = instrument.read_register(núm_registro, núm_decimales)
5  instrument.write_register(núm_registro, valor, núm_decimales)
```

Figura 6.5: Uso de la librería *minimalModbus*

En la Figura 6.5 se observa el código básico para realizar una comunicación *Modbus*. A continuación se detalla cada uno de los pasos resaltados:

1. Importar librería.
2. Configuración de la interfaz, en el primer parámetro de esta función se

especifica el dispositivo que se va a utilizar. El segundo parámetro indica la dirección del esclavo y el tercer parámetro el modo de comunicación *(ASCII o RTU)*.

3. Configuración de los parámetros de transmisión, velocidad de transmisión, paridad, número de bits, etc.

4. Lectura de un registro en formato decimal, en el primer parámetro se especifica el número de registro y en el segundo se indica el numero de decimales con el que se va a representar el dato leído. Existe funciones que permiten leer los registros en otros formatos, por ejemplo enteros, hexadecimal, etc.

5. Escritura de un registro en formato decimal, el primer parámetro es el número de registro, el segundo es el valor que se va a asignar y el tercero es el número de decimales con el que se almacena el dato.

Implementación *Profibus*

Esta librería también se encuentra desarrollada en *phyton*, el proyecto de donde se adquirió estas librerías se encuentra en [183]. En este proyecto se ha desarrollado un dispositivo *Profibus DP* maestro para una tarjeta de evaluación *Raspberry Pi*. Este código ha sido adaptado para que las funciones sean compatibles con la interfaz serial *RS-232* de la placa de desarrollo que se utilizó.

```
1 import import pyPROFIBUS
2 phy = pyPROFIBUS.CpPhy (debug=True)
3 master = pyPROFIBUS.DPM1 (phy = phy, masterAddr , debug)
4 esclavo = pyPROFIBUS.DpSlaveDesc (ID_slave, Addr, I_Addr_Size, O_Addr_Size )
5 esclavo.chkCfgTelegram.addCfgDataElement ( DpCfgDataElement(0x..) )
6 master.addSlave (esclavo)
7 master.initialize()
8 inData = master.dataExchange (da = esclavo.slaveAddr, outData = outData)
9 master.destroy()
```

Figura 6.6: Uso de la librería *pyProfibus*

En la Figura 6.6 se observa el código básico para realizar una comunicación *Profibus*. A continuación se detalla cada uno de los pasos resaltados:

1. Importar la librería.

2. Crear una interfaz de capa física, el parámetro de esta función sirve para activar la funcionalidad de depuración

3. Crear el dispositivo *Profibus* máster, en el primer parámetro asignamos la interfaz física que se explicó en el paso anterior. El segundo parámetro indica la dirección que se asigna al máster y, el último parámetro sirve para activar la opción de depuración.

4. Configuración del dispositivo esclavo con el que se establecerá la comunicación, en el primer parámetro se introduce el número de identificación del dispositivo esclavo, el segundo parámetro indica la dirección del esclavo, el tercer y cuarto parámetro se refiere a la cantidad de *bytes* que se puede direccionar como entrada y como salida respectivamente.

5. Configurar los módulos de datos del telegrama *Profibus*, dependerá de las características de cada dispositivo *Profibus*.

6. Esta función se utiliza para vincular el dispositivo esclavo al maestro *Profibus*.

7. Inicia la comunicación con la configuración establecida.

8. Función general para intercambio de datos entre el máster y el esclavo, como parámetros tenemos la dirección del esclavo y los datos a enviar, y devuelve el valor de los registros de entrada del dispositivo esclavo.

9. Función para eliminar los procesos creados.

Implementación *Profinet*

Para compilar el *stack* de *Profinet* se uso de la herramienta de desarrollo de *port.de*. Figura 6.7.

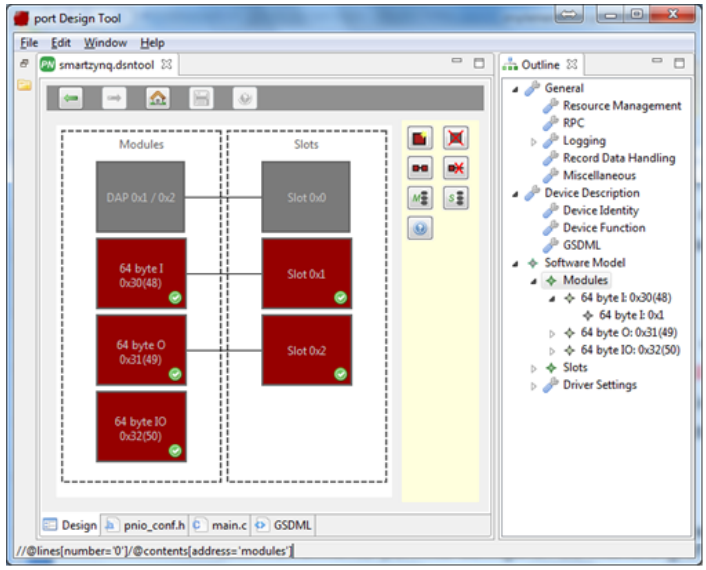


Figura 6.7: Herramienta de desarrollo *Profinet* de *Port.de*

Esta herramienta permite configurar de manera gráfica los parámetros de un dispositivo *Profinet*, y se puede especificar la cantidad de módulos y sub-módulos *I/O* que es capaz de manejar. Como salida se obtiene el archivo *"pnio_conf.h"*, en el que se especifica los parámetros con que se compilara el *stack*, también obtenemos el archivo "main.c", que no es más que una aplicación para inicializar y testear el dispositivo. Por último obtenemos el archivo *GSDML*, en el que se especifica las características del *hardware* del dispositivo *Profinet-IO*, este archivo es utilizado por las herramientas de desarrollo y gestión de proyectos de automatización, como por ejemplo *TwinCad 3* de *BeckHoff*, *STEP7* de *Siemens*, entre otros.

6.3.2. Escenarios de pruebas y resultados

Para la verificación del funcionamiento del *CPS Gateway* desarrollado, se implementaron tres entornos de pruebas con dispositivos industriales comerciales. En el primero caso se verificó la interoperabilidad *Profinet* y *Profibus* del *CPS Gateway*. Para ello se configuró al *CPS Gateway* como un dispositivo *Profinet*

esclavo y *Modbus* maestro. En el segundo caso como dispositivo *Profibus* maestro. En el tercer caso se verificó el funcionamiento en cuanto a la alta disponibilidad *HSR/PRP*, sincronización y procesamiento local de datos.

Caso 1

El entorno de pruebas que se utilizó se muestra en la Figura 6.8. El ordenador en esta caso se utiliza como dispositivo controlador *Profinet* y como terminal para ejecutar las aplicaciones. Para ello ejecuta el *software TwinCat3*. Como ya se ha indicado en la tarjeta *Smart-zynq* se implementó un sistema operativo *Linux* que dispone de la pila del protocolo *Profinet* y librerías *python* para manejo del *Modbus* serie. Los dispositivos industriales comerciales utilizados fueron el *Power Monitoring Device SENTRON PAC3100* como dispositivo esclavo *Modbus*, y el *Beckhoff BK9103 Bus Coupler* se utilizó como dispositivo *Profinet* esclavo.

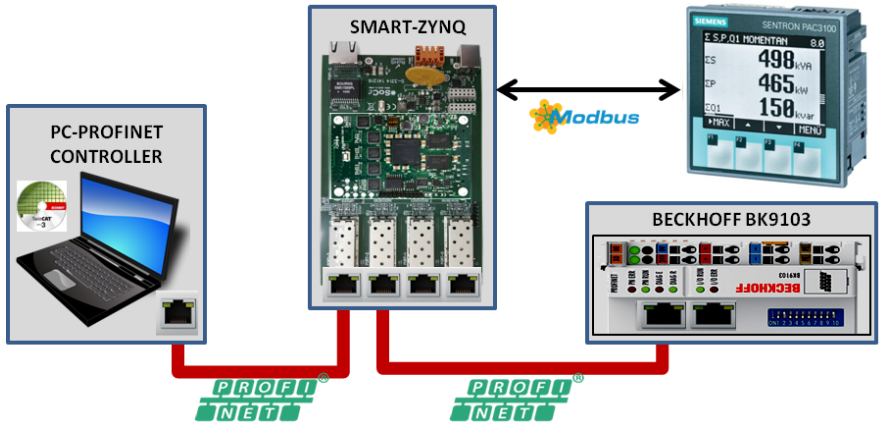


Figura 6.8: Entorno de pruebas *Profinet (esclavo)* y *Modbus (maestro)*

SENTRON PAC3100, es un dispositivo que se utiliza para medición de parámetros eléctricos, y tiene una interfaz *Modbus* serie para la transmisión de los datos.

BeckHoff BK9103 PN-IO, es un dispositivo *Profinet* esclavo, que en este caso está configurado con un modulo de entradas *KL1408* y uno de salidas *KL2408*,

esta configuración se especifica en el *software TwinCad3*.

TwinCad3, es un *software* que permite gestionar proyectos de automatización, además dispone de herramientas que permiten crear un dispositivo *Profinet* maestro utilizando las interfaces de red disponibles en el *PC*.

Resultados

Una vez configurados los equipos se realizaron pruebas básicas para verificar el funcionamiento, por ejemplo, en el caso de *Profinet* con la herramienta *TwinCad3* se asignó nombre y dirección *IP* a los dispositivos esclavos.

También se realizaron pruebas de verificación de alarmas de conexión y desconexión. La función de *blink* se ejecutó para verificar rápidamente la conexión entre el dispositivo maestro y un esclavo. Finalmente, se envían órdenes desde el maestro para modificar los valores de salida y leer las entradas. Los resultados se muestran en la Figura 6.9.

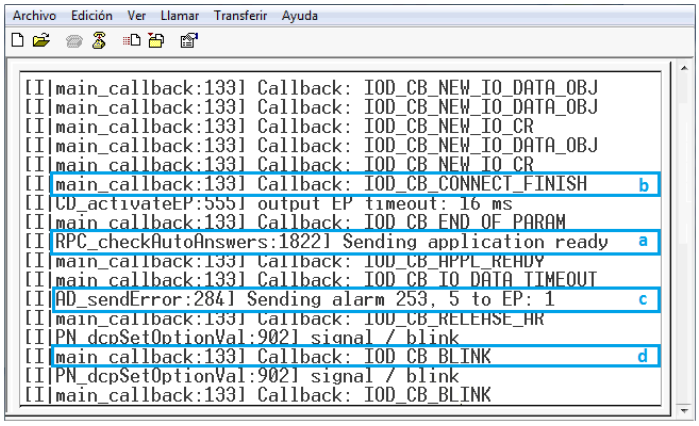


Figura 6.9: Resultados de la comunicación *Profinet*: a) Conexión. b) Desconexion. c) Alarmas . d) Utilización de la función *blink*

En el caso de *Modbus*, se enviaron tramas para la lectura y escritura de registros en formato entero y flotante. Los datos de captura de tiempo se obtuvieron del

núcleo *IP IEEE-1588*, datos de temperatura de un sensor interno de temperatura existente en la placa *SmartZynq* y parámetros eléctricos se obtuvieron del dispositivo *SENTRON PAC3100*. Los resultados se muestran en la Figura 6.10.

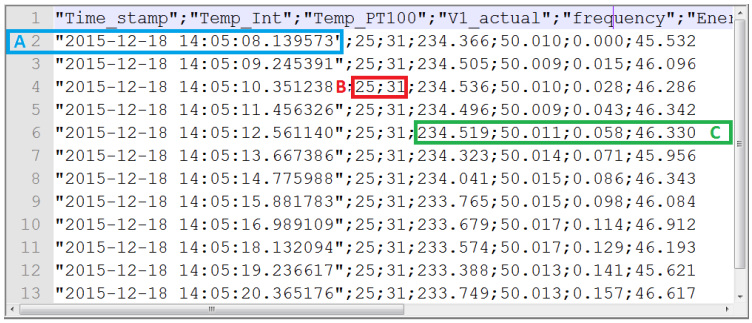


Figura 6.10: Captura de datos *Modbus*

Caso 2

El entorno de pruebas que se utilizó se muestra en la Figura 6.11. El ordenador en esta caso se lo utiliza como terminal para ejecutar las aplicaciones. En el *CPS Gateway* se ejecuta la aplicación para que incorpore la funcionalidad de *Profibus* maestro. El dispositivo industrial *Profibus* utilizado fue un controlador de motor *GE MM-200*. También se utilizó un dispositivo *Profibus* esclavo que se implementó un en un microcontrolador *ATmega32*. El objetivo de utilizar dos dispositivos *Profibus* es verificar el funcionamiento del maestro con varios esclavos.

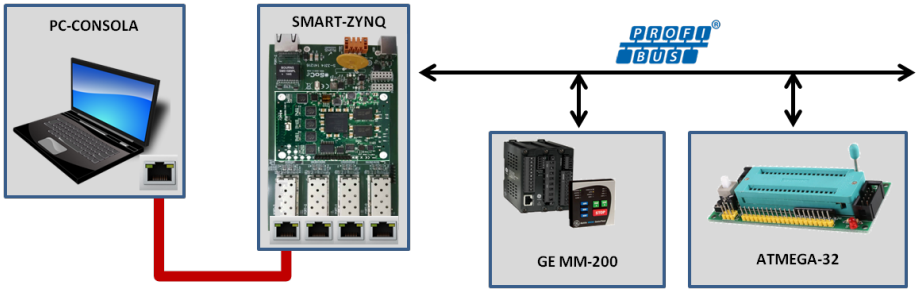


Figura 6.11: Entorno de pruebas *Profibus (maestro)*

Resultados

Para comprobar el funcionamiento se realizó pruebas de escritura y lectura en los dispositivos esclavos, y se realizaron capturas de datos para verificar que las tramas *Profibus* se están generando con el formato que establece el estándar. En la Figura 6.12 se muestran las tramas *SD1(10H)*, *SD2(68H)*, y *SC(E5H)* capturadas.

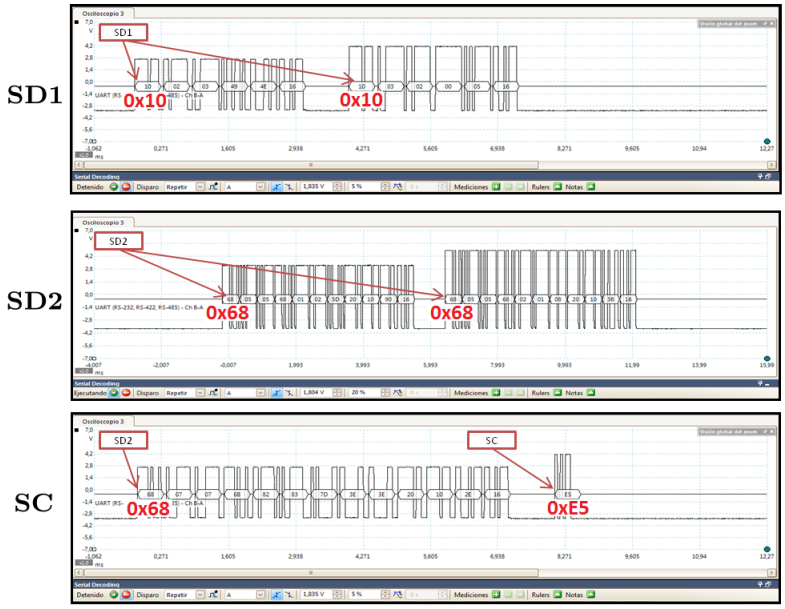


Figura 6.12: Tramas *Profibus* generadas

Caso 3

El entorno de pruebas que se utilizó se muestra en la Figura 6.13, en el esquema se pueden identificar tres *CPS Gateway* interconectados a través de una red *HSR* de 1 *Gbps* que proporciona cero segundos de retardo en caso de fallo. Para acceder de forma remota a los datos recogidos por los *CPS Gateway*, se utiliza una conexión *Ethernet* de 1 *Gbps* entre el *CPS Gateway* (3) y un ordenador. El núcleo *IP IEEE-1588* utilizado permite obtener tiempos de sincronización del orden de

20 *ns*. En [184] se describe en detalle los tiempos de sincronización obtenidos con este núcleo *IP*.

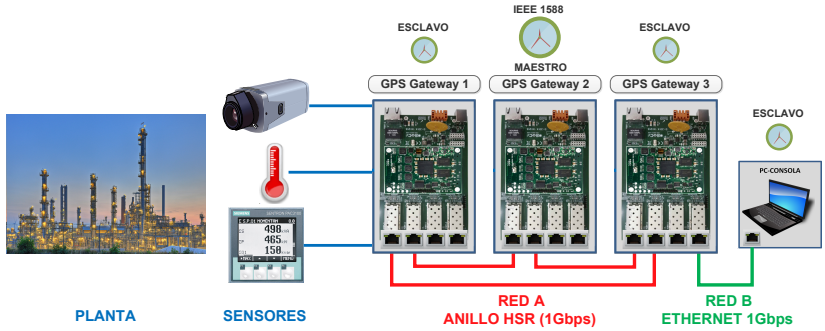


Figura 6.13: Entorno de pruebas para alta disponibilidad (*HSR*) y sincronización

Los tres nodos *CPS Gateway* conectados en la red *HSR* ejecutan las siguientes funciones:

CPS Gateway 1: Recoge los datos de los sensores a través de las interfaces *Ethernet* (vídeo) y serie *(RS-232, RS-485, I_2C)*. Los datos que se capturan en tiempo real son: a) temperatura, utilizando una *PT100* y b) parámetros eléctricos, datos de consumo de energía y frecuencia de red utilizando el dispositivo *SENTRON PAC3100*, que utiliza una interfaz *Modbus* para transmitir la información. Esta información se procesa internamente para realizar acciones de control, en este caso, control de un semáforo. Este nodo también realiza análisis de datos utilizando las librerías *python*. Esta aplicación compara el estado actual de la configuración que se está monitorizando con el modelo de la planta y estima el punto de fallo del sistema. Además, se generan informes gráficos de los resultados internamente y se puede acceder a ellos de forma remota desde el ordenador que se encuentra conectado al *CPS Gateway* 3.

CPS Gateway 2: Funciona como nodo *HSR*, se utiliza para completar el anillo *HSR* e implementar el reloj maestro *IEEE-1588*.

CPS Gateway 3: Funciona como un dispositivo *RedBox* para interconectar el anillo *HSR* (RED A) con una red *Ethernet* normal (RED B), es decir, permite transformar el ordenador en un dispositivo compatible con *HSR*.

Resultados

Una vez configurados los equipos, se realizaron pruebas básicas para verificar la operación. En el *CPS Gateway* 1 se realizó captura, procesado y almacenamiento de los datos de temperatura suministrados por la *PT100*, *timestamp* y de los parámetros eléctricos. Con el *PC* conectado al *CPS Gateway* 3 se accede de forma remota a los datos almacenados en el *CPS Gateway* 1. Esta información se observa gráficamente utilizando un navegador *web*. Los resultados se muestran en la Figura 6.14.

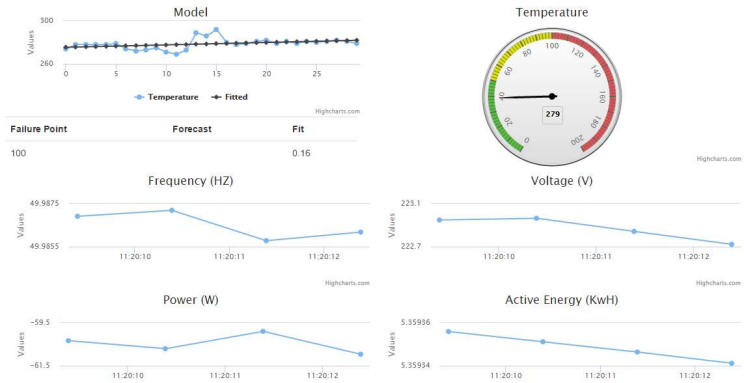


Figura 6.14: Reporte gráfico de los datos obtenidos por el *CPS Gateway* 1

6.4. Configuración Remota

El desafío de implementar un canal de comunicaciones a través de *Ethernet* para configurar remotamente un dispositivo se ha resuelto con la propuesta de un protocolo de Capa 2. Este protocolo de configuración sobre *Ethernet (Configuration Over Ethernet, COE)* permite el acceso remoto a *FPGA*s que implementan *IPs* de infraestructura de red *(switching)*. El *COE* permite acceder a los registros internos de un *IP core* para la configuración y para monitorear su estado. El protocolo *COE* se puede utilizar como un enlace de comunicación entre sistemas situados en diferentes ubicaciones físicas.

Para permitir el uso de este protocolo en canales no seguros, es necesario utilizar métodos de protección para los datos que se transmiten a través de la red.

En este sentido se ha desarrollado una versión segura del protocolo *COE* llamada *Configuration Over Ethernet Secure (COEsec)*. *COEsec* para proteger los mensajes *COE* se basa en ciertas características que utiliza la seguridad *MAC*, comúnmente llamada *MACsec*. *MACsec* usa un esquema de seguridad definido en el estándar *IEEE 802.1AE* para proteger las comunicaciones entre los equipos que están conectados a la misma red *LAN*.

6.4.1. Definición del protocolo de configuración COEsec

Con el objetivo de proporcionar la información necesaria para asegurar las comunicaciones, se ha desarrollado un formato de trama *Ethernet* personalizado que contienen varios campos específicos dentro del *payload* que proporcionan toda la información necesaria para descifrar los datos de configuración de los *IPs*. La Figura 6.15 muestra una trama *COEsec* que incluye campos específicos del protocolo y campos generales de una trama *Ethernet*.

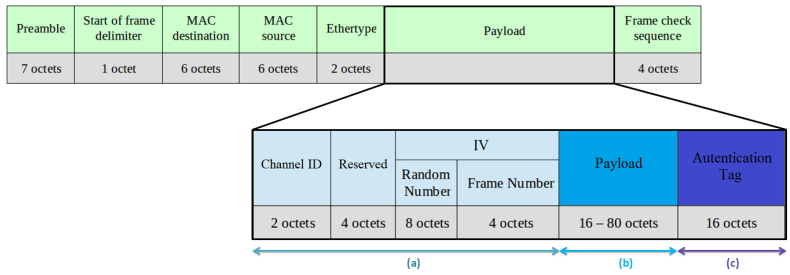


Figura 6.15: Formato de una trama *COEsec*

Como se muestra en la Figura 6.15, los datos de una trama *COEsec* se dividen en tres secciones principales. a) El primero contiene los datos necesarios para las tareas de cifrado y descifrado *(Channel ID y IV)*. b) El segundo tiene los datos de configuración encriptados. c) Por último, una etiqueta proporciona autenticación e integridad a los mensajes. Los campos que forman la trama *COEsec* son:

- *Channel ID*: cada comunicación *COEsec* se identifica por un ID de canal. Este campo se utiliza para asociar la pareja emisor/receptor a una clave secreta que se utiliza para cifrar, descifrar y autenticar los mensajes. Tiene una longitud de dos *bytes*. Este campo se almacena en un registro de la *FPGA*.

- *Reserved*: es un campo de cuatro *bytes* de longitud reservado para uso futuro. Todos ellos deben ponerse a cero.

- *IV*: es un vector de inicialización de doce *bytes* que se utiliza para el cifrado y descifrado de los mensajes y es necesario para proporcionar aleatoriedad a los mensajes y garantizar su confidencialidad. Se divide en dos subcampos: un número aleatorio y un número de trama.

  - *Random Number*: es un número aleatorio de ocho *bytes* de longitud que permite al algoritmo de encriptación garantizar la seguridad y evita que un atacante pueda deducir las relaciones entre campos de los diferentes mensajes encriptados.
  - *Frame Number*: es un número de trama de cuatro *bytes* de longitud que identifica a un par de mensajes (es decir, petición y respuesta) intercambiados entre un transmisor y un receptor y que está asociado a un determinado *Channel ID*. El transmisor debe aumentar el número de tramas por cada nueva trama enviada. En caso de que el transmisor utilice un número de trama no válido, en el receptor se generará un mensaje de respuesta automático, la respuesta contendrá el mismo número de trama que la petición recibida y el *payload* contendrá el número de trama válido que se espera recibir para ese *Channel ID*. Cuando se alcanza el valor máximo del número de trama, el valor se debe poner a cero.

- *Payload*: contiene los datos habitualmente comandos de configuración, encriptados tanto de lectura como de escritura. Hay dos tipos de comandos *COE*: lectura y escritura de registros internos de IPs.

  Para leer un registro es necesario indicar la dirección asociada al registro que se desea leer y un identificador de la operación. El comando sería el siguiente:

  $$R(x)$$

  donde "x" representa la dirección de memoria a leer (16 bits) y R es el identificador del comando *(R - Read)*.

  Para escribir un registro es necesario indicar la dirección asociada al registro que se va a escribir, los datos que se van a escribir en el registro y un identificador de la operación. El comando sería el siguiente:

  $$W(x : d)$$

donde "x" representa la dirección de memoria a escribir (16 bits), "d" es el dato a escribir (32 bits) y W es el identificador del comando *(W - Write)*.

- *Authentication Tag*: para evitar que un atacante modifique los mensajes, se utiliza una etiqueta de autenticación de 16 *bytes* de longitud. Esta etiqueta ha sido generada usando una función *Hash* que asegura que si alguno de los bits de los campos autenticados cambia, la etiqueta tomará un valor diferente. Los campos autentificados son: *Channel ID, Reserved, IV* y el *payload*, así como la cabecera *Ethernet*. La cabecera Ethernet está compuesta por *MAC* destino, *MAC* origen y *Ethertype*.

6.4.2. Implementación

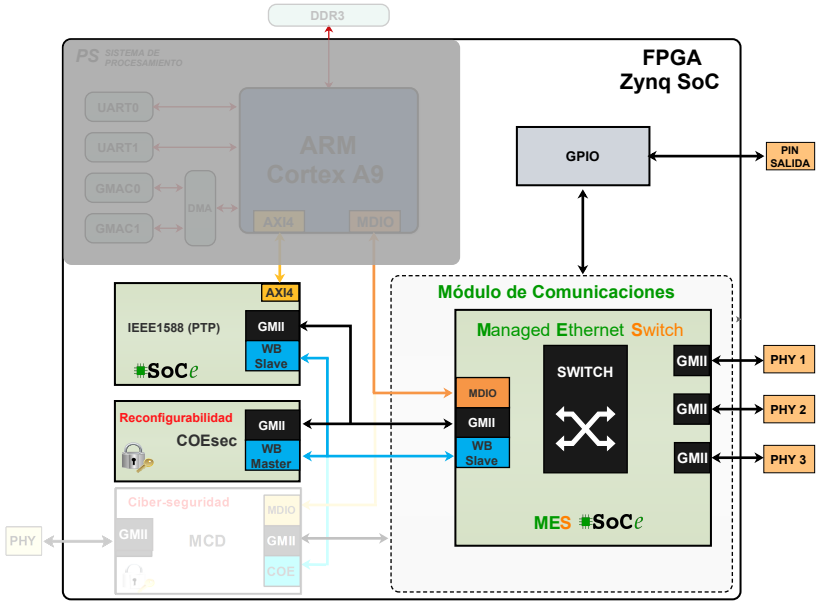


Figura 6.16: Arquitectura para medir la latencia del *IP COEsec*

Con el fin de verificar el funcionamiento del *IP Core COEsec* y el protocolo de comunicaciones planteado en la sección 5.2.4, se implementa la arquitectura presentada en la Figura 6.16. Adicionalmente esta arquitectura permite determinar la latencia que el método de cifrado *AES-GCM* introduce en el diseño.

La arquitectura mostrada en la Figura 6.16 se compone de un *IP Managed Ethernet Switch (MES)* de cuatro puertos compatibles con *IEEE1588* [185], un *IP IEEE1588* y el *IP COEsec*. El IP *COEsec* accede a los registros de los *IP* a través del bus de interconexión *Wishbone* [168].

Los *IP* implementados en la *FPGA* pueden configurarse a través de cualquier puerto *Ethernet*, ya que todas las tramas de configuración están dirigidas al *IP COEsec*. En el *IP MES*, un bit de un registro interno de configuración se conecta a un pin externo de salida, salida_led [0] en la Figura 6.17, con el objetivo de identificar el momento en el que finaliza la ejecución del comando *COE*.

Para verificar el correcto funcionamiento, se simuló el diseño propuesto en el *software Vivado*. En la Figura 6.17 se muestra una captura de pantalla de la simulación del módulo *COEsec*.

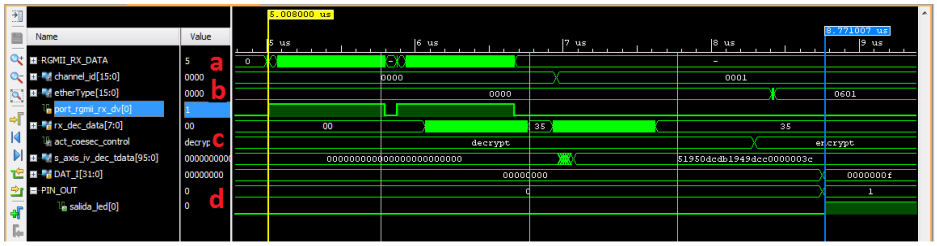


Figura 6.17: Simulación del módulo *COEsec*, trama de escritura

Las simulaciones se utilizaron para determinar el tiempo de procesamiento de una trama *COEsec* desde que ingresó en el *core* hasta que los datos se guardaron en el registro del *IP core* a ser configurado. Para ello, se realizan simulaciones de tramas de escritura y lectura con diferentes velocidades de transmisión, cuyos resultados se resumen en la Tabla 6.2.

Tabla 6.2: *COEsec* tiempo de procesamiento

Tipo	10*Mbps*	100*Mbps*	1*Gbps*
Lectura	$70,860\,\mu s$	$9,296\,\mu s$	$3,156\,\mu s$
Escritura	$78,196\,\mu s$	$10,152\,\mu s$	$3,364\,\mu s$

6.4.3. Escenarios de pruebas y resultados

La arquitectura descrita en el apartado anterior se valida en una prueba de concepto utilizando la configuración representada en la Figura 6.18. Se compone de un *PC*, una *CPS Gateway* y un osciloscopio multipuerto. En este escenario, el *PC* se encarga de enviar y recibir las tramas *COEsec*, que se generan mediante un *script* de *Python*.

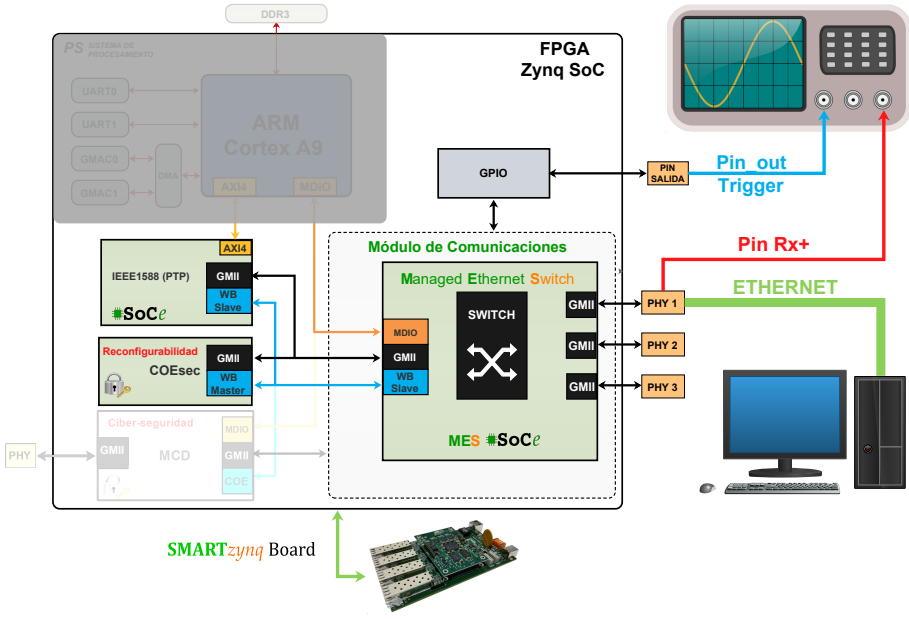


Figura 6.18: Entorno de pruebas para el módulo *COEsec*

La verificación del módulo *COEsec* se realiza de la siguiente manera:

En primer lugar, desde el *PC* se envía una trama de escritura cifrada a un registro interno del *IP MES*. A continuación, se envía una trama de lectura al registro anterior. El comando de lectura activa una respuesta desde el *IP COEsec*, la trama de respuesta cifrada se captura a través del *software WireShark*. Finalmente, la trama capturada se descifra usando un *script* de *Python* para verificar que el valor del registro coincide con el enviado en la trama de escritura.

Resultados

Para medir el tiempo de procesamiento de una trama se incorporó un osciloscopio en la configuración representada en la Figura 6.18. El primer canal del osciloscopio adquiere la trama *Ethernet* antes de ingresar al *PHY* mediante la conexión de la sonda a la linea *pin Rx+*. El segundo canal se conecta al pin de la *FPGA* vinculado al estado del registro que se está configurando. Además, el segundo canal se utiliza como señal de disparo.

La Figura 6.19 muestra la captura de una trama de escritura a una velocidad de 10 $Mbps$. Como se puede ver, el tiempo para procesar una trama escritura desde que entra en el *PHY* hasta que se produce un cambio en el pin de salida es 80,520 μs.

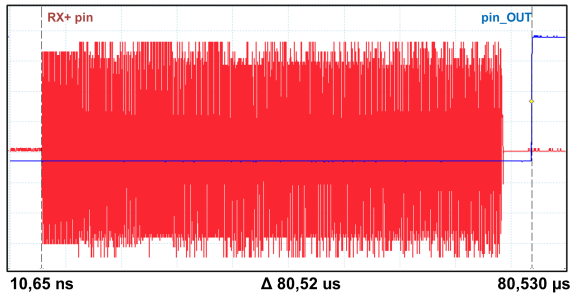


Figura 6.19: Captura de una trama (160 *bytes*) de escritura a 10 $Mbps$

Por otra parte, la diferencia entre el tiempo obtenido en la simulación de 78,196 μs y el tiempo obtenido mediante el osciloscopio es producto del retardo que se genera en el *PHY* desde el ingreso de la trama hasta que *RX+* es activado. En este caso, el retardo es 2,324 μs, que se aproxima a 2,18 μs según el valor especificado en la hoja de datos del *PHY* utilizado [186], valores resumidos en la Tabla 6.3.

Tabla 6.3: Latencia PHY en recepción

Enlace	Parámetro	Valor	Unidad
1000 BASE-T	Start of Packet to RX_CTL Asserted	236	*ns*
100 BASE-T		357	*ns*
10 BASE-T		2,18	*ms*

Por otra parte, para medir el tiempo de procesamiento en tramas transmitidas a velocidades de 100 $Mbps$ y 1 $Gbps$, el primer canal del osciloscopio se conecta al pin *LED_RX* del *PHY*, el cual indica cuando se recibe una trama.

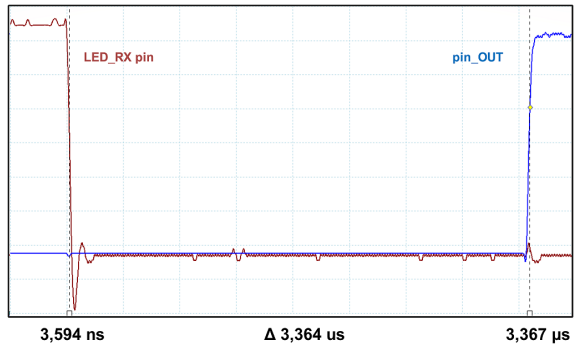


Figura 6.20: Captura de una trama (160 *bytes*) de escritura a $1Gbps$

La Figura 6.20 muestra la captura para una velocidad de 1 $Gbps$, obteniéndose un tiempo de procesamiento de 3,364 μs, que es similar a los 3,756 μs obtenidos en la simulación. Hay que tener en cuenta que para obtener el tiempo total de procesamiento, es necesario considerar la latencia introducida por el *PHY*.

Tabla 6.4: Comparativa entre los datos de simulación vs experimentales (trama de escritura)

Velocidad	**Simulación + retardo-PHY** (μs)	**Experimental** (μs)	**Error** (%)
10 $Mbps$	80,376	80,520	0,18
100 $Mbps$	10,509	10,650	1,34
1 $Gbps$	3,600	3,364	0,47

Finalmente, en la Tabla 6.4 se presentan los resultados obtenidos en simulación y experimentalmente. Como se puede ver, los resultados experimentales se ajustan al comportamiento de las simulaciones. Además, la similitud de los datos presentados en la Tabla 6.4 demuestra el correcto funcionamiento y la fiabilidad del método de medición utilizado.

6.5. *Estándar IEC 61850 y Ciber-seguridad*

En la actualidad, la seguridad informática en las infraestructuras críticas es cada vez más importante. La mayoría de los gobiernos han puesto en marcha programas específicos de seguridad para definir métodos de protección de los datos que se utilizan para gestionar la operación de la red eléctrica, y algunos ataques recientes como la sufrida en la red eléctrica de Ucrania enfatizan esta necesidad de seguridad. Con estas premisas, en estos sistemas críticos es necesario definir un enfoque de ciber-seguridad multicapa, que abarca desde los dispositivos electrónicos del nivel de proceso hasta los complejos sistemas de control que se implementan en la nube.

Las pruebas realizadas en esta sección propone el uso de cifrado con autenticación *AES-GCM* a nivel de Capa 2, para ofrecer seguridad a las tramas *Sample Values (SV)* y *Generic Object Oriented Substation Events (GOOSE)*. Este proceso de cifrado debe ejecutarse en el menor tiempo posible para que cumpla con las restricciones de tiempo que establece el estándar *IEC 61850*. La implementación propuesta basada en *FPGA* permite ejecutar un algoritmo totalmente implementado en *hardware*, para procesar, cifrar y descifrar las tramas *Ethernet*. El *hardware* propuesto es aplicable en cualquier campo, como la *Smart Grid* o la *Industria 4.0*, donde la seguridad es crítica y los datos de control se transmiten sobre enlaces *Ethernet* utilizando protocolos de comunicaciones de Capa 2.

Teniendo en cuenta el contexto de los *SAS*, las aplicaciones con estrictos requisitos de temporización utilizan los mensajes *GOOSE* y *SV*, en los que los datos se correlacionan directamente con la capa de enlace de datos *Ethernet* (Capa 2), con la finalidad de facilitar un procesamiento ágil. Un ejemplo destacable es un mensaje de disparo *GOOSE* de la clase *P2/3* que necesita un tiempo de respuesta menor a $3\,ms$ entre aplicaciones de dos *IEDs* diferentes. Para los mensajes *SV*, además del requisito de tiempo de respuesta, también se generan grandes volúmenes de datos. Por ejemplo, a una frecuencia de red de $60\,Hz$, con 80 muestras por ciclo y con un tamaño de trama estándar de 180 *bytes*, es necesario un ancho de banda de $7\,Mbit/s$ para cada *stream*.

6.5.1. Implementación

Para verificar los tiempos de procesamiento obtenidos con el modelo y las simulaciones realizadas en *Vivado*, se ha validado experimentalmente la arquitectura

del *MCD* propuesta en el apartado anterior. La configuración se muestra en la Figura 6.21.

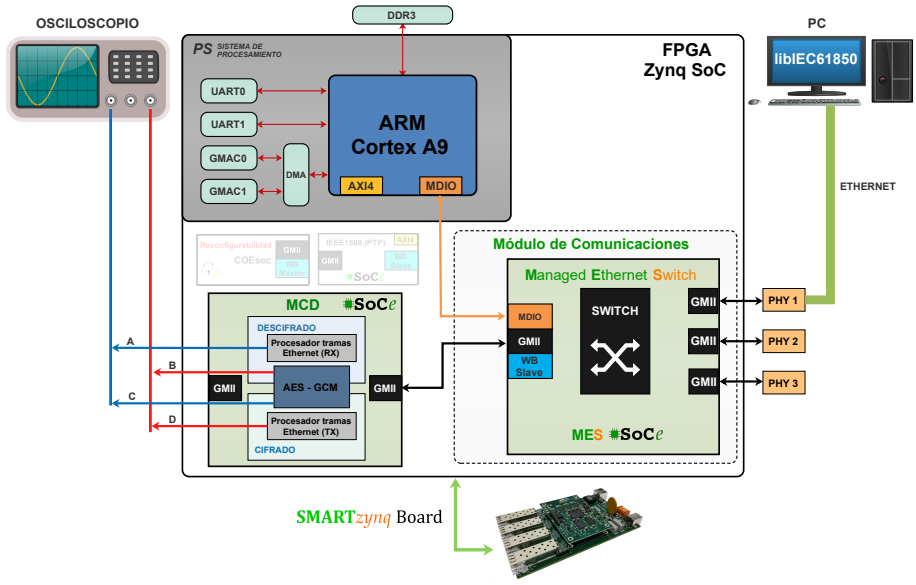


Figura 6.21: Arquitectura para medir el tiempo de procesamiento de *MCD*

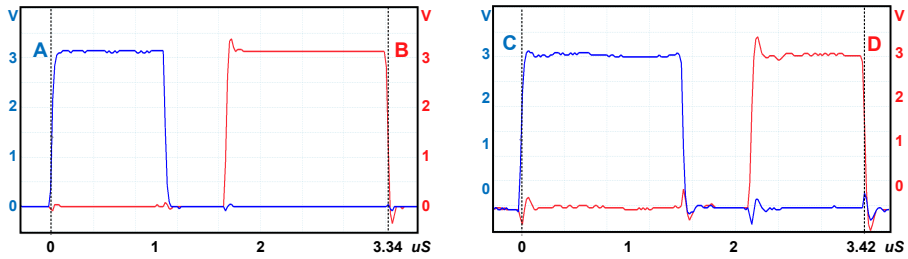
Como se mencionó en la sección anterior, el *hardware* que se utilizó es la tarjeta *SmartZynq*, en la que se incluyó el *IP core* del *MCD* y un *switch IP MES*, que es el que recibirá las tramas *Ethernet* cifradas. El ordenador es el encargado de enviar las tramas *Ethernet GOOSE* y *SV*, para ello se utilizó la librería *libiec61850* [187]. El proyecto *libiec61850* proporciona una serie de funciones para implementar servidores y clientes para los protocolos de comunicación *IEC 61850/MMS*, *IEC 61850/GOOSE* e *IEC 61850-9-2/SV*. El objetivo de este proyecto es proporcionar una implementación del estándar *IEC 61850* que sea portable y que pueda ejecutarse en sistemas embebidos y microcontroladores. También proporciona una serie de ejemplos básicos que pueden utilizarse como punto de partida para aplicaciones específicas.

Para realizar las mediciones del tiempo de procesamiento se agregó al *MCD* cuatro puntos de prueba identificados en la Figura 6.21 como *A, B, C y D,* que se conectaron a pines de la *FPGA*. Para el caso de cifrado de las tramas se utilizan los pines *A* y *B*. *A* indica cuando se ha recibido una trama sin cifrado y *B* indica

que el proceso de cifrado ha terminado. Para el caso de descifrado de las tramas se utilizan los pines *C* y *D*. *C* indica cuando se ha recibido una trama cifrada y *D* indica que el proceso de descifrado ha terminado. Para las mediciones se utilizó un osciloscopio multicanal. En el primer canal se conecta al pin que indica el ingreso de la trama *Ethernet (A o C)*, el segundo canal está conectado al pin *(B, D)* que indica que el proceso de cifrado o descifrado ha terminado. Además, el primer canal es utilizado como señal de disparo.

Resultados

En la Figura 6.22 se muestra las capturas de una trama *GOOSE* cifrada y sin cifrar de un tamaño de 160 *bytes* a una velocidad de 1 *Gbps*. El tiempo que tarda desde que una trama *GOOSE* sin cifrar entra al *MCD*, hasta que termina el proceso de cifrado es de 3,34 μs. Para una trama cifrada el proceso de descifrado tarda 3,42 μs.



(a) Cifrado **(b) Descifrado**

Figura 6.22: Tiempo de procesamiento de tramas *GOOSE* en el *MCD*

Finalmente, en la Tabla 6.5 se resumen los datos de tiempo de procesamiento obtenidos en las simulaciones basadas en el modelado, simulaciones del *MCD* en lenguaje de descripción de *hardware VHDL* y las tomadas experimentalmente con el osciloscopio. Los datos que se muestran en la Tabla 6.5 corresponden al tiempo que el *MCD* necesita para cifrar y descifrar una trama *GOOSE* de un tamaño de 160 *bytes* a una velocidad de transferencia de 1 *Gbps*.

Considerando la similitud de los datos, se concluye que el modelo inicial realizado en *VisualSim* y el método de medición utilizado, es fiable y puede ser utilizado en futuras implementaciones.

Tabla 6.5: Tiempos de procesamiento obtenidos en las simulaciones vs experimental

Trama *GOOSE* (160 *bytes*)	(1)	(2)	(3)	Error (%)	
	Modelo (μs)	VHDL (μs)	Experimental (μs)	(1)vs(3)	(2)vs(3)
Cifrado	3,87	3,38	3,34	14,49	1,19
Descifrado	3,95	3,44	3,42	14,82	0,58

6.5.2. Escenario de pruebas y resultados

La arquitectura del *MCD* propuesta en la sección anterior se ha validado en una prueba de concepto, la configuración se muestra en la Figura 6.23.

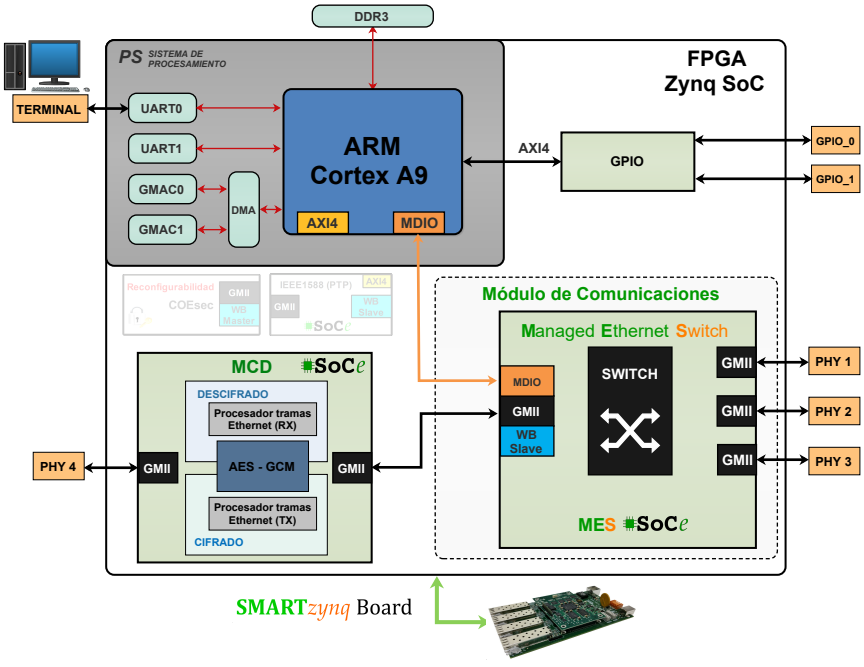


Figura 6.23: Arquitectura *CPS Gateway* para medir el tiempo de procesamiento de tramas *GOOSE*

Como se mencionó en la sección anterior el *hardware* que se utilizó es la tarjeta *SmartZynq*, en la que se incluyó el *IP MCD* y un *IP MES*. También se añadió al diseño un bloque *General Purpose Input Output (GPIO)* en el que se configuró un pin como entrada *(GPIO_0)* y otro como salida *(GPIO_1)*.

La configuración de prueba de concepto utilizada se muestra en la Figura 6.24. En el esquema se pueden identificar dos *CPS Gateway* interconectados a través de un enlace *Ethernet* de 1 *Gbps*. En los *CPS Gateway* se instaló el sistema operativo *Linux* y la librería *libiec61850*. En el *CPS Gateway* 1 se implemento las funcionalidades de un dispositivo *GOOSE Publisher*, y en el *CPS Gateway* 2 un dispositivo *GOOSE Subscriber*. El ordenador se utiliza como terminal para ejecutar las aplicaciones en las *CPS Gateway*.

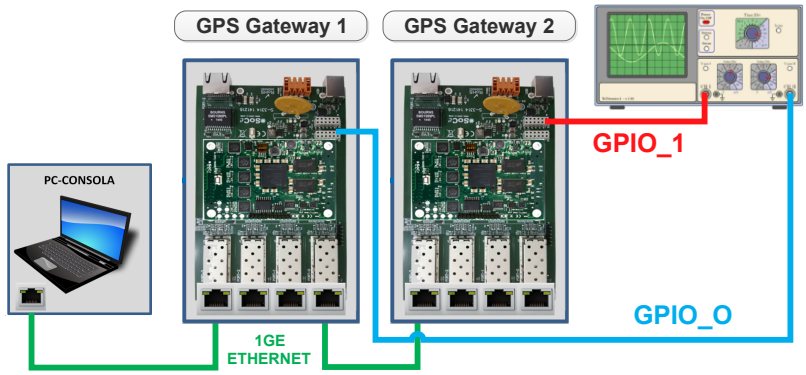


Figura 6.24: Entorno de pruebas para el descifrado de tramas *Ethernet (GOOSE)*

Resultados

Para validar los tiempos de procesamiento, se realizaron mediciones con un osciloscopio. El primer canal se conecta al *GPIO_0* del *CPS Gateway* 1. Además, el primer canal se utiliza como señal de disparo. El segundo canal está conectado al *GPIO_1* de *CPS Gateway* 2. El objetivo de este esquema es medir el tiempo de procesamiento desde el momento en que se activa el pin *GPIO_0* del *CPS Gateway* 1 hasta que su estado se refleja en el pin *GPIO_1* del *CPS Gateway* 2. Para ello, *CPS Gateway* 1 dispone de un programa mínimo *(GOOSE Publisher)* que

lee el estado del pin *GPIO_0* y que genera una trama *Ethernet GOOSE* con un tipo específico de *Ethertype*. El *MCD* lo identifica y realiza el proceso de cifrado en *CPS Gateway* 1. El *MCD* implementado en el *CPS Gateway* 2 detectará la trama en base a su *Ethertype*, y realizará la funcionalidad de descifrado y autenticación de trama. El *CPS Gateway* 2, *GOOSE Subscriber*, lee la trama *Ethernet GOOSE*, extrae la información de estado del *GPIO_0* del *CPS Gateway* 1 y lo replica en el *GPIO_1* local.

Los tiempos de ejecución de las aplicaciones implementadas son variables porque el sistema operativo *Linux* no garantiza una ejecución determinista. En este sentido, con el fin de determinar la estabilidad de las aplicaciones, se utilizó un osciloscopio en modo persistente. La Figura 6.25 muestra la captura del tiempo de procesamiento desde el momento en que se activa el pin *GPIO_0*, se realizan los procesos de cifrado en el *CPS Gateway* 1 y descifrado en el *CPS Gateway* 2), hasta que su estado se refleja en el pin *GPIO_1* del *CPS Gateway* 2.

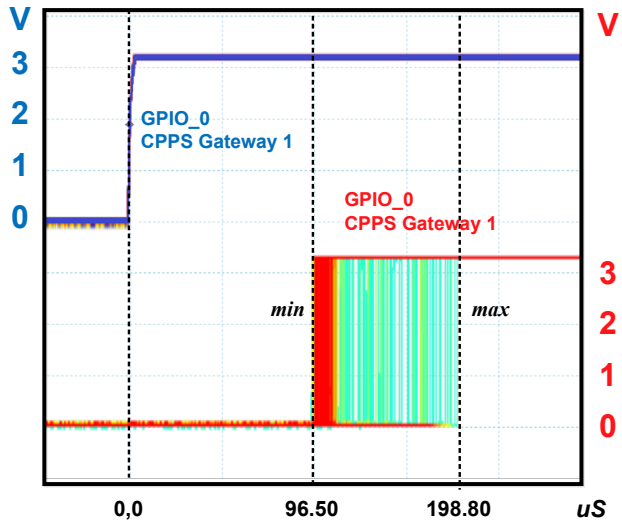


Figura 6.25: Tiempo de procesamiento de las aplicaciones *GOOSE Publisher Subscriber*

El tiempo total desde el cambio de estado en el pin *GPIO_0 del CPS Gateway 1* hasta el cambio en el pin *GPIO_1 del CPS Gateway 2* está en un rango definido por un tiempo mínimo de $96,5\,\mu s$ y un máximo de $198,80\,\mu s$, estos

tiempos corresponden al uso de tramas *Ethernet* de 100 *bytes* como mecanismo para transferir información entre los dos *CPS Gateway*.

Para los ensayos se generaron tramas *Ethernet* de 100 y 1200 *bytes* para cubrir una amplia gama de tramas de control utilizadas en aplicaciones industriales. En el sector eléctrico las tramas de *Sample Values (SV)* son de 160 a 180 *bytes* y las tramas de *GOOSE* son de 92 a 250 *bytes*.

La Tabla 6.6 resume los datos de tiempo de procesamiento obtenidos de la medición experimental con el osciloscopio. En (1) se detalla el tiempo de ejecución de la prueba sin cifrado, en (2) el tiempo de ejecución de la prueba con cifrado, y en (3) se indica el retardo introducido por el proceso de cifrado y autenticación *(CPS Gateway 1)* más el proceso de descifrado *(CPS Gateway 2)*. También se presentan los datos de valor mínimo, máximo, promedio ($\overline{x}$), así como los datos estadísticos de varianza (σ^2) y desviación estándar (σ).

Los tiempos de procesamiento que se han obtenido en las pruebas experimentales demuestran que es factible utilizar el de cifrado simétrico como mecanismo de seguridad para las tramas *Ethernet (GOOSE* y *SV)*, ya que están por debajo de los $3\,ms$ que recomienda el estándar *IEC 61850*.

Tabla 6.6: Tiempos de procesamiento obtenidos experimentalmente

Tramas (*bytes*)	**(1)** sin_cifrado (μs)					**(2)** con_cifrado (μs)					**(3)** (2) - (1) (μs)	
	min	max	$\overline{x}$	σ^2	σ	min	max	$\overline{x}$	σ^2	σ	Δd_min	Δd_max
100	86,21	189,00	106,80	924,73	30,41	96,50	198,80	124,93	988,70	31,44	9,80	10,29
1200	171,80	278,70	187,76	1397,31	37,38	211,00	319,40	249,34	1343,11	36,65	39,20	40,70

6.6. Resumen

En este capítulo, se han descrito las pruebas realizadas para validar experimentalmente las arquitecturas del *CPS Gateway* propuestas. La validación se ha realizado en una plataforma que integra una unidad de procesamiento y una $FPGA$ en un único circuito integrado. El *hardware* utilizado en este sentido fue la tarjeta de desarrollo *Smart-zynq*, que tiene como elemento principal de procesamiento un *All Programmable SoC* de *Xilinx*, específicamente de la familia *Xilinx Zynq-7000*. Este *AP-SoC* integra una *FPGA* de última generación de 28nm y una de un sistema de procesamiento que contiene un procesador *ARM Cortex-A9* de doble

núcleo. También se utilizó la tarjeta de expansión de periféricos *Smart-zynq carrier*, que incorpora una serie de conectores que permiten ampliar las capacidades de comunicación del sistema. Además de la plataforma *SoC*, también fue necesario utilizar algunos módulos de *hardware* y librerías de *software* desarrolladas por terceros con la finalidad de proporcionar interoperabilidad, alta disponibilidad, sincronización, reconfiguración y ciber-seguridad.

Para demostrar el funcionamiento y la aplicabilidad de las diferentes características que la arquitectura *CPS Gateway* debe cumplir para ser utilizada en entornos industriales como el eléctrico, se han implementado tres arquitecturas en la tarjeta *Smart-zynq*.

En la primera, se ha demostrado el funcionamiento del *CPS Gateway* en lo que respecta a la interoperabilidad, la alta disponibilidad y la sincronización. En la implementación que se ha propuesto, además del módulo de procesamiento, que permite ejecutar el *software* necesario para gestionar todos los componentes, se han incluido en la PL tres módulos adicionales: un módulo de comunicaciones, un módulo *IEEE 1588* para gestionar la sincronización y un módulo de entrada/salida. El módulo de comunicaciones consta de un *switch HSR/PRP* y otro *Profinet*. Se ha propuesto el uso del *switch HSR/PRP* para aumentar la robustez del sistema, alcanzando tiempos de recuperación de cero segundos a través de enlaces de fibra o cobre. El *switch Profinet* permite establecer comunicaciones con dispositivos como *PLCs* o dispositivos finales de *E/S*. El módulo *IEEE 1588* se ha utilizado para realizar la sincronización de tiempo en el rango de nanosegundos utilizando el protocolo *PTP*. El módulo de entrada/salida se utiliza para añadir más funcionalidad al sistema a través de puertos digitales de entrada/salida, así como interfaces I_2C o convertidores analógico/digitales. También es importante señalar que se han implementado los protocolos *Modbus* y *Profibus* utilizando las interfaces serie del sistema de procesamiento, con el fin de establecer la interoperabilidad con sensores y actuadores antiguos que no son compatibles con el estándar *IEC 61850*. Las pruebas realizadas con dispositivos comerciales y los resultados satisfactorios que se han obtenido demuestran que la tecnología utilizada en la implementación del *CPS Gateway* permite satisfacer las necesidades de comunicación requeridas por la *Smart Grid*.

En la segunda arquitectura, se ha propuesto el uso de un protocolo seguro y un *IP core* para configurar y monitorizar las *IPs* implementadas en la sección reconfigurable del *CPS Gateway*. La implementación del protocolo se lo hizo a nivel de Capa 2, y es gestionado totalmente en *hardware*. Ademas, la información esta cifrada utilizando potentes algoritmos criptográficos. Los resultados de la imple-

mentación demuestran la pequeña cantidad de recursos de la *FPGA* necesarios para implementar la solución. Con base en las pruebas experimentales, se ha determinado que el tiempo requerido para procesar una trama de configuración *Ethernet (160 bytes,* $1\,Gbps$*)* añadiendo el cifrado fue de $3,364\,\mu s$, estos resultados ha evidenciado que este enfoque de cifrado puede ser aplicado a otras tramas de control críticas, como las tramas *GOOSE* y *SV* utilizadas en una subestación eléctrica, que tienen que ser ejecutadas en los $3\,ms$ recomendados por el estándar *IEC 61850*.

Finalmente, la implementación de la tercera arquitectura ha permitido desmostar que es posible procesar las tramas *Ethernet GOOSE y SV* dentro de los $3\,ms$ que exige el estándar *IEC 61850* pero añadiendo un mecanismo de cifrado simétrico. En esta arquitectura se ha utilizado un Módulo de Cifrado/Descifrado *(MCD)* que tiene la capacidad de identificar las tramas *GOOSE* y *SV*, extraer los datos para cifrar/descifra y autenticar utilizando el algoritmo *AES-GCM*. Con esta arquitectura se han implementado dos dispositivos *CPS Gateway*, en el *CPS Gateway 1* se han incorporado las funcionalidades de un dispositivo *GOOSE Publisher*, y en el *CPS Gateway 2* las de un dispositivo *GOOSE Subscriber*. Los dos dispositivos se interconectaron mediante un enlace *Ethernet* de $1\,Gbps$, con la finalidad de medir el tiempo de procesamiento desde el momento en que se genera la trama *GOOSE* en el *CPS Gateway 1* hasta que es descifrada y procesada por el *CPS Gateway 2*. Con esta configuración se realizaron pruebas con tramas *GOOSE* de diferentes tamaños. Por ejemplo, para tramas de 100 *bytes*, el tiempo promedio de procesamiento que se obtuvo fue de $124,93\,\mu s$, de los cuales $10,29\,\mu s$ corresponde a la latencia que añade el *MCD*. Para tramas de 1200 *bytes*, el tiempo promedio de procesamiento que se obtuvo fue de $249,34\,\mu s$ de los cuales $40,70\,\mu s$ corresponde a la latencia que añade el *MCD*.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Introducción

Como capítulo final de esta tesis, se presentan las diferentes conclusiones derivadas de la investigación realizada, los resultados obtenidos y las principales aportaciones de la misma. Adicionalmente, se describen las publicaciones derivadas del presente trabajo.

También se mencionan algunas líneas de investigación abiertas que pueden conducir a trabajos futuros.

7.2. Conclusiones

A continuación, se resumen las principales conclusiones de la investigación realizada.

1. En el estado del arte, están diferenciados dos bloques de estudio. En el primero se analizaron aspectos generales sobre la red eléctrica actual y los estándares necesarios para el despliegue de redes de comunicaciones seguras, fiables y eficientes. En este sentido, en las subestaciones se está utilizando el estándar *IEC 61850* para garantizar la interoperabilidad entre dispositivos de diferentes fabricantes. Aunque la norma *IEC 61850* se está incorporando en todos los niveles de la subestación, cubriendo todas las necesidades de comunicación, todavía es necesario utilizar protocolos que garanticen la interoperabilidad con equipamiento antiguo, como es el caso del estándar *IEC 61158*, que define los diferentes buses de campo que se utilizan en el sector industrial. También se identificaron estándares adicionales para garantizar la sincronización en el orden de los nanosegundos *(IEEE 1588)*, redundancia en las comunicaciones *(IEC 62439-3)* y ciber-seguridad *(IEC 62351-6)*. Por último, los estándares *IEC 61970/61968/62325*, aunque no intervienen en las comunicaciones de las subestaciones, son indispensables para el intercambio de información entre participantes y operadores del mercado energético.

2. En la segunda parte del estado del arte, se analizaron aspectos generales de los *CPS*. Los *CPS* son sistemas complejos y multidisciplinares que se caracterizan por la integración de componentes que permiten la interacción con el mundo físico, sistemas de procesamiento e interfaces de comunicaciones. Como resultado de este estudio, se concluye que para que el *CPS* pueda satisfacer las necesidades operativas de las aplicaciones actuales, debe tener la capacidad de detectar y procesar variables físicas en tiempo real, integrar varias interfaces de comunicación estandarizadas que garanticen la interoperabilidad entre componentes y sistemas heterogéneos, disponer de mecanismos de ciber-seguridad que eviten la captura o inyección de información no deseada en los sistemas, ejecutar *software* que siempre pueda operar de la misma manera frente a perturbaciones externas o internas, disponer de sistemas de procesamiento en tiempo real para la ejecución de algoritmos de control distribuido y análisis de datos, entre otros.

3. Considerando las recomendaciones planteadas por los diferentes estándares de comunicaciones involucrados en la *Smart Grid*, se identificaron los requerimientos de operación de los dispositivos utilizados en las redes de comunicaciones del sector eléctrico, entre los que podemos mencionar el tiempo de transferencia, sincronización y tiempo de restablecimiento. En las partes 5, 8-1, 9-2 de la norma *IEC 61850* se identificaron cinco perfiles de comunicación que se utilizan para encapsular los distintos mensajes que se generan en una subestación, estos son: *SV, GOOSE, GSSE, PTP*

y *MMS*. Los perfiles mencionados anteriormente se agrupan en siete tipos, que van desde el tipo 1, que cubre los mensajes rápidos, hasta el tipo 7, para los mensajes de comando y control de acceso. De estos siete tipos de mensajes, los que necesitan tiempos de transferencia más reducidos ($\leq 3\,ms$) son los de tipo 1A *(GOOSE)* y tipo 4 *(SV)*, de ahí que sean los que demandan mayores requerimientos de procesamiento. El estándar *IEC 61850* no especifica ningún método de sincronización, lo que define el estándar son varias clases de rendimiento para la sincronización de *IEDs*. La clase T5 es la que exige precisión de sincronización menor a un $1\,\mu s$. En este sentido, se identifica el estándar *IEEE 1588 Precision Time Protocol (PTP)* como la más adecuada opción para ser considerada en la arquitectura del *CPS Gateway* que se propone.

4. Adicionalmente a estos requisitos establecidos para la *Smart Grid*, en el contexto de la operación en las Subestaciones Eléctricas, debe considerarse el tiempo de restablecimiento, que es el tiempo durante el cual la subestación tolera una interrupción de un servicio. El tiempo de recuperación de la red de comunicaciones debe ser menor que el tiempo de restablecimiento. Según el estándar *IEC 61850-5* para mensajes de mayor prioridad como el enclavamiento, disparo y bloqueo *(GOOSE)* se tolera un retardo inferior a $4\,ms$, por lo que surge la necesidad de implementar una red de alta disponibilidad que evite la pérdida de información en caso de fallo del enlace de comunicaciones. El estándar *IEC 61850* en la parte 90-4 establece que los protocolos de comunicaciones *RSTP, PRP* y *HSR* se utilizarán para implementar una red de alta disponibilidad. *RSTP* no proporciona una recuperación rápida, sin embargo, es lo suficientemente rápida para la mayoría de las aplicaciones que se implementan a nivel de estación. En cambio, los protocolos *HSR* y *PRP* son los únicos que garantizan un tiempo de recuperación cero segundos en caso de fallo de la red, es decir que no existirá pérdida de información.

Otros requisitos que están fuera del alcance del estándar *IEC 61850* pero que son importantes considerar en la arquitectura *CPS Gateway* son la interoperatividad y la ciber-seguridad.

5. Para garantizar la interoperabilidad, la arquitectura *CPS Gateway* tiene que integrar diversas interfaces de comunicaciones serie y *Ethernet* para dar soporte a una gran diversidad de buses de campo. Para los buses de campo *Ethernet* que precisan capacidades operativas en tiempo real, es necesario incorporar *hardware* complementario, mediante módulos externos

que integren las interfaces y la pila del protocolo o *IP cores* que se utilizarán en las implementaciones basadas en *FPGA*.

6. En el estándar *IEC 62351-6* se definen métodos para ayudar a proteger las comunicaciones *IEC 61850*. Sin embargo, los mecanismos de seguridad especificados en las versiones actuales de los estándares presentan algunas carencias. Por ejemplo, para garantizar la integridad de los datos en las tramas *GOOSE* y *SV*, el estándar propone el uso de códigos de autenticación de mensajes *(MAC)* utilizando el algoritmo computacional *SHA*. Estos mensajes son firmados digitalmente mediante *RSA* para proporcionar la autenticacíon de origen. Sin embargo, a pesar de que se emplearon costosos procesadores con aceleradores de cifrado, los tiempos de ejecución del proceso de de firma digital *RSA* excedían los tiempos de transferencia máximos de $3\,ms$ establecidos en el estándar *IEC 61850*. Por lo tanto, está previsto que el estándar *IEC 62351-6* se actualice sobre la base de los requisitos de operación definidos en la norma *IEC 61850-90-5*, en donde se propone el uso de criptografía simétrica para garantizar la seguridad de las comunicaciones de Capa 2 *(GOOSE y SV)*.

7. Teniendo en cuenta las características de los CPS y los requerimientos de operación definidos en los diferentes estándares de comunicaciones utilizados en la Smart Grid, en especial el estándar *IEC 61850*, en este trabajo se planteó una arquitectura de referencia de un *CPS* para la *Smart Grid*. Esta arquitectura permite la integración directa de los nodos en la red de comunicaciones y su procesamiento en tiempo real, necesario en ciertas secciones y operaciones de la *Smart Grid*. Incorpora mecanismos avanzados de sincronización, comunicaciones redundantes para ofrecer alta disponibilidad, compatibilidad con la infraestructura de automatización de subestaciones actualmente en fase de despliegue y para asegurar las comunicaciones en infraestructuras críticas como las *SAS* se propone como mecanismo de ciberseguridad el uso de cifrado simétrico *(AES-GCM)* a nivel de Capa 2. Otros requisitos que están fuera del alcance de los estándares pero son importantes considerar son la interoperatividad, la capacidad de reconfiguración y la posibilidad de realizar análisis local de datos.

8. Teniendo en cuenta la flexibilidad en términos de *hardware* y *software* que debe tener la arquitectura *CPS Gateway* propuesta, en este trabajo se ha considerado el uso de los dispositivos reconfigurables como las *FPGAs* y los *SoC* de última generación como plataforma para realizar la implantación de la arquitectura del *CPS Gateway*. Las *FPGAs* son dispositivos extremadamente versátiles que pueden configurarse para implementar complejos sistemas digitales. La *FPGA* ha pasado de ser una herramienta para la

creación de prototipos a ser una solución esencial para el desarrollo de dispositivos que requieren altas capacidades de procesamiento, requerimientos de operación en tiempo real, interoperabilidad, flexibilidad, seguridad y alta disponibilidad. Las actuales *FPGAs* integran secciones reconfigurables con secciones fijas multiprocesador, lo que permite implementar arquitecturas más complejas como la que se ha propuesto en este trabajo.

9. Para demostrar el funcionamiento y la aplicabilidad de las diferentes características de la arquitectura *CPS Gateway* se realizaron tres experimentos utilizando dispositivos modernos. Estos experimentos se realizaron en una variedad de configuraciones, desde experimentos básicos con equipo de laboratorio hasta experimentos utilizando dispositivos industriales reales. En la primera configuración, se ha demostrado el funcionamiento del *CPS Gateway* en lo que respecta a la interoperabilidad, la alta disponibilidad y la sincronización. Para ello se ha incluido en la sección *FPGA* del dispositivo un módulo de comunicaciones que permite establecer comunicaciones industriales *Ethernet* y *HSR/PRP* y un módulo *IEEE 1588 PTP* para realizar la sincronización de tiempo en el rango de nanosegundos. Las pruebas realizadas con dispositivos comerciales y los resultados satisfactorios que se han obtenido demuestran que la tecnología utilizada en la implementación del *CPS Gateway* permite satisfacer las necesidades de comunicación requeridas por la *Smart Grid*. En la segunda configuración, se ha demostrado el uso de un protocolo seguro y un *IP core* para configurar y monitorizar las *IPs* implementadas en la sección reconfigurable del *CPS Gateway*. El protocolo se ha implementado a nivel de Capa 2, y es gestionado totalmente en *hardware*. Ademas, la información transmitida esta cifrada y autenticada utilizando potentes algoritmos criptográficos. Finalmente, la tercera configuración ha permitido desmostar que es posible procesar las tramas *Ethernet* seguras *(GOOSE y SV)* dentro de los $3\,ms$ que exige el estándar *IEC 61850*. En esta configuración se ha utilizado un módulo de cifrado/descifrado que tiene la capacidad de identificar las tramas *GOOSE* y *SV*, extraer los datos para cifrar/descifrar y autenticar utilizando un algoritmo de cifrado simétrico.

7.3. Resumen de las principales aportaciones

En esta sección se presentan las principales aportaciones de esta tesis. El orden de aparición está determinado por la importancia de las diferentes contribuciones

en este documento.

1. **Propuesta de arquitecturas de un *Cyber Physical System Gateway***

   Debido a las necesidades de computación y a la flexibilidad a nivel de hardware que las arquitecturas requieren, en este trabajo se ha propuesto el uso de tecnología *SoC* reconfigurable para la implementación de un *CPS Gateway* para la *Smart Grid*. Para el planteamiento de las arquitecturas, primero se ha determinado el entorno en el que se utilizará el dispositivo y los requisitos de funcionamiento. En este sentido, en las arquitecturas propuestas se han considerado la implementación de una serie de medidas para abordar todos los requisitos operativos de los dispositivos para la *Smart Grid* recogidos en la sección 3.2, por ejemplo: Para garantizar el procesamiento de datos y la operación en tiempo real se propone utilizar dispositivos *SoC* reconfigurables que se integren en el mismo silicio la unidad de procesamiento y el área reconfigurable. Para establecer comunicaciones de alta disponibilidad se propone utilizar los protocolos *HSR/PRP* implementados en *hardware*. Para garantizar la interoperabilidad y establecer la comunicación con todos los niveles de la subestación y fuera de ella, se propone utilizar soluciones basadas en el estándar *IEC 61850* y buses de campo como *Profinet, Modbus, Profibus*, etc. Como mecanismo de reconfiguración remota, se propone el uso de un *IP core* y un protocolo de comunicaciones seguro implementado a nivel de Capa 2. Por último, como medida de ciberseguridad para las tramas de Capa 2 como *GOOSE* y *SV*, se propone implementar una solución basada en hardware que utiliza la criptografía de clave simétrica para cifrar/descifrar y autenticar las tramas.

2. **Evaluación experimental de las arquitecturas de un *Cyber Physical System Gateway***

   Se han realizado varios experimentos para validar las arquitecturas del *CPS Gateway*, como se ha descrito en el Capítulo 6. La validación se ha realizado en una plataforma que integra una unidad de procesamiento y una *FPGA* en un único circuito integrado. El *hardware* utilizado en este sentido fue la tarjeta de desarrollo *Smart-zynq*. Además de los experimentos de laboratorio, la arquitectura también fue validada con dispositivos industriales de uso comercial. Se han implementado tres arquitecturas para validar las diferentes funcionalidades. En la primera configuración, se ha demostrado el funcionamiento del *CPS Gateway* en lo que respecta a la interoperabilidad, la alta disponibilidad y la sincronización. En la segunda configuración,

se ha demostrado el uso de un protocolo seguro y un *IP core* para realizar la configuración remota de las *IPs* implementadas en la *PL* del *CPS Gateway.* Finalmente, en la tercera configuración se ha demostrado que es posible procesar las tramas *Ethernet GOOSE y SV* dentro de los $3\,ms$ que exige el estándar *IEC 61850* añadiendo mecanismo de ciber-seguridad.

3. **Identificación de los requisitos de operación de los dispositivo para la *Smart Grid***

   En base a la información recogida sobre la red eléctrica y los estándares de comunicaciones necesarios para implementar una red inteligente, en el capítulo 3 se han identificado y descrito los requerimientos de operación que debe cumplir un dispositivo *CPS Gateway* para poder ser utilizado en la *Smart Grid*. En este sentido, en base a los parámetros de funcionamiento definidos en la norma *IEC 61850*, un dispositivo cuando trabaja con mensajes críticos como *GOOSE* o *SV* debe garantizar un tiempo de transferencia inferior a $3\,ms$, una sincronización de tiempo en el orden de los nanosegundos y un tiempo de restablecimiento en caso de fallo de cero segundos. Otros requisitos que están fuera del alcance del estándar *IEC 61850* pero que son importantes de considerar al momento de plantear una arquitectura *CPS Gateway* son la interoperabilidad, la capacidad de reconfiguración y la ciberseguridad.

4. **Aspectos generales relativos a la *Smart Grid***

   En esta tesis se ha realizado un estudio sobre la red eléctrica moderna denominada *Smart Grid*, se identifico la diferencia entre una red tradicional y se planteó una definición de *Smart Grid*. También, se identificaron las características y las tecnologías de comunicaciones necesarias para la implementación y se ofrece una visión general de los estándares utilizados en un ecosistema *Smart Grid*, y se identificaron los estándares de comunicaciones necesarios para la implementación de una red inteligente en la que se garantice la interoperabilidad entre dispositivos *(IEC 61850)*, la seguridad *(IEC 62351)*, la alta disponibilidad *(IEC 62439-3)* y el sincronismo *(IEEE 1588)*.

5. **Aspectos generales relativos a los *Cyber Physical Systems***

   En esta tesis se ha realizado un estudio sobre los *CPS*. En el mismo se presentan algunas definiciones sobre los *CPS*, un esquema general y una descripción de sus partes. También, se han identificado algunas funcionalidades que deben tener los *CPS*, entre las que podemos mencionar: la ejecución de aplicaciones en tiempo real, el procesamiento determinista,

la gestión de redes heterogéneas, las comunicaciones de alta disponibilidad, la flexibilidad, los mecanismos de sincronización de alta precisión y la ciber-seguridad. Para diseñar e implementar *CPS* que puedan ofrecer estas nuevas funcionalidades, se necesitan conocimientos y usar tecnologías de una amplia variedad de campos. Adicionalmente, muchas de las tecnologías que se utilizan actualmente no se han desarrollado especialmente para los *CPS*, por lo que hay que adaptarlas o incluso tienen que ser desarrolladas específicamente para los sistemas ciberfísicos. Por ejemplo, hay grandes posibilidades de desarrollo en tecnologías de actuadores y sensores, redes de comunicaciones, ciber-seguridad, métodos para ofrecer fiabilidad y estabilidad, mecanismos para garantizar procesamiento en tiempo real, control distribuido, interacción humana y en el análisis complejo de datos.

Las aportaciones que se han derivado del desarrollo de estas tesis se han publicado en cuatro artículos en revistas científicas y tres en congresos.

En la publicación R1 se ha planteado el diseño, la implementación y la verificación experimental de una arquitectura de un Smart Sensor que satisfaga los requisitos operativos necesarios para el *Industrial Internet of Things (IIoT)*. Esta arquitectura aprovecha las características de las actuales *FPGA* para implementar un dispositivo que incorpora características de operación en tiempo real, capacidad de realizar análisis local de datos, interfaces de comunicación de alta disponibilidad como *HSR/PRP*, interoperatividad y ciber-security.

En la publicación R2 se ha propuesto una arquitectura de un *Cyber Physical Production System (CPPS) Gateway* que puede ser implementado en una *FPGA* integrando interfaces de comunicación de alta disponibilidad *HSR/PRP*, interfaces de comunicación *Ethernet* industrial y conexiones a sensores y actuadores a través del buses de campo serie.

En la publicación R3 se ha propuesto la implementación de un protocolo seguro y un *IP core* para la configuración y monitorización de las *IPs* implementadas en una *FPGA*. La implementación propuesta utiliza un protocolo ligero de Capa 2 totalmente implementado en hardware y esta protegido por algoritmos criptográficos potentes como *AES-GCM*. El protocolo seguro y el *IP core* que se han propuesto son aplicables en cualquier campo en el que se requiera una configuración remota a través de un enlace Ethernet de los *IP cores* de una *FPGA*.

En la publicación R4 se ha propuesto una arquitectura de un *CPPS Gateway*

que puede ser implementado en una plataforma *SoC*, que permite establecer comunicaciones industriales *Modbus* y *Profibus* con dispositivos que se pueden encontrar en un entorno industrial.

En las publicaciones C1 y C2 se han presentando la aplicación del protocolo de sincronización *IEEE1588* sobre redes *Ethernet* de alta disponibilidad con el objetivo de alcanzar una precisión en la sincronización de tiempo menor a $1\,\mu s$.

En las publicaciones C3 se ha propuesto el uso de cifrado a nivel de Capa 2, con el objetivo de ofrecer seguridad a las tramas *SV* y *GOOSE*. Los tiempos de procesamiento del algoritmo de cifrado/descifrado y autenticación *AES-GCM* obtenidos cumplen con las restricciones de tiempo que establece el estándar *IEC 61850* validando por tanto la propuesta realizada.

7.4. Publicaciones científicas en el contexto de este trabajo

En esta sección se presentan todas las publicaciones científicas que han formado parte de este trabajo. Los artículos están separados en publicaciones en revistas científicas y publicaciones en congresos.

Publicaciones en revistas científico-técnicas

R1) **M. Urbina**, T. Acosta, J. Lázaro, A. Astarloa, U. Bidarte, Smart Sensor: *"SoC architecture for the Industrial Internet of Things"*, IEEE Internet of Things Journal, 2019, DOI: 10.1109/JIOT.2019.2908264. Índice de impacto **(JCR): 5.863.**

R2) **M. Urbina**, A. Astarloa, J. Lázaro, U. Bidarte, I. Villalta, M. Rodriguez, *"Cyber-Physical Production System Gateway Based on a Programmable SoC Platform"*, IEEE Access, pp. 20408 - 20417, 2017, DOI: 10.1109/ACCESS.2017.2757048. Índice de impacto **(JCR): 3.557.**

R3) **M. Urbina**, N. Moreira, M. Rodriguez, T. Acosta, J. Lázaro, A. Astarloa, *"Secure Protocol and IP Core for Configuration of Networking Hardware*

IPs in the Smart Grid", Energies, vol. 11, pp. 1-13, 2018, DOI: 10.3390/en11 030510. Índice de impacto **(JCR): 2.676.**

R4) **M. Urbina**, A. Astarloa, J. Lázaro, T. Acosta, *"CPPS Gateway - Implementation of Modbus and Profibus on a SoC programmable platform"*, IEEE Latin America Transactions, vol. 16, pp. 335 - 341, 2018, DOI: 10.1109/TLA. 2018.8327384. Índice de impacto **(JCR): 0.502.**

Publicaciones en congresos internacionales

C1) A. Astarloa, N. Moreira, U. Bidarte, **M. Urbina**, D. Modrono, *"FPGA based nodes for sub-microsecond synchronization of cyber-physical production systems on high availability ring networks"*, International Conference on ReConFigurable Computing and *FPGA*s (ReConFig), pp. 1-6, Cancun (México), 2015, DOI: 10.1109/ReConFig.2015.7393316.

C2) A. Astarloa, N. Moreira, J. Lázaro, **M. Urbina**, A. Garcia, *"1588-aware High-Availability Cyber-Physical Production Systems"*, IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), pp. 25-30, Beijing (China), 2015, DOI: 10.1109/ISPCS.2015.7324675.

Publicaciones en congresos nacionales

C3) **M. Urbina**, M. Rodriguez, J. Lázaro, A. Astarloa, U. Bidarte. *"Uso de criptografía en Sistemas de Automatización de Subestaciones: Cifrado de tramas Sample Values y GOOSE"*, Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI), pp. 1-6, Barcelona (España),2018. URL: http://www.saaei.org/edicion18/docs/Libro_Resumenes_ SAAEI18.pdf.

En la Tabla 7.1 se relacionan las aportaciones derivadas de la presente tesis con los trabajos publicados:

Este trabajo ha sido apoyado por el Ministerio de Economía y Competitividad de España dentro del proyecto TEC2017-84011-R, ZE-2017/00022 - Proyecto

Tabla 7.1: Relación entre las aportaciones derivadas de la presente tesis con las publicaciones

N°	Aportación	Publicaciones
1	**Propuesta de una arquitectura de un *Cyber Physical System Gateway***	R1, R2, R3, R4
2	**Evaluación experimental de la arquitectura de un *Cyber Physical System Gateway***	R1, R2, R3, R4
3	**Identificación de los requisitos de operación de los dispositivo para la *Smart Grid***	R1, R2, R4
4	**Aspectos generales relativos a la *Smart Grid***	R3, C3
5	**Aspectos generales relativos a los *Cyber Physical Systems***	R2, R4, C1, C2

NEWCAUTO cofinanciado por el Gobierno Vasco y con fondos FEDER 2014-2020 y desarrollado en la Unidad de Formación e Investigación UFI11/16 de la UPV/EHU y apoyado por el Departamento de Educación del Gobierno Vasco dentro del fondo para grupos de investigación del sistema universitario vasco IT978-16 y dentro del proyecto TFactory ER-2014/0016.

7.5. Líneas de trabajo futuro

En esta sección se presentan varias líneas de investigación que propone el autor para dar continuidad al trabajo presentado en esta tesis. Estas líneas son:

- **Utilización de escenarios de prueba más complejos y reales**

  Aunque una de las aportaciones de esta tesis es precisamente la utilización de criptografía simétrica para ofrecer ciber-seguridad a las tramas *GOOSE* y *SV*, esta método sólo fue presentado teóricamente y validado experimentalmente en escenarios de laboratorio. Por lo tanto, deben realizarse pruebas utilizando equipos certificados para verificar que el mecanismo de ciber-seguridad no interfiere en el funcionamiento de la red. Además, es necesario estudiar el mecanismo para establecer y distribuir las claves para cifrar/descifrar y realizar la autenticación.

- **Utilización de dispositivos *MPSoC***

  La necesidad de incorporar un mayor número de funcionalidades *software* en los sistemas electrónicos modernos como el *CPS Gateway* que se plantea en este trabajo, ha ocasionado que el poder de procesamiento de los

microprocesadores de propósito general no sea suficiente. En este sentido, han surgido nuevas arquitecturas de circuitos integrados que integran varios procesadores, denominados *MPSoC*. Los *MPSoC* tienen procesadores de diferentes características y pueden configurarse por separado de manera que el *software* que se ejecuta en un procesador no interfiere en la ejecución de los otros. En este sentido, podría ser interesante estudiar la aplicación de estos nuevos dispositivos en la implementación del *CPS Gateway*, en donde las diferentes funcionalidades se ejecutarían por separado sin interferir cada una en el rendimiento de las otras. Como método de gestión de la plataforma *MPSoC* se puede utilizar mecanismos de procesamiento asimétrico en donde cada procesador del *MPSoC* puede ejecutar un sistema operativo o un *software* especifico por separado. Otra opción seria el uso de la virtualización en forma de hipervisor. Un hipervisor es una capa de *software* que se encarga de gestionar los recursos *hardware* de una plataforma *MPSoC*. El hipervisor gestiona el *hardware* y crea particiones en las que los sistemas operativos se ejecutan. Una arquitectura basada en el uso de un hipervisor es flexible, lo que permite añadir nuevas funcionalidades al producto, manteniendo la separación y garantizando que las funcionalidades añadidas no interfieren con el rendimiento de las funcionalidades existentes. Aunque un hipervisor se presenta como una buena opción para futuras implementaciones de una arquitectura compleja como la propuesta en esta tesis, todavía presenta algunas dificultades y limitaciones. Para garantizar el rendimiento y reducir la latencia que introduce un hipervisor, este tiene que tener la menor cantidad posible de lineas de código, por lo que en muchos de los casos no existe controladores que soporte el *hardware* seleccionado. En este caso es necesario desarrollar controladores de *hardware* personalizados que permitan compartir los recursos, optimizando el rendimiento y la robustez de los sistemas virtualizados.

Bibliografía

[1] IEEE, "Ieee smart grid vision for communications: 2030 and beyond," 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6648362

[2] M. G. Kanabar, I. Voloh, and D. McGinn, "Reviewing smart grid standards for protection, control, and monitoring applications," *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp. 1–8, 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6175811/

[3] W. Wolf, "Cyber-physical Systems," *Computer*, vol. 42, no. 3, pp. 88–89, 2009. [Online]. Available: https://www.computer.org/csdl/mags/co/2009/03/mco2009030088-abs.html

[4] R. Rajkumar, I. L. I. Lee, L. S. L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pp. 731–736, 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5523280

[5] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia CIRP*, vol. 17, pp. 9–13, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212827114003497

[6] A. Koubaa and B. Andersson, "A Vision of Cyber-Physical Internet," *Proc. of the Workshop of Real-Time Networks (RTN 2009), Satellite Workshop to (ECRTS 2009)(July 2009)*, pp. 1–6, 2009. [Online]. Available: http://recipp.ipp.pt/bitstream/10400.22/3837/1/COM_AnisKoubaa_2009_CISTER.pdf

[7] T. Skeie, S. Johannessen, and O. Holmeide, "Timeliness of real-time IP communication in switched industrial Ethernet networks," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 1, pp. 25–39, 2006. [Online]. Available: https://ieeexplore.ieee.org/document/1593599

[8] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of Cyber-Physical Systems," *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, 2011. [Online]. Available: https://ieeexplore.ieee.org/document/6096958

[9] I. E. Agiriano, I. Calvo, A. Noguero, and E. Zulueta, "Towards Middleware-Based Cooperation Topologies for the Next Generation of CPS." *iJOE*, vol. 8, no. S4, pp. 20–27, 2012. [Online]. Available: http://online-journals.org/i-joe/article/view/2273

[10] T. Pearson, "Save time and money with COTS middleware for network equipment," pp. 1–4, 2015. [Online]. Available: https://www.edn.com/Pdf/ViewPdf?contentItemId=4014331

[11] International Organization for Standardization, "ISO/IEC 19500-1: Common Object Request Broker Architecture (CORBA) - Part 1: Interfaces," no. April, 2012. [Online]. Available: https://www.iso.org/standard/53349.html

[12] M. Henning, "A new approach to object-oriented middleware," *IEEE Internet Computing*, vol. 8, no. 1, pp. 66–75, 2004. [Online]. Available: https://ieeexplore.ieee.org/document/1260706

[13] O. M. Group, "Data Distribution Service for Real-time Systems v1.2." no. January, 2007. [Online]. Available: https://www.omg.org/spec/DDS/1.2/PDF

[14] W. W. Group, "Web Services Architecture," *Group*, no. February, 2004. [Online]. Available: http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[15] F. Perez, D. Orive, M. Marcos, E. Estévez, G. Morán, and I. Calvo, "Access to process data with OPC-DA using IEC61499 service interface function blocks," *ETFA 2009 - 2009 IEEE Conference on Emerging Technologies and Factory Automation*, pp. 2–5, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/5347024

[16] a. Dabholkar and a. Gokhale, "An Approach to Middleware Specialization for Cyber Physical Systems," *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/5158836

[17] I. Calvo, I. Etxeberria-Agiriano, and A. Noguero, "Distribution middleware technologies for Cyber Physical Systems," *2012 9th International*

Conference on Remote Engineering and Virtual Instrumentation, REV 2012, pp. 12–15, 2012. [Online]. Available: https://ieeexplore.ieee.org/document/6293151

[18] I. Calvo, O. G. De Albéniz, A. Noguero, and F. Pérez, "Towards a modular and scalable design for the communications of electrical protection relays," *IECON Proceedings (Industrial Electronics Conference)*, pp. 2511–2516, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/5415221

[19] Object Management Group, "Data Distribution Service middleware," 2019. [Online]. Available: https://www.omgwiki.org/dds/

[20] G. N. Ericsson, "Cyber Security and Power System Communication - Essential Parts of a Smart Grid Infrastructure," *IEEE Transactions on Power Delivery*, vol. 25, no. 3, pp. 1501–1507, 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5452993

[21] F. Cleveland, "IEC 62351 Security Standards for the Power System Information Infrastructure," International Electrotechnical Commission, Tech. Rep., 2012. [Online]. Available: http://iectc57.ucaiug.org/wg15public/Public%20Documents/White%20Paper%20on%20Security%20Standards%20in%20IEC%20TC57.pdf

[22] "IEC 61850: Standard for the Design of Electrical Substation Automation," 2003. [Online]. Available: https://webstore.iec.ch/publication/6028

[23] "IEC/TS 62351-6 ed1.0: Power Systems Management and Associated Information Exchange - Data and Communication Security - Part 6: Security for IEC 61850," 2007. [Online]. Available: https://webstore.iec.ch/publication/6909

[24] "IEC 61850-1 ed2.0: Communication Networks and Systems for Power Utility Automation - Part 1: Introduction and Overview," 2013. [Online]. Available: https://webstore.iec.ch/publication/6007

[25] F. Hohlbaum, M. Braendle, and F. Alvarez, "Cyber security practical considerations for implementing IEC 62351," *PACWorld 2010*, 2010.

[26] S. Fuloria, R. Anderson, K. McGrath, K. Hansen, and F. Alvarez, "The Protection of Substation Communications," in *Proceedings of SCADA Security Scientific Symposium*, 2010. [Online]. Available: https://www.cl.cam.ac.uk/~rja14/Papers/S4-2010.pdf

[27] N. Hasan Ali, B. Mohd. Ali, M. A. Abdala, M. L. Othman, and F. b. Hashim, "Comparisons process-to-bay level peer-to-peer network delay in IEC 61850 substation communication systems," *Journal of Electrical Systems and Information Technology*, vol. 1, no. 3, pp. 266–275, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2314717214000440

[28] L. Andersson, K. P. Brand, C. Brunner, and W. Wimmer, "Reliability investigations for SA communication architectures based on IEC 61850," *2005 IEEE Russia Power Tech, PowerTech*, 2005.

[29] ABB, "ABB review, IEC 61850," Tech. Rep., 2010. [Online]. Available: https://library.e.abb.com/public/a56430e1e7c06fdfc12577a00043ab8b/3BSE063756_en_ABB_Review_Special_Report_IEC_61850.pdf

[30] M. Adamiak, D. Baigent, and R. Mackiewicz, "IEC 61850 Communication Networks and Systems In Substations: An Overview for Users," *Protection & Control Journal*, pp. 61–68, 2009. [Online]. Available: http://www.gedigitalenergy.com/multilin/journals/issues/spring09/iec61850.pdf

[31] DNP Users Group, "Distributed Network Protocol," 2019. [Online]. Available: https://www.dnp.org/default.aspx

[32] A. Ortega, A. A. Shinoda, and C. M. Schweitzer, "Performance analysis of smart grid communication protocol DNP3 over TCP/IP in a heterogeneous traffic environment," *2013 IEEE Colombian Conference on Communications and Computing, COLCOM 2013 - Conference Proceedings*, 2013.

[33] B. Miller and D. Rowe, "A survey SCADA of and critical infrastructure incidents," *Proceedings of the 1st Annual conference on Research in information technology - RIIT '12*, p. 51, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2380790.2380805

[34] R. Amoah, S. Camtepe, and E. Foo, "Securing DNP3 Broadcast Communications in SCADA Systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 4, pp. 1474–1485, 2016.

[35] N. Moreira, E. Molina, J. Lázaro, E. Jacob, and A. Astarloa, "Cybersecurity in substation automation systems," *Renewable and Sustainable Energy Reviews*, vol. 54, pp. 1552–1562, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032115012034

[36] Q. Huang, S. Jing, J. Li, D. Cai, J. Wu, and W. Zhen, "Smart Substation: State of Art and Future Development," *IEEE transaction*

on Power Delivery, vol. 8977, no. c, p. 8, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7536155

[37] M. Short, F. Abugchem, and M. Dawood, "Tunneling horizontal IEC 61850 traffic through audio video bridging streams for flexible microgrid control and protection," *Energies*, vol. 9, no. 3, 2016.

[38] I. Ali and M. S. Thomas, "Substation communication networks architecture," *2008 Joint International Conference on Power System Technology POWERCON and IEEE Power India Conference, POWERCON 2008*, 2008.

[39] "IEC 61850-8-1 ed2.0: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3," 2011. [Online]. Available: https://webstore.iec.ch/publication/6021

[40] "IEC 61850-9-2 ed2.0: Specific Communication Service Map- ping (SCSM) - Sampled values over ISO/IEC 8802-3," 2011. [Online]. Available: https://webstore.iec.ch/publication/6023

[41] S. Fries and R. Falk, "Security Considerations for Multicast Communication in Power Systems," *International Journal On Advances in Security*, vol. 6, no. 3 and 4, pp. 111–121, 2013.

[42] R. Schlegel, S. Obermeier, and J. Schneider, "A security evaluation of IEC 62351," *Journal of Information Security and Applications*, vol. 34, pp. 197–204, jun 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S2214212616300771

[43] V. Skoko, B. Atlagic, and N. Isakov, "Comparative realization of IEC 60870-5 industrial protocol standards," in *2014 22nd Telecommunications Forum Telfor (TELFOR)*, vol. 7. IEEE, nov 2014, pp. 987–990. [Online]. Available: http://ieeexplore.ieee.org/document/7034572/

[44] V. Medina, I. Gómez, E. Dorronzoro, D. Oviedo, S. Martín, J. Benjumea, and G. Sánchez, "IEC-60870-5 application layer for an open and Flexible Remote Unit," *IECON Proceedings (Industrial Electronics Conference)*, no. ISlE, pp. 2454–2458, 2009.

[45] ABB, "IEC 60870-5-101 Remote Communication Protocol for REC 523," Tech. Rep., 2004.

[46] J. Lobo, M. Cuenca, D. Gregor, M. Arzamendia, R. Gregor, and S. Toledo, "Design and implementation of a gateway between IEC 61850 and IEC 60870-5-101 standards for power electrical systems," *CHILECON 2015* -

2015 IEEE Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies, Proceedings of IEEE Chilecon 2015, pp. 535–541, 2016.

[47] G. Sánchez, I. Gómez, J. Luque, J. Benjumea, and O. Rivera, "Using Internet protocols to implement IEC 60870-5 telecontrol functions," *IEEE Transactions on Power Delivery*, vol. 25, no. 1, pp. 407–416, 2010.

[48] "IRIG Standard 200-04, Overview of IRIG-B time code standard," pp. 1–6, 2011. [Online]. Available: http://metis.ipfn.ist.utl.pt/@api/deki/files/184/=TN-102_IRIG-B.pdf

[49] "IRIG Standard 200-04, IRIG serial time code formats," 2004. [Online]. Available: http://www.irigb.com/pdf/wp-irig-200-04.pdf

[50] J. C. Eidson, *Measurement, control, and communication using IEEE 1588*. Springer Science & Business Media, 2006.

[51] C. Ozansoy, A. Zayegh, and A. Kalam, "Time synchronisation in a IEC 61850 based substation automation system," *2008 Australasian Universities Power Engineering Conference*, no. January, pp. 1–7, 2008.

[52] Y.-S. Li, G. Crispieri, and H. Wohlwend, "Using Network Time Protocol (NTP): Introduction and Recommended Practices," International SEMATECH Manufacturing Initiative, Austin,USA, Tech. Rep., 2006. [Online]. Available: https://www.nist.gov/publications/using-ntp-introduction-and-recommended-practices

[53] "IEEE 1588-2008 Standard for a precision clock synchronization protocol for networked measurement and control systems," pp. 1–300, jul 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4579760/

[54] H. Flatt, "Mapping of PRP / HSR Redundancy Protocols onto a Configurable FPGA / CPU Based Architecture," 2013.

[55] H. Weibel, "Tutorial on Parallel Redundancy Protocol (PRP)," 2003. [Online]. Available: https://www.zhaw.ch/storage/engineering/institute-zentren/ines/forschung-und-entwicklung/time-synchronisation/tutorial-on-prp.pdf

[56] J. T. Yu, "A practical and effective approach to implementing High Availability Seamless Redundancy (HSR)," in *2017 IEEE Conference on Dependable and Secure Computing*. IEEE, aug 2017, pp. 392–399. [Online]. Available: http://ieeexplore.ieee.org/document/8073824/

[57] H. Flatt, S. Schriegel, T. Neugarth, and J. Jasperneite, "An FPGA based HSR architecture for seamless PROFINET redundancy," in *2012 9th IEEE International Workshop on Factory Communication Systems*. IEEE, may 2012, pp. 137–140. [Online]. Available: http://ieeexplore.ieee.org/document/6242555

[58] S. Kumar, N. Das, and S. Islam, "High performance communication redundancy in a digital substation based on IEC 62439-3 with a station bus configuration," in *2015 Australasian Universities Power Engineering Conference (AUPEC)*. IEEE, sep 2015, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/7324838/

[59] H.-D. Ngo, H.-S. Yang, D.-W. Ham, J. Rhee, Y. An, J. Han, Y. Lee, and N. Lee, "An improved High-availability Seamless Redundancy (HSR) for dependable Substation Automation System," in *16th International Conference on Advanced Communication Technology*. Global IT Research Institute (GIRI), feb 2014, pp. 921–927. [Online]. Available: http://ieeexplore.ieee.org/document/6779094/

[60] A. Naumann, I. Bielchev, N. Voropai, and Z. Styczynski, "Smart grid automation using IEC 61850 and CIM standards," *Control Engineering Practice*, vol. 25, no. 1, pp. 102–111, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2013.12.001

[61] UML, "Introduction to OMG's Unified Modeling Language," 2019. [Online]. Available: http://www.uml.org/

[62] M. Uslar, M. Specht, C. Dänekas, J. Trefke, S. Rohjans, J. M. González, C. Rosinger, and R. Bleiker, *Standardization in Smart Grids*, 2013, vol. 71. [Online]. Available: https://www.springer.com/us/book/9783642349157

[63] H. Kirrmann, K. Weber, O. Kleineberg, and H. Weibel, "Seamless and low-cost redundancy for substation automation systems (high availability seamless redundancy, HSR)," in *2011 IEEE Power and Energy Society General Meeting*, vol. 2. IEEE, jul 2011, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/6038906/

[64] "Texas Instruments: Processor SDK Linux - HSR_PRP," 2018. [Online]. Available: http://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/linux/Industrial_Protocols_HSR_PRP.html

[65] ZHAW, "Institute of Embedded Systems: PRP-1 Software Stack," 2019. [Online]. Available: https://www.zhaw.

ch/en/engineering/institutes-centres/ines/products-and-services/high-availability/prp-1-software-stack/#c46687

[66] S. Kumar, N. Das, and S. Islam, "Software implementation of two seamless redundant topologies in a digital protection system based on IEC 62439-3," in *Proceedings of the 2016 Australasian Universities Power Engineering Conference, AUPEC 2016*, 2016.

[67] "Texas Instruments: Sitara Processor AM5728." [Online]. Available: http://www.ti.com/product/AM5728

[68] Z. Lin and S. Pearson, "An inside look at industrial Ethernet communication protocols," Texas Instruments, Tech. Rep., 2017. [Online]. Available: http://www.ti.com/lit/wp/spry254b/spry254b.pdf

[69] H. J. Kashyap, "Secure Dynamic Reconfiguration of FPGAs," *ACM Trans. Reconfig. Technol. Syst*, vol. 7, 2014.

[70] F. Devic, L. Torres, J. Crenne, B. Badrignans, and P. Benoît, "SecURe DPR: Secure update preventing replay attacks for dynamic partial reconfiguration," in *International Conference on Field Programmable Logic and Applications, (FPL)*, no. Cl, 2012, pp. 57–62.

[71] J. Vliegen, N. Mentcns, and I. Verbauwhede, "A single-chip solution for the secure remote configuration of FPGAs using bitstream compression," in *International Conference on Reconfigurable Computing and FPGAs, (ReConFig)*, 2013.

[72] J. Castillo, P. Huerta, and J. I. Martínez, "Secure IP downloading for SRAM FPGAs," *Microprocessors and Microsystems*, vol. 31, no. 2, pp. 77–86, 2007.

[73] K. Poulsen, "Slammer Worm Crashed Ohio Nuke Plant Network," 2003. [Online]. Available: http://www.securityfocus.com/news/6767

[74] P. Fairley, "Cybersecurity at U.S. Utilities Due For an Upgrade," *IEEE Spectrum*, 2016.

[75] "SQL Slammer worm lessons learned for consideration by the electricity sector," *North American Electric Reliability Council*, 2003.

[76] J. Liu, Y. Xiao, S. Member, S. Li, W. Liang, and C. L. P. Chen, "Cyber Security and Privacy Issues in Smart Grids," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 981–997, 2012.

[77] Helion Technology, "Helion Technology: RSA and Modular Exponentiation cores," p. 2019. [Online]. Available: https://www.heliontech.com/downloads/Helion_PB_-_ModExp_FPGA.pdf

[78] "Tiempo: TPKA - Asynchronous publickey accelerator IP." [Online]. Available: http://www.tiempo-ic.com/uploads/Docs/TPKA_Datasheet.pdf

[79] "SILEX: Public Key Cryptography IP core (BA414EP)." [Online]. Available: https://www.silexinsight.com/products/security/public-key-asymmetric/

[80] "The Athena Group, Inc: F5200B Embedded Suite B Cryptography Microprocessor IP Core." [Online]. Available: https://www.athena-group.com/f5200-embedded

[81] "SafeXcel: High-Performance Security Co-Processor 1840," 2018. [Online]. Available: http://www2.gemalto.com/library/EMB/SafeNet_Product_Brief_SafeXcel_1840.pdf

[82] "SafeXcel-3141: Reliance Series High-Performance Security System on a Chip for IPsec & SSL/TLS Acceleration." [Online]. Available: http://www2.gemalto.com/library/EMB/SafeNet_Product_Brief_SafeXcel_3141.pdf

[83] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*, 2014. [Online]. Available: https://ptolemy.berkeley.edu/books/leeseshia/releases/LeeSeshia_DigitalV2_2.pdf

[84] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 12, pp. 4242–4268, 2014. [Online]. Available: http://www.itiis.org/digital-library/manuscript/file/894/TIIS+Vol+8,+No+12-1.pdf

[85] E. A. Lee, "Computing Foundations and Practice for Cyber- Physical Systems : A Preliminary Report," Tech. Rep., 2007. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-72.pdf

[86] L. Hu, N. Xie, Z. Kuang, and K. Zhao, "Review of Cyber-Physical System Architecture," in *2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*. IEEE, 2012, pp. 25–30. [Online]. Available: http://ieeexplore.ieee.org/document/6196100

[87] H.-M. Huang, T. Tidwell, C. Gill, C. Lu, X. Gao, and S. Dyke, "Cyber-physical Systems for Real-time Hybrid Structural Testing: A Case Study," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, 2010, pp. 69–78. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1795194.1795205

[88] J. N. Carbone, S. C. Suh, A. Eroglu, and U. J. Tanik, *Applied Cyber-Physical Systems*, 2014. [Online]. Available: http://link.springer.com/10.1007/978-1-4614-7336-7

[89] Z. Song, Y. Chen, C. R. Sastry, and N. C. Tas, *Optimal Observation for Cyber-physical Systems*, 2009. [Online]. Available: http://link.springer.com/10.1007/978-1-84882-656-4

[90] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog Computing for Sustainable Smart Cities: A Survey," vol. 50, no. 3, mar 2017. [Online]. Available: http://arxiv.org/abs/1703.07079

[91] M. E. Porter and J. E. Heppelmann, "How Smart, Connected Product Are Transforming Competition," *Harvard business review*, no. November, pp. 64 – 89, 2014. [Online]. Available: https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition

[92] K. Henning, W. Wolfgang, and H. Johannes, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0," p. 82, 2013. [Online]. Available: https://www.acatech.de/wp-content/uploads/2018/03/Final_report___Industrie_4.0_accessible.pdf

[93] S. Yinbiao, K. Lee, P. Lanctot, F. Juanbin, H. Hao, B. Chow, J.-P. Desbenoit, G. Stephan, L. Hui, X. Guodong, S. Chen, D. Faulk, T. Kaiser, H. Satoh, O. Jinsong, W. Shou, Z. Yan, S. Junping, Y. Haibin, Z. Peng, L. Dong, and W. Qui, "Internet of Things: Wireless Sensor Networks," Tech. Rep. December, oct 2014. [Online]. Available: http://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf

[94] X. Yu and Y. Xue, "Smart Grids: A Cyber-Physical Systems Perspective," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1058–1070, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7433937

[95] T. Lennvall, M. Gidlund, and J. Akerberg, "Challenges when bringing IoT into industrial automation," in *2017 IEEE AFRICON*. IEEE, sep 2017, pp. 905–910. [Online]. Available: http://ieeexplore.ieee.org/document/8095602/

[96] I-scoop, "The Internet of Things (IoT) - essential IoT business guide." [Online]. Available: https://www.i-scoop.eu/internet-of-things-guide/

[97] M. H. ur Rehman, E. Ahmed, I. Yaqoob, I. A. T. Hashem, M. Imran, and S. Ahmad, "Big Data Analytics in Industrial IoT Using a Concentric Computing Model," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 37–43, feb 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8291112/

[98] CyPhERS Cyber-physical European Roadmap & Strategy, "CPS : State of the Art," Tech. Rep., 2014. [Online]. Available: http://www.cyphers.eu/sites/default/files/D5.1.pdf

[99] CyPhERS, "Cyber-Physical European Roadmap & Strategy," Tech. Rep., 2014. [Online]. Available: http://cyphers.eu/sites/default/files/d6.1+2-report.pdf

[100] J. A. Stankovic, J. W. Sturges, and J. Eisenberg, *A 21st Century Cyber-Physical Systems Education.* Washington, D.C.: National Academies Press, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/8220381

[101] Foundations for Innovation in Cyber-Physical Systems, "Strategie R&D Opportunities for 21st Century Cyber Physical Systems: Connecting Computer and information systems with the physical world," Tech. Rep., 2013. [Online]. Available: https://cps-vo.org/file/7136/download/18871

[102] National Institute of Standards and Technology, "Foundations for Innovation in Cyber-Physical Systems," p. 60, 2013. [Online]. Available: http://www.nist.gov/el/upload/CPS-WorkshopReport-1-30-13-Final.pdf

[103] S. A. Haque, S. M. Aziz, and M. Rahman, "Review of Cyber-Physical System in Healthcare," *International Journal of Distributed Sensor Networks*, pp. 1–20, 2014. [Online]. Available: http://www.hindawi.com/journals/ijdsn/2014/217415/

[104] Acatech, "Living in a networked world," Tech. Rep., 2015. [Online]. Available: http://www.cyphers.eu/sites/default/files/acatech_STUDIE_agendaCPS_eng_ANSICHT.pdf

[105] European Parliamentary Research Service, "Industry 4.0. Digitalisation for productivity and growth," Tech. Rep., 2015. [Online]. Available: http://www.europarl.europa.eu/RegData/etudes/BRIE/2015/568337/EPRS_BRI(2015)568337_EN.pdf

[106] Deloitte, "Industry 4.0. Challenges and solutions for the digital transformation and use of exponential technologies," Tech. Rep., 2015. [Online]. Available: http://www2.deloitte.com/content/dam/Deloitte/ch/Documents/manufacturing/ch-en-manufacturing-industry-4-0-24102014.pdf

[107] DLG - Expert, "Industry 4.0 - Summary Report," Tech. Rep., 2015. [Online]. Available: https://www.cenit.com/fileadmin/dam/Corporate/PDFs/2015_5_Expertenwissen_E.pdf

[108] International Electrotechnical Commission, "Factory of the future," p. 49, 2011. [Online]. Available: https://webstore.iec.ch/publication/23389

[109] K. Nawa, N. P. Chandrasiri, T. Yanagihara, T. Komori, and K. Oguchi, "Cyber physical system for vehicle application," in *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2012, pp. 135–138. [Online]. Available: https://ieeexplore.ieee.org/document/6392540

[110] L. Zhang, "Convergence of physical system and cyber system modeling methods for aviation cyber physical control system," in *IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2014, pp. 542–547. [Online]. Available: https://ieeexplore.ieee.org/document/6932714

[111] ——, "Multi-view Approach for Modeling Aerospace Cyber-physical Systems," in *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. IEEE, 2013, pp. 1319–1324. [Online]. Available: https://ieeexplore.ieee.org/document/6682242

[112] T. Kurpick, C. Pinkernell, M. Look, and B. Rumpe, "Modeling cyber-physical systems," in *Proceedings of the Modelling of the Physical World Workshop*. ACM Press, 2012, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2491617.2491619

[113] M. D. Ilic, Le Xie, and U. A. Khan, "Modeling future cyber-physical energy systems," in *EEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*. IEEE, 2008, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/4596708

[114] M. A. A. Faruque and F. Ahourai, "A model-based design of Cyber-Physical Energy Systems," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2014, pp. 97–104. [Online]. Available: https://ieeexplore.ieee.org/document/6742873

[115] S. Soegijoko, "A brief review on existing cyber-physical systems for healthcare applications and their prospective national developments," in *International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME)*. IEEE, 2013, pp. 2–2. [Online]. Available: https://ieeexplore.ieee.org/document/6698452

[116] C. Jose and P. Vicente, "Entornos de Sistemas Multiagente y Ciber-Físicos en la Ciberdefensa," *SIC MAS&T y CPS*, vol. 111, pp. 100–102, 2014. [Online]. Available: https://revistasic.es/images/pdf/111-colaboracion-multiagente.pdf

[117] B. Fleming, "Microcontroller Units in Automobiles," *IEEE Vehicular Technology Magazine*, pp. 4–8, 2011. [Online]. Available: http://ieeexplore.ieee.org/document/6004783

[118] European-Commission, "ecall: Time saved = lives saved," 2018. [Online]. Available: https://ec.europa.eu/digital-single-market/en/ecall-time-saved-lives-saved

[119] D. B. Rawat, S. Reddy, N. Sharma, B. B. Bista, and S. Shetty, "Cloud-assisted GPS-driven dynamic spectrum access in cognitive radio vehicular networks for transportation cyber physical systems," in *IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2015, pp. 1942–1947. [Online]. Available: http://ieeexplore.ieee.org/document/7127765

[120] R. Frank, H. Weitz, G. Castignani, and T. Engel, "Collaborative traffic sensing: a case study of a mobile phone based traffic management system," in *IEEE 11th Consumer Communications and Networking Conference (CCNC)*, 2014, pp. 579–584. [Online]. Available: http://ieeexplore.ieee.org/document/7056314

[121] B. Syed, A. Pal, K. Srinivasarengan, and P. Balamuralidhar, "A smart transport application of cyber-physical systems: Road surface monitoring with mobile devices," in *Sixth International Conference on Sensing Technology (ICST)*, 2012, pp. 8–12. [Online]. Available: http://ieeexplore.ieee.org/document/6461796

[122] S. Mahmud and A. Alrabady, "A new decision making algorithm for airbag control," *IEEE Transactions on Vehicular Technology*, pp. 690–697, 1995. [Online]. Available: http://ieeexplore.ieee.org/document/406638

[123] J. Wen, V. Sarihan, B. Myers, and G. Li, "Multidisciplinary Approach for Robust Package Design of MEMS Accelerometers," *IEEE Transactions on*

Components, Packaging and Manufacturing Technology, pp. 1934–1938, dec 2011. [Online]. Available: http://ieeexplore.ieee.org/document/6119119

[124] Bai Zhanyuan, Xu Aidong, and Jin Ni, "A design of the Intelligent electronic control seat belt retractor based on automotive active safety technology," in *International Conference on Computer Application and System Modeling (ICCASM)*, 2010, pp. 403–406. [Online]. Available: http://ieeexplore.ieee.org/document/5620489

[125] N. Barbour, E. Brown, J. Connelly, J. Dowdle, G. Brand, J. Nelson, and J. O'Bannon, "Micromachined inertial sensors for vehicles," in *Proceedings of Conference on Intelligent Transportation Systems*, 1997, pp. 1058–1063. [Online]. Available: http://ieeexplore.ieee.org/document/660620

[126] NITRD, "High-Confidence Medical Devices: Cyber-Physical Systems for 21st Century Health Care," Tech. Rep., 2009. [Online]. Available: https://www.nitrd.gov/About/MedDevice-FINAL1-web.pdf

[127] A. Gupta, M. Kumar, S. Hansel, and A. K. Saini, "Future of all technologies- The Cloud and Cyber Physical Systems," 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.378.7667&rep=rep1&type=pdf

[128] R. Baheti and H. Gill, "Cyber-physical systems," *The impact of control technology*, vol. 12, no. 1, pp. 161–166, 2011.

[129] E. O. Mendez and S. Ren, "Design of cyber-physical interface for automated vital signs reading in electronic medical records systems," in *IEEE International Conference on Electro/Information Technology*, 2012, pp. 1–10. [Online]. Available: http://ieeexplore.ieee.org/document/6220696

[130] S. Don and M. Dugki, "Medical Cyber Physical Systems and Bigdata Platforms," *Proceedings of the Medical Cyber Physical Systems Workshop*, 2013. [Online]. Available: https://pdfs.semanticscholar.org/da6e/dfbc894cc8ee4e4998585200573e96aa81f5.pdf

[131] J. Wang, H. Abid, S. Lee, L. Shu, and F. Xia, "A Secured Health Care Application Architecture for Cyber-Physical Systems," *Control Engineering and Applied Informatics*, pp. 101–108, dec 2011. [Online]. Available: http://arxiv.org/abs/1201.0213

[132] K. K. Venkatasubramanian, S. Nabar, S. K. S. Gupta, and R. Poovendran, "Cyber Physical Security Solutions for Pervasive Health Monitoring Systems," in *User-Driven Healthcare: Concepts, Methodologies, Tools, and Ap-*

plications, 2012, pp. 447–465. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2770-3.ch022

[133] M. Sung, C. Marci, and A. Pentland, "Wearable feedback systems for rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, no. 1, pp. 1–12, 2005. [Online]. Available: http://dx.doi.org/10.1186/1743-0003-2-17

[134] European-Commission, "Smart Cities," https://ec.europa.eu/digital-single-market/en/smart-cities, 2018.

[135] Comisión de las comunidades Europeas, "Eficiencia energética: alcanzar el objetivo del 20 %," Tech. Rep., 2008. [Online]. Available: https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:52008DC0772&from=ES

[136] Cyber-Physical European Roadmap & Strategy, "CPS Technologies," Tech. Rep., 2014. [Online]. Available: www.cyphers.eu/sites/default/files/D4.2.pdf

[137] T. Novak and A. Gerstinger, "Safety- and Security-Critical Services in Building Automation and Control Systems," *IEEE Transactions on Industrial Electronics*, pp. 3614–3621, nov 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5196777

[138] Z. Pan, S. Hariri, and Y. Al-Nashif, "Anomaly based intrusion detection for Building Automation and Control networks," in *11th International Conference on Computer Systems and Applications (AICCSA)*, 2014, pp. 72–77. [Online]. Available: http://ieeexplore.ieee.org/document/7073181

[139] W. Granzer, F. Praus, and W. Kastner, "Security in Building Automation Systems," *IEEE Transactions on Industrial Electronics*, pp. 3622–3630, nov 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5332331

[140] N. Adam, "Workshop on future directions in cyber-physical systems security," Tech. Rep., 2010. [Online]. Available: http://feihu.eng.ua.edu/NSF_CPS/year1/w2_read.pdf

[141] Energetics Incorporated, "Cyber-Physical Systems: Situation Analysis of Current Trends, Technologies, and Challenges," Tech. Rep., 2012. [Online]. Available: https://cps-vo.org/file/7139/download/18874

[142] CPS_Week, "Cyber physical systems summit," Tech. Rep., 2010. [Online]. Available: http://iccps2012.cse.wustl.edu/_doc/CPS_Summit_Report.pdf

[143] Cyber-Physical European Roadmap & Strategy, "Research Agenda and Recommendations for Action," Tech. Rep. [Online]. Available: cyphers.eu/sites/default/files/d6.1+2-report.pdf

[144] B. Roberts and J. McDowall, "Commercial successes in power storage," *IEEE Power and Energy Magazine*, pp. 24–30, mar 2005. [Online]. Available: http://ieeexplore.ieee.org/document/1405867

[145] M. A. Awadallah and B. Venkatesh, "Energy Storage in Flywheels: An Overview," *Canadian Journal of Electrical and Computer Engineering*, pp. 183–193, jan 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7120216

[146] M. I. Daoud, A. S. Abdel-Khalik, A. Massoud, S. Ahmed, and N. H. Abbasy, "On the development of flywheel storage systems for power system applications: A survey," *International Conference on Electrical Machines*, pp. 2119–2125, 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6350175

[147] B. Davidson, I. Glendenning, R. Harman, A. Hart, B. Maddock, R. Moffitt, V. Newman, T. Smith, P. Worthington, and J. Wright, "Large-scale electrical energy storage," *IEE Proceedings A Physical Science, Measurement and Instrumentation, Management and Education, Reviews*, pp. 345–385, 1980. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/ip-a-1.1980.0054

[148] L. Chen, Y. Liu, A. B. Arsoy, P. F. Ribeiro, M. Steurer, and M. R. Iravani, "Detailed Modeling of Superconducting Magnetic Energy Storage (SMES) System," *IEEE Transactions on Power Delivery*, pp. 699–710, apr 2006. [Online]. Available: http://ieeexplore.ieee.org/document/1610681

[149] V. Vongmanee, "The renewable energy applications for uninterruptible power supply based on compressed air energy storage system," in *2009 IEEE Symposium on Industrial Electronics & Applications*, vol. 2. IEEE, 2009, pp. 827–830. [Online]. Available: http://ieeexplore.ieee.org/document/5356337

[150] R. Zhang, Y. Xu, D. Harrison, J. Fyson, F. Qiu, and D. Southee, "A study of flexible supercapacitors for future energy storage," pp. 1–4, 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6661997

[151] F. Deng and J. Liu, "Design of an intelligent energy gateway for energy internet," in *2016 IEEE PES Asia-Pacific Power and Energy Engineering*

Conference (APPEEC), Oct 2016, pp. 1465–1468. [Online]. Available: https://ieeexplore.ieee.org/document/7779733

[152] T. Sauter and M. Lobashov, "End-to-end communication architecture for smart grids," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 4, pp. 1218–1228, April 2011. [Online]. Available: https://ieeexplore.ieee.org/document/5559470

[153] R. Khan, K. Mclaughlin, D. Laverty, and S. Sezer, "Design and implementation of security gateway for synchrophasor based real-time control and monitoring in smart grid," *IEEE Access*, vol. 5, pp. 11 626–11 644, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7961134

[154] X. Chen, P. Ye, J. Wei, H. Yu, and K. Pan, "Implementation of multi-port ethernet interface preprocessor board for iec61850 process bus," in *2011 International Conference on Advanced Power System Automation and Protection*, vol. 2, Oct 2011, pp. 1608–1612. [Online]. Available: https://ieeexplore.ieee.org/document/6180754

[155] W. Wei-ming, D. Xiong-ying, and L. Yan, "The research and development of an intelligent merging unit based on iec61850-9-2," in *2011 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, July 2011, pp. 1238–1241. [Online]. Available: https://ieeexplore.ieee.org/document/5994084

[156] W. Gao, Y. Liu, G. Chen, and Q. Yang, "Development of a fpga-based protective relay in active distribution networks," in *2018 China International Conference on Electricity Distribution (CICED)*, Sep. 2018, pp. 1226–1230. [Online]. Available: https://ieeexplore.ieee.org/document/8592226

[157] H. Kirrmann, C. Honegger, D. Ilie, and I. Sotiropoulos, "Performance of a full-hardware ptp implementation for an iec 62439-3 redundant iec 61850 substation automation network," in *2012 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication Proceedings*, Sep. 2012, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/6336631

[158] X. S. Jin, R. R. Gokaraju, and E. F. Pajuelo, "Fpga implementation of a phaselet method for high speed distance relaying - preliminary results," in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, Oct 2017, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8286236

[159] P. Ferrari, A. Flammini, S. Rinaldi, and G. Prytz, "Applying ptp-to-sntp time-gateway to iec61850 systems," in *ETFA2011*, Sep. 2011, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/6059153

[160] Intel, "Overcoming Smart Grid Equipment Design Challenges with FPGAs," 2018. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01191-smart-grid-design.pdf

[161] M. Kang, S. Cai, Q. Tong, Bo Zhi, and X. P. Zhao, "Designation and development of hardware platform for intelligent terminal," in *2012 China International Conference on Electricity Distribution*, Sep. 2012, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/6508727

[162] System on Chip Engineering, "PreciseTimeBasic: IEEE 1588-2008 PTPv2 IP core," 2019. [Online]. Available: http://soc-e.com/products/precisetimebasic-ieee-1588-2008-v2-ptp-ip-core/

[163] Flexibilis, "Flexibilis Redundant Switch (FRS)," 2019. [Online]. Available: http://www.flexibilis.com/products/frs/

[164] System on Chip Engineering, "HSR-PRP Switch IP core," 2019. [Online]. Available: http://soc-e.com/products/hsr-prp-switch-ip-core-all-hardware-low-latency-switch-for-fpgas/

[165] Nettimelogic, "HSR & PRP," 2019. [Online]. Available: https://www.nettimelogic.com/red-hsr-prp.php

[166] Institute of Embedded Systems, "High Availability," 2019. [Online]. Available: https://www.zhaw.ch/en/engineering/institutes-centres/ines/products-and-services/high-availability/

[167] M. Urbina, A. Astarloa, J. LÃ¡zaro, U. Bidarte, I. Villalta, and M. Rodriguez, "Cyber-physical production system gateway based on a programmable soc platform," *IEEE Access*, vol. 5, pp. 20 408–20 417, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8052102

[168] "WISHBONE System-on-Chip (SoC)Interconnection Architecturefor Portable IP Cores," OPENCORES.ORG, Tech. Rep., 2002. [Online]. Available: https://cdn.opencores.org/downloads/wbspec_b3.pdf

[169] "IEC 61850-90-4:2013: Network engineering guidelines," 2013. [Online]. Available: https://webstore.iec.ch/publication/6025

[170] M. Dworkin, "Recommendation for Block Cipher Modes of Operation Methods and Techniques," p. 66, 2001. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf

[171] M. Montes and D. Penazzi, "Dos nuevos algoritmos de cifrado autenticado. Silver y CPFB," *Workshop de Seguridad Informática*, pp. 31–45, 2014. [Online]. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/42068/Documento_completo.pdf?sequence=1&isAllowed=y

[172] R. Usselmann, "Advanced Encryption Standard / Rijndael IP Core," 2002. [Online]. Available: https://opencores.org/project/aes_core/downloads

[173] K. Gaj and P. Chodowiec, "Fpga and asic implementations of aes," in *Cryptographic Engineering*, 2009.

[174] Mirabilis Design, "VisualSim: Modeling, Simulation, Exploration and Collaboration," 2019. [Online]. Available: https://www.mirabilisdesign.com/visualsim/

[175] Xilinx, "Zynq-7000 All Programmable SoC Family Product Tables and Product Selection Guide," 2019. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/zynq-7000-product-selection-guide.pdf

[176] Xilinx, "Zynq-7000 SoC:Technical Reference Manual," 2018. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf

[177] SoC-e, "SMARTzynq module: 5 Port Gigabit Ethernet Industrial Embedded Switch Module," 2016. [Online]. Available: http://soc-e.com/products/smart-zynq-module/

[178] F. Xia, L. Ma, J. Dong, and Y. Sun, "Network QoS management in cyber-physical systems," *Proceedings - The 2008 International Conference on Embedded Software and Systems Symposia, ICESS Symposia*, pp. 302–307, 2008.

[179] Xilinx, "Vivado Design Suite," 2019. [Online]. Available: https://www.xilinx.com/products/design-tools/vivado.html

[180] ——, "Xilinx Software Development Kit," 2019. [Online]. Available: https://www.xilinx.com/products/design-tools/embedded-software/sdk.html

[181] Port, "PROFINET Stacks, Tools and Driver," 2019. [Online]. Available: https://www.port.de/en/products/profinet.html

[182] "MinimalModbus 0.6: Easy-to-use Modbus RTU and Modbus ASCII implementation for Python," 2014. [Online]. Available: https://pypi.python.org/pypi/MinimalModbus/0.6

[183] M. Büsch, "PROFIBUS software stack," 2018. [Online]. Available: https://bues.ch/cms/hacking/profibus.html

[184] N. Moreira, J. Lázaro, U. Bidarte, J. Jimenez, and A. Astarloa, "On the Utilization of System-on-Chip Platforms to Achieve Nanosecond Synchronization Accuracies in Substation Automation Systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1932–1942, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7386687

[185] SoC-e, "MES, Managed Ethernet Switch IP Core ," 2016. [Online]. Available: http://soc-e.com/mes-managed-ethernet-switch-ip-core/

[186] MARWELL, "Integrated 10/100/1000 Mbps Energy Efficient Ethernet Transceivers," https://www.marvell.com/documents/eoxwrbluvwybgxvagkkf/, 2018.

[187] MZ Automation GmbH, "Open source libraries for IEC 61850 and IEC 60870-5-104," 2019. [Online]. Available: https://libiec61850.com/libiec61850/