Universidad de Granada

DEPENDABLE SYSTEMS OVER SYNCHRONOUS NETWORKS

Presented by

JOSÉ LUIS GUTIÉRREZ RIVAS

To apply for the
PHD DEGREE IN
INFORMATION AND COMMUNICATION TECHNOLOGIES

Advisors
ANTONIO JAVIER DÍAZ ALONSO
EDUARDO ROS VIDAL

Signed: José Luis Gutiérrez Rivas

July 2018

Editor: Universidad de Granada. Tesis Doctorales
Autor: José Luis Gutiérrez Rivas
ISBN: 978-84-1306-063-7
URI: http://hdl.handle.net/10481/54624

José Luis Gutiérrez Rivas: *Dependable Systems over Synchronous Networks,* Tesis Doctoral, © Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License Julio de 2018

VISTO BUENO

El **Prof. Dr. D. Antonio Javier Díaz Alonso**, Profesor Titular de Universidad, y el profesor **Prof. Dr. D. Eduardo Ros Vidal** , Catedrático de Universidad, del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada,

CERTIFICAN:

Que la memoria titulada:

"Dependable Systems over Synchronous Networks"

ha sido realizada por **D. José Luis Gutiérrez Rivas** bajo nuestra dirección en el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada para optar al grado de **Doctor en Informática**.

En Granada, a 25 de Julio de 2018.

Los Directores de la tesis doctoral:

Fdo. Antonio Javier Díaz Alonso y Eduardo Ros Vidal

DECLARACIÓN

El doctorando José Luis Gutiérrez Rivas y los directores de la tesis Antonio Javier Díaz Alonso y Eduardo Ros Vidal garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, Julio de 2018

José Luis Gutiérrez Rivas	Antonio Javier Díaz Alonso y Eduardo Ros Vidal

A mis padres, José Luis y María Esperanza.

ABSTRACT

This thesis dissertation presents our work with critical distributed applications in industrial network infrastructures. This work focuses on providing all elements on the grid with redundancy features to increase fault tolerance at both local and distributed levels with particular emphasis on timing features.

This dissertation is structured in four parts.

In the first part, we review the state-of-the-art, paying special attention to all the elements that conform a critical distributed system. This section starts with the evolution that safety-critical (SC) systems have experienced during the last years and their adaptation from single to multi-core architectures. Then, the progressive growth of power grid technologies into Smart Grid systems and their relationship with critical applications and their event synchronization needs. Different timing technologies are detailed with particular emphasis in the main one used in this thesis, the White Rabbit technology (WR), which is capable of providing sub-nanosecond accuracies over Ethernet-based networks. Finally, it has been included a brief market survey to compare our contribution to other existing technologies in the market.

In the second part, we review the methods to increase reliability in mixed-critical end-systems using multi-core architectures. We focus on the developed methods to isolate non-critical and critical parts in terms of hardware and software without increasing the certification costs of the system. This deployment is based on an industrial use case that describes an emergency stop of an industrial motor controller, used as proof of concept. This part ends with an analysis of the fault tolerance features of the system due to the implementation of redundant hardware components, safe communication channels and redundant software architectures.

In the third part, we move from inter-core communication problems to inter-processor communication networks. We review the methods to increase reliability, scalability and compatibility in industrial networks, focusing on data and time distribution. We firstly introduce the development of different clocks for the WR technology to increase scalability and industrial compatibility. Later, we describe the methods developed to provide WR timing networks with fault tolerance and single point of failure avoidance in ring topologies. This requires of switchover mechanisms

to change from a primary to a backup time reference, which is also described in this part of the text. The same way and for the sake of data transmission, we describe the redundancy mechanisms developed to guarantee data distribution and reception, thus increasing services availability and reducing network latency. Finally, we analyze the bandwidth and reliability of data distribution.

The fourth part corresponds to the integration of all previous concepts, redundant implementations and compatibility methods into a real mission-critical distributed control system over a synchronous network scenario. This system is composed of network devices with redundancy capabilities, acquisition modules and Remote Terminal Units (RTUs) interconnected in a ring network topology. This deployment includes the dissemination of redundant timing references using WR for the core of the network with the best accuracy possible (below 1 ns). Moreover, other industrial timing solutions like the Precision Time Protocol (PTP) and IRIG-B are used for the acquisition modules and RTUs. Data is also exchanged through reliable communication channels. Finally, a safety tool has been used to evaluate all the elements that form the system in terms of their criticality and integrity levels.

RESUMEN

Esta tesis presenta el trabajo realizado sobre aplicaciones distribuidas críticas en infraestructuras de red industriales. Dicho trabajo se centra en proporcionar a todos los elementos de la red capacidades de redundancia para así incrementar la tolerancia a fallos tanto a nivel local como distribuido, poniendo especial énfasis en aspectos de sincronización.

Esta tesis está estructurada en cuatro partes.

En la primera revisaremos el estado del arte, poniendo especial atención en todos aquellos elementos que forman parte de un sistema crítico distribuido. Esta sección comienza con la evolución experimentada por los sistemas críticos (SC) a lo largo de los últimos años y su adaptación desde las arquitecturas mononúcleo a las multinúcleo. Posteriormente nos centraremos en el crecimiento progresivo experimentado por las redes eléctricas hacia los sistemas inteligentes, conocidos como Smart Grid, junto a su relación con aplicaciones críticas, además de sus necesidades relacionadas con la sincronización de los diferentes eventos que tiene lugar a lo largo de la red. Mostraremos diferentes tipos de tecnologías de sincronización, con especial atención sobre la principalmente utilizada en esta tesis, la tecnología White Rabbit (WR). Esta tecnología es capaz de proporcionar una precisión por debajo del nanosegundo en redes Ethernet. Para finalizar este apartado, hemos incluido un breve estudio de mercado para comparar nuestra contribución con otras tecnologías existentes en el mercado.

En la segunda parte revisaremos los métodos utilizados para incrementar la fiabilidad en sistemas finales (nodos hoja) de criticidad mixta utilizando arquitecturas multinúcleo. Pondremos especial atención en el desarrollo de métodos para aislar las partes críticas de las no críticas tanto en software como en hardware, sin incrementar los costes del proceso de certificación del sistema final. Este desarrollo está basado en un caso de uso industrial utilizado como prueba de concepto, en el que se define la parada de emergencia de un controlador de un motor industrial. Finalizaremos esta parte con un análisis de las características de tolerancia a fallos añadidas al sistema gracias a la duplicidad de los recursos hardware, la definición de arquitecturas software redundantes y de la integración de canales de comunicación confiables entre diferentes procesadores.

En la tercera parte, pasaremos de la problemática de las comunicaciones entre núcleos de un mismo dispositivo, a las comunicaciones entre diferentes procesos distribuidos a lo largo de la red. Revisaremos diferentes métodos para incrementar la fiabilidad, escalabilidad y compatibilidad en redes industriales, poniendo especial interés en la distribución de tiempo y datos. En primer lugar, introduciremos el desarrollo de diferentes tipos de relojes para la tecnología WR con el fin de incrementar su escalabilidad y su compatibilidad industrial. A continuación, describiremos los métodos desarrollados para proporcionar a las redes de sincronización WR mecanismos para incrementar la tolerancia a fallos y así evitar puntos únicos de error en topologías de red en anillo. Esto requiere la implementación de mecanismos de conmutación para cambiar de una fuente primaria de sincronización a una de respaldo (backup), los cuales son también descritos en esta parte del documento. Del mismo modo, describiremos los mecanismos de redundancia desarrollados para garantizar la distribución y recepción de datos, incrementando así la disponibilidad de los servicios proporcionados en la red y reduciendo la latencia de transmisión. Finalmente, analizaremos el ancho de banda y la fiabilidad con la que se transmiten dichos datos.

La cuarta y última parte corresponde a la integración de los conceptos anteriormente descritos, tales como pueden ser las implementaciones redundantes y los métodos para incrementar la compatibilidad entre los diferentes elementos que forman un sistema de control distribuido para aplicaciones de misión-crítica sobre redes sincronizadas real. Dicho sistema estará formado por dispositivos de red que integran capacidades redundantes, módulos de adquisición de datos y terminales remotos (RTUs), todos ellos interconectados en una red anillada. Este despliegue incluye la diseminación de una referencia de tiempo duplicada, en la cual se utiliza WR para el núcleo de sincronización de la red, proporcionando la mejor precisión posible (por debajo de 1 ns). Por otro lado, se han utilizado otras soluciones industriales para sincronizar los módulos de adquisición y RTUs, como son el Protocolo de Precisión de Tiempo (PTP) e IRIG-B. Además, los datos se envían a través de la red de forma segura gracias a los canales de comunicación confiables. Finalmente, se ha utilizado una herramienta de seguridad capaz de evaluar los elementos que forman el sistema según su grado de criticidad para así definir sus niveles de integridad.

AGRADECIMIENTOS

En realidad me ha costado la misma vida ponerme a escribir los agradecimientos, pero por fin estoy aquí. Espero no olvidarme de nadie ni ponerme cansino.

Empecemos por la etapa UGR, por esos ratos en la máquina de café o en el *Chill Out* del CITIC, buscando el sentido de la vida académica, hablando de videojuegos, o simplemente echando unas risas con Niceto, Fran, Leo, Richard, Fergu, Antares y Gonzalo. Además de ellos, sois muchos los que habéis formado parte de esta etapa: A. Mora, David, Luca, Almu, Pepe, Fránsfuga, Paco, Mario, Pablo, Jesús, Rodrigo, Pablo Guzmán y cómo no, Manolo (no es del CITIC, pero como si lo fuera). A José, por el tiempo invertido en el HSR, y a Felipe, detractor de los *transparent clocks*, por sus horas midiendo *offsets*. No quiero olvidarme del personal técnico y administrativo del CITIC como Paco Illeras, Beate Krug y Paco Blas, siempre dispuestos a ponerte las cosas fáciles.

A menos de 100 metros empieza una etapa diferente en Seven, pero igualmente divertida con Emilio, Miguel Méndez, Ana, Germán, Fidel, Rafa, Alicia, Miguel Ángel, Andrés, Pablo, Pilar, Paco, Benoit ..., sin olvidarme de Fran ni de Marta, por haberos dado la tabarra con mis *necesito 6 switches*, *se ha roto uno*, *se ha roto otro pero no sé cómo...*, *ahora necesito 22 WR-LENs*, etc.

A *Los Wannabe*: Joaquín, Piñeiro, Carei, Mary y Lore, por toda una vida juntos y por todos esos ratos compartidos, a pesar de ser una panda de quemasangres profesionales (¿para cuándo el *WannaCoffee*?); y en especial, a Santi, por su cariño, comprensión y por confiar en mí incluso en aquellas ocasiones en las que ni yo mismo soy capaz de hacerlo, mil gracias.

A mi familia, por apoyar todas y cada una de las decisiones que he tomado a lo largo de mi vida. A mi padre, por transmitirme valores como la honestidad y la responsabilidad con su propio ejemplo. A mi madre, por su cariño, generosidad y absoluta entrega. Y a mi hermana Nuria, por su bondad y por haber cuidado siempre de mí desde que tengo uso de razón.

Y finalmente, pero no por ello menos importante, me gustaría darle las gracias a mis directores de tesis Javier y Eduardo por haberme dado la oportunidad de empezar este camino, además del apoyo y guía durante todos estos años.

¡Gracias a todos!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

1oo2	1 out of 2 channel architecture
5G	5^{th}-Generation Wireless Systems
ACAS	on-board collision avoidance
ACP	Avionic Computing Platform
AMP	Asymmetric Multi-core Processing
ARM	Advanced RISC Machine
ASIL	Automotive Safety Integrity Level
BC	Boundary Clocks
BMCA	Best Master Clock Algorithm
BOM	Bill of Materials
CAN	Controller Area Network Bus
CERN	European Organization for Nuclear Research
C2C	Core-to-Core
DAL	Design Assurance Level
DAN	Doubly Attached Node
DC	Diagnostic Coverage
DCS	Distributed Control Systems
DMTD	Dual Mixer Time Difference
DDMTD	Digital Dual Mixer Time Difference
DDR	Double Data-Rate SDRAM
DER	Distributed Energy Resources
DNP	Distributed Network Protocol
E2E	End-to-End
EDIF	Electronic Design Interchange Format
EDN	Electrical Distribution Network

EBI	External Bus Interface
EMC^2	Embedded multi-core systems for mixed Criticality applications in dynamic and changeable real-time environments project
ES	Emergency Stop
FAA	US Federal Aviation Administration
FFU	Fast Forwarding Unit
FMEDA	Failure Modes, Effects, and Diagnostic Analysis
FPGA	Field Programmable Gate Array
FSU	Fast Switchover Unit
FTA	Fault Tree Analysis
GM	Grandmaster
GNSS	Global Navigation Satellite Systems
GOOSE	Generic Object Oriented Substation Events
GPS	Global Positioning System
GPWS	Ground proximity warning systems
GSSE	Generic Substation State Events
HFT	Hardware Fault Tolerance
HSR	High-availability Seamless Redundancy Protocol
HY	Hybrid Clock
HW	Hardware
ICMP	Internet Control Message Protocol
ICS	Industrial Control Systems
IED	Intelligent Electronic Devices
IRIG	Inter-range instrumentation group time codes
IRIG-B	Inter-range Instrumentation Group B - Time Code
LRE	Link Redundancy Entity
LUT	Look-Up Table

MB	MicroBlaze Soft Processor
Mini-BP	Mini Back-Plane
NIC	Network Interface Controller
NoC	Network on Chip
NSC	Non Safety Critical
NTP	Network Time Protocol
OC	Ordinary Clocks
OHWR	CERN's Open Hardware Repository
OS	Operating System
P2P	Peer-to-Peer
PLB	Processor Local Bus
PLC	Programmable Logic Controllers
PMU	Phasor Measurement Unit
PPS	Pulse Per Second
PRP	Parallel Redundancy Protocol
PSU	PTP Support Unit
PTP	Precision Time Protocol
PTPv1	IEEE 1588-2002
PTPv2	IEEE 1588-2008
QCS	Quality Control Systems
QDRII	Quad Data Rate SRAM II
RECOMP	Reduced Certification Costs Using Trusted Multi-core Platforms project
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
RSB	RECOMP Sensor Board
RSTP	Rapid Spanning Tree Protocol
RTOS	Real-Time Operating System

RTU Remote Terminal Unit

SAN Single Attached Node

SAS Substation Automation Systems

SC Safety Critical

SCADA Supervisory Control and Data Acquisition

SCB Switch Core Board

SFF Safe Failure Fraction

SFP Small Form-factor Pluggable Transceiver

SIL Safety Integrity Level

SKA Square Kilometre Array Project

SMV Sampled Measured Values

SNTP Simple Network Time Protocol

SoA State-of-the-Art

SoC System on Chip

SRAM Static Random Access Memory

THR Tolerable Hazard Rate

STO Safe Torque Off

SyncE Synchronous Ethernet

SW Software

TAI International Atomic Time

TCB Trusted Computing Base

TC Transparent Clock

UTC Coordinated Universal Time

WAMPAC Wide Area Monitoring Protection and Control

WAMS Wide Area Monitoring System

WC2C Wittenstein Core-to-Core Library

WCE Worst Case Execution

WR-PTP White Rabbit Precision Time Protocol

WR White Rabbit

WRS White Rabbit Switch

Part I

INTRODUCTION

1 INTRODUCCIÓN

La ambición es el camino al éxito,
la tenacidad, el vehículo en que se llega.

— Bill Bradley

INDEX

Este Capítulo introductorio comienza con una visión general del contexto y dominio de aplicación del trabajo realizado en esta tesis. La Sección 1.2 enumera los diferentes objetivos científicos establecidos en una primera etapa inicial. La Sección 1.3 resume los proyectos de índole nacional e internacional en los que se han desarrollado los conceptos, diseños e implementaciones necesarios para lograr dichos objetivos. La Sección 1.4 presenta la metodología seguida y las herramientas utilizadas. Finalmente, la Sección 1.5 resume brevemente la organización de los capítulos de este documento.

1.1 GENERAL

En los últimos años, los sistemas de control y más en concreto los sistemas de control distribuidos (DCS) han ido evolucionando para mejorar tanto su operativa como su productividad. La introducción de nuevas funcionalidades embebidas en dispositivos e instrumentos de campo en el marco de los DCS ha forzado la evolución del modelo de comunicación, partiendo de una comunicación local entre diferentes núcleos de un procesador hacia un esquema de comunicación distribuida entre diferentes procesos para así mejorar el mantenimiento y los procesos de control, y por tanto, asegurar el correcto funcionamiento y estabilidad de la operativa de dichos sistemas. Esto ha dado lugar al incremento de la cantidad de datos intercambiados a través de la red, forzando el despliegue de grandes infraestructuras de red capaces de soportar comunicaciones con anchos de banda mayores. La Figura 1.1 representa las bases de un DCS junto a los modelos de comunicación inter-núcleo e inter-proceso.

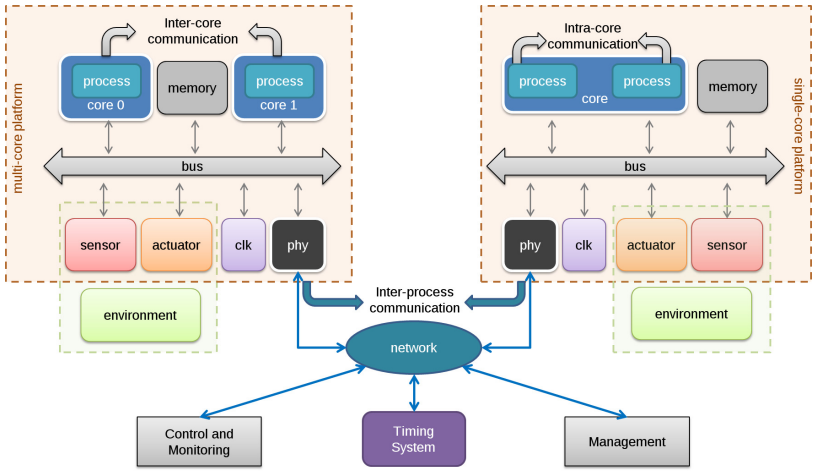

Figure 1.1
Arquitectura de un sistema de control distribuido. Los dispositivos multi-núcleo actúan como sistemas de adquisición o actuadores, intercambiando datos entre sus núcleos locales (comunicación inter-núcleo) o con otros dispositivos (comunicación inter-proceso) para interactuar con el entorno.

Es en este contexto donde el concepto de red inteligente, Smart Grid, toma lugar. Smart Grid combina la integración de diferentes sistemas de ingeniería eléctrica, almacenamiento de energía y avances en nuevas tecnologías de la información y de la comunicación en el dominio de la red eléctrica, desde su proceso de generación hasta su comercialización. Esto permite la interconexión de elementos como el control, instrumentación, medidas y administración de la energía en un sistema de manten-

imiento global para facilitar la distribución de la energía de una forma racional y eficiente [1].

Esta evolución en el sistema de mantenimiento global fuerza la utilización de infraestructuras de comunicación extendidas y confiables como puede ser Ethernet, además de una mejora en la disponibilidad de los servicios proporcionados en la red, junto a la necesidad de garantizar la seguridad y la fiabilidad debido a su naturaleza crítica [2, 3]. Por este motivo, estos sistemas deben incorporar diferentes mecanismos para incrementar su tolerancia a fallos además de facilitar la transferencia del control y toma de decisiones tan pronto como sea posible y con la menor latencia posible [4].

Dichas características relacionadas con la fiabilidad deben ser también consideradas para los datos gestionados por los nodos finales de una red (p.ej., dispositivos de adquisición). Estos dispositivos deben implementar mecanismos de fiabilidad para garantizar que los datos se han obtenido correctamente y que no hayan sido modificados antes de ser transmitidos por diferentes procesos. A este respecto, los estándares de seguridad, como el IEC 61508 [5], definen diferentes aproximaciones para evitar la corrupción de los datos. Entre estos mecanismos se encuentran la utilización de hipervisores y el desarrollo de técnicas de aislamiento entre diferentes procesos y recursos.

Con respecto a las comunicaciones entre diferentes procesos distribuidos, la información y datos de control que se transmiten a través de redes Smart Grid son considerados críticos, debido a que la no recepción de mensajes de control y monitorización puede dar lugar a lesiones personales, además de enormes pérdidas monetarias. En este contexto, IEC 61850 [6] sugiere la implementación de protocolos de redundancia para evitar la pérdida de datos. Existen diferentes protocolos que pueden ser desarrollados para este fin, pero son el Protocolo de Redundancia Paralela (PRP) y el Protocolo de Redundancia de Alta Disponibilidad (HSR), estandarizados por la Comisión Electrotécnica Internacional en Ginebra, los más utilizados [7].

Además, será necesario que todos los dispositivos de la red estén sincronizados para compartir una noción de tiempo común, que será utilizada en los procesos de captura y generación de eventos [8]. Por este motivo, deberá integrarse en Smart Grid la posibilidad de realizar dicha sincronización con la mayor fiabilidad y precisión posibles. Refiriéndonos a la sincronización, el requisito más estricto a día de hoy proviene de las unidades de medida de fasor (PMUs), demandando precisiones por debajo de los 10 ns [9]. Los Sistemas Globales de Navegación por Satélite (GNSS), a pesar de ser ampliamente utilizados en aplicaciones

industriales y militares, están sujetos a pérdidas de rendimiento en la sincronización, a la vez que pueden sufrir ataques que vulneren su seguridad (interferencias, redireccionamientos indeseados, etc). Por este motivo, IEEE recomienda utilizar métodos alternativos a GNSS que utilicen sistemas terrestres [10]. La utilización de una solución capaz de combinar GNSS y tecnologías cableadas como puede ser IRIG-B, o incluso tecnologías basadas en Ethernet como NTP y PTP, se está volviendo muy cada vez más popular en este tipo de infraestructuras [11].

Unificando todos estos conceptos, una red inteligente estará formada por un conjunto de módulos de adquisición, sensores y actuadores ejecutando software confiable en arquitecturas multinúcleo, formando así un DCS. Todos los elementos estarán interconectados a través de una infraestructura de red redundante y sincronizada con la mejor precisión posible. La Figura 1.2 muestra la arquitectura que unifica y resume este concepto, el cual es la estructura principal de esta tesis doctoral.

Los dispositivos de adquisición están formados por sensores que interactúan con el entorno. La información extraída por los sensores es transmitida a través de canales de comunicación confiables. Estos canales duplican los datos y la información de sincronización para así garantizar la fiabilidad antes fallos, además de la disponibilidad de los servicios de la red. Finalmente, el sistema de control monitoriza y diagnostica el comportamiento de los elementos de la red. Los eventos son marcados y almacenados junto a un sello de tiempo muy preciso, para así poder realizar posteriores análisis del estado de la red o comportamiento del sistema.

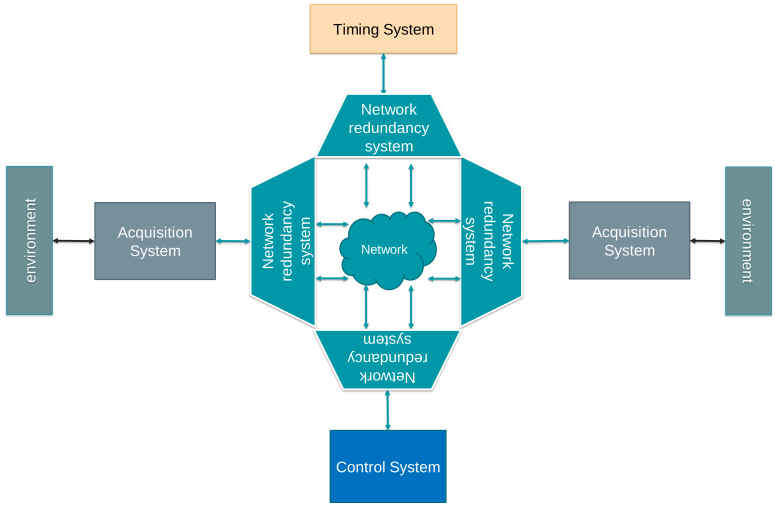

Figure 1.2
Arquitectura de un sistema Smart Grid. Los sistemas de control reciben los datos de los módulos de adquisición a través de topologías de red redundantes, asegurando así la disponibilidad de los servicios de la red y la recepción de datos. El sistema de control evalúa estos datos e interacciona con el entorno en función de los resultados de los datos analizados. Todos los dispositivos de la red están sincronizados con la misma referencia temporal.

Los objetivos de esta tesis se centran en proporcionar mecanismos de tolerancia a fallos a los sistemas Smart Grid a todos los niveles de los que está constituida la red de distribución del mismo. Esto conlleva asegurar la fiabilidad de las comunicaciones entre procesos distribuidos y en arquitecturas multinúcleo, mediante el desarrollo de mecanismos de redundancia y aislamiento para asegurar la integridad de los datos, garantizando así las comunicaciones entre los diferentes procesos de la red mediante el desarrollo de protocolos de redundancia como puede ser el HSR, junto a la distribución de una fuente común de sincronización a todos los dispositivos con la mejor precisión posible. La Sección 1.2 enumera estos objetivos.

1.2 OBJETIVOS CIENTÍFICOS

Los objetivos científicos establecidos para el desarrollo de esta tesis están relacionados con la fiabilidad de los sistemas de control para aplicaciones de misión crítica distribuidas a través de redes sincronizadas, poniendo especial atención en aplicaciones industriales tales como Smart Grid y Sistemas de Monitorización de Área Extensa (WAMS). El propósito de dichos objetivos se centra en el desarrollo de un esquema de comunicación confiable inter- e intra-chip capaz de propagar datos de criticidad mixta, considerando la distribución de una referencia de tiempo

común un elemento especialmente crítico. Estos objetivos son los siguientes:

- Diseño y desarrollo de redes de monitorización y control de gran despliegue con capacidades de sincronización de alta precisión desde el núcleo de la red hasta los nodos finales.
- Diseño y desarrollo de métodos para incrementar la fiabilidad de nodos de criticidad mixta multi-núcleo para aplicaciones industriales siguiendo las recomendaciones de los estándares de certificación industriales, tales como la implementación de mecanismos de aislamiento, canales de comunicación seguros, utilización de procesadores con capacidades de planificación de tareas, etc.
- Desarrollo de técnicas para mejorar la latencia y el determinismo en la transmisión de datos en redes confiables.
- Desarrollo e implementación de métodos y técnicas para mejorar la escalabilidad y compatibilidad de la sincronización en infraestructuras industriales.
- Desarrollo e implementación de protocolos de redundancia para incrementar la robustez y tolerancia a fallos en la distribución de tiempo.
- Desarrollo de mecanismos para conmutar de una fuente primaria de tiempo a una de respaldo, evitando cualquier tipo de impacto en la sincronización de la red.
- Desarrollo de protocolos de redundancia para incrementar la robustez y tolerancia a fallos en la transmisión de datos.
- Evaluar diferentes estrategias para reducir la latencia en la distribución de datos e incrementar la disponibilidad de los servicios proporcionados por la red crítica.
- Desarrollo de mecanismos de baja latencia para sistemas de alerta a través de la red.
- Aplicación y casos de uso. Estudiar diversos escenarios industriales relacionados con las aplicaciones críticas con el fin de integrar los resultados de los desarrollos anteriormente descritos en un entorno real.

1.3 MARCO DEL PROYECTO DE TESIS

El trabajo llevado a cabo en esta tesis ha sido principalmente realizado en el marco de cuatro proyectos de investigación de

índole nacional e internacional. Estos proyectos son RECOMP[1], SKA[2], AMIGA6[3] y EMC[2]4. Del mismo modo, esta tesis ha sido respaldada por la empresa Seven Solutions S.L. y parcialmente financiada por el Ministerio de Economía, Crecimiento y Competitividad[5] como parte del programa *Ayudas para contratos para la formación de investigadores en empresas 2014 (Doctorados Industriales)*[6].

1.3.1 *RECOMP*

Reduced Certification Costs Using Trusted Multi-core Platforms (RECOMP) es un proyecto europeo ARTEMIS-JU (contrato número 100202) en el que participan más de 40 socios industriales y académicos. Su principal objetivo es establecer métodos, herramientas y plataformas para habilitar una recertificación efectiva en costes para los sistemas de criticidad mixta y críticos. El dominio de aplicación de dichos desarrollos están enfocados a la automoción, ingeniería aeroespacial e industria.

Las bases de RECOMP se centran en la utilización de plataformas multi-núcleo para incrementar la potencia de procesamiento de sistemas embebidos en lugar de utilizar arquitecturas mononúcleo. El incremento en el número de núcleos se considera un reto, ya que las guías de desarrollo no contemplan dichas arquitecturas para certificar un sistema.

Por otro lado, el incremento en la necesidad de productos flexibles en el mercado ha forzado la definición de nuevos requisitos para actualizar las características de los sistemas con partes críticas y no críticas, dando lugar a un incremento de los costes en el proceso de recertificación de un producto. Solventar y/o reducir el impacto de los costes del proceso de certificación mediante el desarrollo de mecanismos y herramientas para reducir rehacer por completo el proceso mediante el aislamiento y particionamiento en tiempo y espacio de procesos y recursos, es el principal objetivo de RECOMP. Este será el punto de partida de esta tesis. El Capítulo 4 describe los métodos desarrollados para incrementar la fiabilidad de la comunicación inter-núcleo junto a su utilización en un caso de uso industrial real y sus resultados.

RECOMP finalizó en 2013, proporcionando diseños, arquitecturas y plataformas de referencia, junto a diferentes métodos y

1 http://atcproyectos.ugr.es/recomp/
2 http://skatelescope.org/
3 http://amiga.iaa.es/
4 https://www.artemis-emc2.eu/
5 http://www.mineco.gob.es
6 https://goo.gl/HDcPnw

herramientas para reducir los costes del proceso de certificación y recertificación de sistemas de criticidad mixta utilizando arquitecturas multi-núcleo.

1.3.2 *EMC²*

EMC². *Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments* es un proyecto ARTEMIS JU dentro del programa piloto "Plataformas de Computación para Sistemas Embebidos" (AIPP5).

Hoy en día, los sistemas embebidos son un principal impulsor para mejorar la mayoría de los productos mecatrónicos con nuevas e incluso más baratas funcionalidades. Estos sistemas apoyan a la actual sociedad de la información como un habilitador de las comunicaciones entre sistemas. Un gran reto surge de la necesidad de afrontar la eficiencia en costes en cuanto a la integración de diferentes aplicaciones con diferentes niveles de seguridad y fiabilidad en una única plataforma en cualquier contexto.

EMC² busca soluciones para la adaptabilidad dinámica en sistemas abiertos, proporcionando herramientas para manejar aplicaciones de criticidad mixta bajo condiciones en tiempo real, escalabilidad y flexibilidad, junto a un despliegue y mantenimiento escalable de dichas herramientas a lo largo de todo el ciclo de vida del sistema.

El objetivo de EMC² es establecer la tecnología multi-núcleo en todos los dominios relacionados con sistemas embebidos. EMC² es un proyecto formado por 99 socios de la industria y academia relacionados con los sistemas embebidos de 19 países europeos con un esfuerzo total 800 personas anuales y un presupuesto total alrededor de 100 millones de euros.

EMC² ha sido el proyecto que ha logrado unificar los conceptos de SC y misión crítica en la misma dirección que las tecnologías distribuidas como pueden ser el Internet de las Cosas, Smart Grid y WAMS. Es en este marco de trabajo donde esta tesis tiene lugar. El Capítulo 5 presenta y analiza el trabajo realizado para EMC², poniendo especial interés en la escalabilidad, estabilidad y fiabilidad de la sincronización de toda la red y de la trasmisión de datos. El Capítulo 6 representa el esfuerzo en la integración de los mecanismos desarrollados, técnicas y tecnologías que dan forma a un escenario real de Smart Grid.

1.3.3 *SKA*

El proyecto *The Square Kilometre Array* (SKA) está destinado a ser el radio-telescopio más grande y sensitivo en el mundo, ubicando miles de receptores de ondas de radio en Australia y Sudáfrica. La combinación de las señales que provienen de las antenas de cada región crearán un telescopio con una superficie colectora equivalente a la de un plato con un área de un kilómetro cuadrado.

SKA planea resolver cuestiones fundamentales sin respuesta sobre el Universo, como puede ser la creación de las estrellas y las galaxias tras el Big Bang, su evolución, el rol que ejerce el magnetismo en el cosmos, la naturaleza de la gravedad y la búsqueda de vida más allá de la Tierra.

SKA es un proyecto científico y de ingeniería global dirigido por la Organización SKA, una compañía ubicada en el observatorio de Jodrell Bank, Manchester, UK.

SKA requiere sistemas capaces de distribuir referencias temporales muy precisas para sincronizar todas las antenas y dispositivos de adquisición en todas las localizaciones con la misma noción de tiempo, para así construir una imagen muy precisa del firmamento. Además, la robustez y fiabilidad de dicho sistema de sincronización forma parte de los requisitos de SKA, alineándose así perfectamente con los objetivos de esta tesis doctoral.

1.3.4 *AMIGA6*

AMIGA6: Gas en el interior y en el entorno de las galaxias. Preparación científica para SKA y contribución al diseño de flujos de datos toma como punto de partida los resultados de proyectos AMIGA previos, además de participar en la fase de preparación para la explotación científica de los resultados de SKA. Los ciclos de vida HI tanto en galaxias aisladas como formando densos grupos son bastante desconocidos ya que sus columnas de baja densidad solo son alcanzables gracias a SKA. Los principales objetivos de AMIGA6 son: perfeccionar modelos de acumulación de gas frío utilizando galaxias aisladas y analizar el rol de la eliminación de las mareas HI en grupos compactos de Hickson.

Como parte de la preparación para el reto propuesto por SKA en cuanto a la explotación de los datos recogidos, AMIGA6 dota a la ciencia fundamental de investigación aplicada en 3 paquetes de trabajo. AMIGA6 es liderado por el coordinador español de SKA, y recoge todos los grupos involucrados en el flujo de datos de SKA: Transporte de Datos y Señales (SaDT), Procesamiento

Central de Señales (CSP) y Procesamiento de Datos Científicos (SDP).

1.4 MÉTODOS Y HERRAMIENTAS

La metodología utilizada para la realización de esta tesis ha sido principalmente experimental, tomando como punto de partida el análisis teórico.

Se desarrolló una metodología experimental para caracterizar redes WR para así determinar los parámetros fundamentales relacionados con la tecnología. Entre otros, se puso especial interés en aquellos asociados con las medidas de determinismo del sistema (latencia e inestabilidad del reloj interno) junto al ancho de banda de la red en situaciones de carga intensa de datos. El resultado de esta fase estará compuesto por la definición de modelos de sistema, el análisis de dichos modelos y sus posibles mejoras.

Para este fin, fue necesario desarrollar implementaciones reales basadas en las mejoras propuestas extraídas de diferentes análisis llevados a cabo en la primera etapa de la tesis. Para ello, un primer análisis teórico fue necesario para determinar las limitaciones de los experimentos que debían realizarse. Además, se utilizaron las guías de desarrollo de aplicaciones confiables para aviónica, automoción e industria como parte de la bibliografía. Estos análisis dieron como resultado diferentes aspectos a tener en cuenta en cuanto a topologías de red, técnicas de medida, posibilidad de tener múltiples fuentes de sincronización, junto a las consideraciones a tener en cuenta para posteriormente realizar los experimentos.

En relación a las tareas de desarrollo, se han utilizado diferentes lenguajes de programación dependiendo del nivel de abstracción de la implementación a realizar. Para el desarrollo de software, se utilizó SystemC[7] para la primera parte del diseño conceptual de las aplicaciones críticas. Para el resto de los desarrollos realizados el lenguaje de programación predominante ha sido C.

Estos programas en C han sido compilados para diferentes arquitecturas a lo largo del desarrollo de esta tesis: procesadores dentro de FPGAs (como MicroBlaze y LM32) y procesadores avanzados RISC (ARM). En muchas ocasiones se han utilizado entornos de desarrollo y herramientas de compilación como Buildroot[8], utilizado especialmente para todos los desarrollos rela-

7 http://www.accellera.org/downloads/standards/systemc
8 https://buildroot.org/

cionados con el switch WR (WRS). En cuanto al sistema operativo (OS) se ha utilizado Ubuntu 14.04 LTS.

Las principales plataformas de desarrollo han sido la Avionics Computing Platform (ACP), el WRS[9] y la WR-LEN[10]. Todas estas plataformas incluyen FPGAs de Xilinx (Virtex 6 y procesadores Zync SoC), por lo que el principal lenguaje de programación ha sido VHDL. Por este motivo, se han utilizado dos entornos de desarrollo (IDEs): la Xilinx ISE Design Suite[11] para la Virtex-6, y Vivado[12] para desarrollos basados en Zync.

El material de medida está principalmente formado por osciloscopios y counters. Las medidas de grano grueso se han realizado utilizando un osciloscopio PicoScope 3000 Series[13], mientras que las medidas de grano fino han sido realizadas con un Tektronix FCA3000 Counter[14].

1.5 ORGANIZACIÓN DE LOS CAPÍTULOS

Más allá de este Capítulo introductorio 1, el cual comienza con una visión general de los temas que se abordarán en esta tesis junto a los objetivos científicos establecidos, la organización de este documento continúa con el Capítulo 3, el cual presenta el estado del arte relacionado con los sistemas de control críticos distribuidos. Incluye la evolución de los sistemas de criticidad mixta desde las implementaciones mono-núcleo hasta las multi-núcleo junto a los desafíos que esto conlleva, los conceptos principales de los sistemas de control distribuido y su evolución hacia Smart Grid, seguido de la importancia de la distribución de un sistema de sincronización en estas redes, finalizando con un estudio de mercado sobre dispositivos confiables de distribución de tiempo para aplicaciones industriales.

El Capítulo 4 expone los métodos para incrementar la fiabilidad de las aplicaciones críticas utilizando plataformas multi-núcleo en cuanto a hardware, software y comunicación inter-núcleo.

El Capítulo 5 presenta un estudio de la problemática relacionada con la distribución de tiempo y datos fiable en aplicaciones industriales. Este Capítulo incluye la descripción de diferentes mecanismos para incrementar la compatibilidad industrial, la fiabilidad y tolerancia a fallos de la distribución de tiempo y datos.

9 http://sevensols.com/index.php/products/white-rabbit-switch/
10 http://sevensols.com/index.php/products/wr-len/
11 https://www.xilinx.com/products/design-tools/ise-design-suite.html
12 https://www.xilinx.com/products/design-tools/vivado.html
13 https://www.picotech.com/products/oscilloscope
14 https://www.tek.com/datasheet/fca3000-and-fca3100-series

El Capítulo 6 propone un escenario Smart Grid real como prueba de concepto para los desarrollos realizados y presentados en los Capítulos 4 y 5 de esta tesis.

Finalmente, el Capítulo 7 resume las principales contribuciones de la investigación realizada y sugiere diferentes líneas de trabajo futuro.

2 INTRODUCTION

Ambition is the path to success,
persistence is the vehicle you arrive in.

— Bill Bradley

INDEX

This introductory Chapter starts with a general overview of the framework and application domain of this thesis. Section 2.2 enumerates the different scientific objectives established at the earliest definition phase. Section 2.3 summarizes the national and international research projects in which the concepts, designs and developments have been done. Section 2.4 presents the methods and tools used during the conduct of this thesis. Finally, Section 2.5 briefly summarizes the organization of the chapters of this document.

2.1 GENERAL

In recent years, control systems and more precisely Distributed Control Systems (DCS) have been evolving to improve plant operation and also to increase productivity. The introduction of advanced embedded functionalities in field devices and instruments in the framework of DCS has forced to evolve from the classical inter-core communication approach to a distributed inter-process communication scheme to improve management and control processes in order to ensure plant operation stability. This has resulted in an increment in the amount of exchanged data over the network and thus, becoming necessary the deployment of large network infrastructures supporting higher communication bandwidths. Fig. 2.1 represents the basis of a DCS and its inter-core and inter-process communication models.

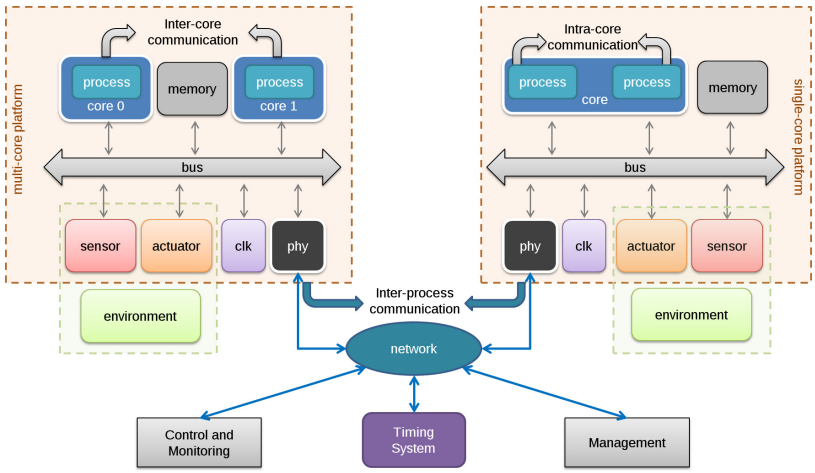

Figure 2.1
Distributed Control System Architecture. Multi-core devices act as acquisition systems or actuators, exchanging data between their local cores (inter-core communication) or with other devices (inter-process communication) to interact with the environment.

It is within this context that the Smart Grid concept takes place. Smart Grid combines the integration of different electrical engineering, energy storage and advances in new information and communication technologies within the electrical power domain, from the generation process to its commercialization. This allows the interconnection between the control, instrumentation, measurement and energy administration in a global management system in order to facilitate rational and efficient use of energy [1].

This evolution on the global management system imposes the utilization of reliable and widespread communication network

infrastructures like Ethernet and a desirable improvement in network services availability, security and safety due to their critical nature [2, 3]. For this reason, these systems need to incorporate different mechanisms to increase fault tolerance and the avoidance of single point of failure. Furthermore, they also facilitate the transfer of control decisions and actions as soon as possible with the lowest possible latency [4].

These reliability features must also be considered for the data handled at leaf-nodes (e.g. acquisition device). These devices must implement dependability mechanisms to ensure that data have been correctly obtained and not been modified before being transmitted by inter- or intra-core processes. In this regard, the utilization of hypervisors, isolation mechanisms and other suggestions described in safety standards such as IEC 61508 [5], define different approaches to avoid data corruption.

Regarding inter-process communications, the information and control data that are transmitted through Smart Grid networks are considered critical since a non-received control event message could lead to personal injuries and also the loss of huge sums of money. In this regard, IEC 61850 [6] suggests the implementation of redundancy protocols to avoid data loss. Several protocols can be developed for this domain but, the Parallel Redundancy Protocol (PRP) and the High-availability Seamless Redundancy (HSR) protocol, standardized by the International Electrotechnical Commission in Geneva, are the most commonly used [7].

Furthermore, the correct synchronization of the devices that form the grid is crucial to maintain a common notion of time of when and what occurred on the network [8]. For this reason, reliability and high-accurate timing must also be integrated in Smart Grid. In terms of time synchronization, the highest accuracy requirement comes from Phasor Measurement Units (PMUs), demanding accuracies below 10 ns [9]. Global Navigation Satellite Systems (GNSS), in spite of being globally used, are subject to lose synchronization performance and to be compromised in terms of security (vulnerability to spoofing, jamming, etc). IEEE recommendations focus on providing an alternative method to GNSS by using terrestrial systems [10]. The utilization of a solution combining GNSS and wired technologies such as IRIG-B and recently, the Ethernet-based PTP IEEE 1588 [12], is becoming widely popular [11].

Bringing all these concepts and features together, it follows that the grid is finally composed of acquisition modules, sensors and actuators running dependable software on multi-core architectures, which are integrated into a DCS. Everything is in-

terconnected through a redundant network infrastructure and synchronized with the best accuracy possible. Fig. 2.2 depicts the architecture that embodies this concept and summarizes the structure of this doctoral thesis.

Acquisition devices consist of sensors that interact with the environment. The information extracted by the sensors is transmitted over reliable communication channels. These channels duplicate the data and timing information in order to guarantee network fault tolerance and services availability. Finally, the control system monitors and diagnoses the behavior of the elements of the grid. Events are precisely time-stamped and stored in a historical database for later analysis.

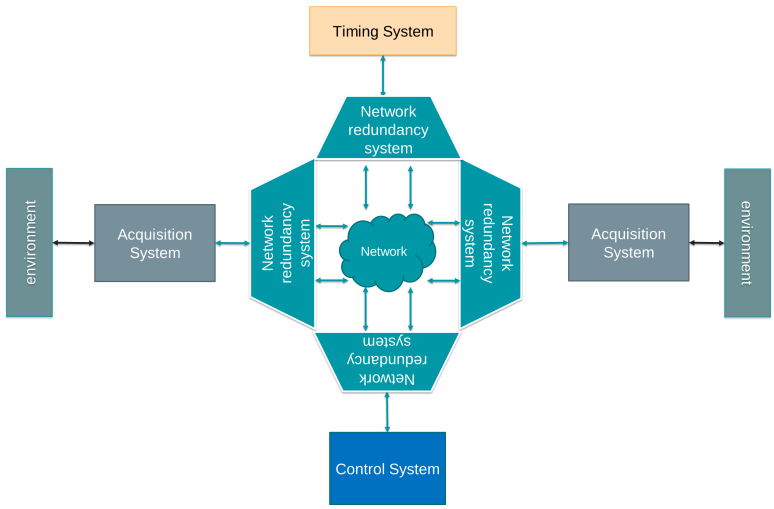

Figure 2.2
Smart Grid system architecture. The control system receives data from the acquisition modules through a redundant network topology, thus ensuring service availably and data reception. The control system evaluates these data and interacts with the environment according to the data analyzed. All devices from the network are synchronized to the same time reference.

Finally, the objectives of this thesis focus on providing Smart Grid systems with fault tolerance mechanisms at all levels of the distributed network. This entails ensuring reliability of multicore inter-core communications by the development of redundancy and isolation mechanisms to ensure data integrity, guaranteeing inter-process communications over Ethernet networks by the development of redundancy protocols such as HSR, and disseminating a common time reference to all devices with the best accuracy possible. Next Section 2.2 enumerates these objectives.

2.2 SCIENTIFIC OBJECTIVES

The scientific objectives established for the development of this thesis are related to the dependability of mission-critical control systems over synchronized networks, focusing on industrial applications such as Smart Grid and Wide Area Monitoring Systems (WAMS). This research targets the development of a dependable inter- and intra-chip communication scheme capable of disseminating mixed criticality data, especially considering timing distribution as a critical distributed element. More specifically, our main goals are the following:

- Design and development of widely deployed monitoring and control networks with high-accuracy timing capacities from the core to end-nodes.
- Design and development of methods to increase reliability in mixed-criticality multi-core nodes for industrial applications following the suggestions of industrial certification standards, such as isolation mechanisms, reliable communication channels, processor scheduling capabilities, etc.
- Development of techniques to improve data latency and determinism for dependable networks.
- Development and implementation of methods and techniques to improve timing scalability and compatibility for industrial infrastructures.
- Development and implementation of redundancy protocols to increase robustness and fault tolerance for time distribution.
- Development of mechanisms to change from a primary to a backup time reference so that the impact in time synchronization is seamless.
- Development of redundancy protocols to increase robustness and fault tolerance for data transmission.
- Assessment of strategies to reduce latency data distribution and increase service availability in critical networks.
- Development of low-latency mechanisms for system alerts over the network.
- Application uses cases. Study of diverse industrial network scenarios for time-critical or mission-critical applications to integrate the resulting developments.

2.3 PROJECT FRAMEWORK

The work carried out within this thesis has been mainly done in the framework of four national and international research projects named RECOMP[1], SKA[2], AMIGA6[3] and EMC[2][4]. Furthermore, it has been supported by Seven Solutions S.L. and partly funded by the Ministry of the Economy, Growth and Competitiveness[5] as part of the *Ayudas para contratos para la formación de investigadores en empresas 2014 (Doctorados Industriales)* program[6].

2.3.1 *RECOMP*

Reduced Certification Costs Using Trusted Multi-core Platforms (RECOMP) is an European ARTEMIS-JU project (contract number 100202) with more than 40 industrial and academic partners. Its main aim is to establish methods, tools and platforms for enabling cost-effective recertification of safety-critical (SC) and mixed-criticality systems. Development domains were automotive, aerospace, industrial control systems and transportation systems.

RECOMP bases focus on the utilization of multi-core platforms to increase processing power of embedded systems instead of using single core architectures. Increasing the number of cores becomes a design challenge for SC systems, as there are no established recommendations guides to achieve certification.

On the other hand, the increased need for product flexibility in this market forces the definition of new requirements on the customization and upgrade features of both safety and non-safety parts of these systems. This leads to an increase in expenditure for the recertification. This issue is the main problem that RECOMP addresses by the development of mechanisms and tools that reduce performing a complete certification process by isolation and partitioning in time and/or space techniques, which is also the starting point of this thesis. Chapter 4 describes the methods developed to increase inter-core communication reliability within the project, their use in an industrial use case and their results.

1 http://atcproyectos.ugr.es/recomp/
2 http://skatelescope.org/
3 http://amiga.iaa.es/
4 https://www.artemis-emc2.eu/
5 http://www.mineco.gob.es
6 https://goo.gl/HDcPnw

RECOMP ended in 2013 and provided reference designs and platform architectures, together with the required design methods and tools, for achieving cost-effective certification and recertification of mixed-criticality, component based, multi-core systems.

2.3.2 *EMC²*

EMC². *Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments* is an ARTEMIS Joint Undertaking project in the Innovation Pilot Programme "Computing platforms for embedded systems" (AIPP5).

Embedded systems are the key innovation driver to improve almost all mechatronic products with cheaper and even new functionalities. They support today's information society as inter-system communication enabler. A major industrial challenge arises from the need to face cost efficient integration of different applications with different levels of safety and security on a single computing platform in an open context.

EMC² finds solutions for dynamic adaptability in open systems, provides handling of mixed criticality applications under real-time conditions, scalability and utmost flexibility, full scale deployment and management of integrated tool chains, through the entire lifecycle.

The objective of EMC² is to establish Multi-Core technology in all relevant Embedded Systems domains. EMC² is a project of 99 partners of embedded systems industry and research institutions from 19 European countries with an effort of about 800 person years and a total budget of about 100 million Euro.

EMC² has been the project that has unified the concepts of SC and mission-critical systems to emerging distributed technologies such as Internet of Things, Smart Grid and WAMS. It is in such framework where this thesis takes place. Chapter 5 presents and analyzes the work carried out for EMC², focusing on timing stability, scalability and reliability. Chapter 6 represents the integration of the deployed methods, techniques and technologies forming a real Smart Grid scenario.

2.3.3 *SKA*

The Square Kilometre Array (SKA) project is intended to be the largest and most sensitive radio telescope in the World. Thousands of linked radio wave receptors will be located in Australia and Southern Africa. The combination of signals from the an-

tennas in each region create a telescope with a collecting area equivalent to a dish with an area of about one square kilometer.

SKA intends to address fundamental unanswered questions about the Universe, such as the creation of stars and galaxies after the Big Bang, their evolution, the role of magnetism in the cosmos, the nature of gravity, and the search for life beyond Earth.

SKA is a global science and engineering project led by the SKA Organization, a company located at Jodrell Bank Observatory, Manchester, UK.

SKA requires of accurate timing references and dissemination systems to synchronize all the antennas and acquisition devices in all locations with the same notion of time, in order to build a precise image of the sky. Moreover, the reliability and robustness of the timing system is also one of the requirements of SKA, being perfectly aligned to the basis of this doctoral thesis.

2.3.4 *AMIGA6*

AMIGA6: Gas in and around galaxies. Preparation for SKA science and contribution to the design of the SKA data flow builds on the results of previous AMIGA projects and is part of the preparatory work being performed by AMIGA for SKA scientific exploitation. The HI life-cycles both in isolated galaxies and dense groups are still poorly understood since their low column densities are only reachable by the SKA. The main aims are: to refine models of cold gas accretion using isolated galaxies, and to analyze the role of HI tidal removal in the suppressed SF in Hickson Compact Groups.

As preparation for the challenge that SKA data exploitation will constitute, AMIGA6 complements fundamental science with applied research in 3 SKA working packages, so contributing to the Big-Data SKA consortia. AMIGA6 is lead by the coordinator of the Spanish participation in the SKA, and gathers all Spanish groups involved in the data flow of the SKA: Signal and Data Transport (SaDT), Central Signal Processor (CSP), and Science Data Processor (SDP).

2.4 METHODS AND TOOLS

The methodology used has mainly been experimental starting with a first stage of theoretical analysis.

An experimental methodology was developed to characterize WR networks in order to determine the fundamental parameters related to the technology. Among others, the ones associated to

the measurement of the determinism of the system (latency and jitter of the internal clock) together with the network bandwidth during high-load scenarios. The output of this stage is composed of the definition of system models, their analysis and possible enhancements.

It was also necessary to develop real implementations based on the proposed improvements extracted from system model analysis carried out in the first phase of the thesis. To this end, a first theoretical analysis of the system was necessary to determine the constraints of the experiments that had to be realized. Added to this, development guides for safety applications (avionics, automotive and industry) have been part of the reference bibliography. These analysis ended in the most important aspects regarding network topologies, measurement techniques, multi-timing sources, and the considerations that must be taken into account to develop experiments.

With respect to the development tasks, different programming languages have been used depending on the implementation abstraction level. For software development, SystemC[7] has been used for the first concept design of critical applications. The main programming language for the rest of software developments has been C.

These C programs have been compiled for different architectures along the development of this thesis: soft processors inside FPGAs (MicroBlaze and LM32) and Advanced RISC Machine (ARM) processors. In many occasions, tool-chain environments such as Buildroot[8] have been used, specially for the White Rabbit Switch (WRS). Ubuntu 14.04 LTS has been the main Operating System (OS) used.

The main development platforms used have been the Avionics Computing Platform (ACP), the WRS[9] and the WR-LEN[10]. They include Xilinx FPGAs (Virtex 6 and Zync SoC processors), so that the main programming language has been VHDL. Motivated by this, two Integrated Development Environments (IDEs) have also being part of these developments, Xilinx ISE Design Suite[11] for Virtex-6 designs, and Vivado[12] for Zync.

Measuring instrumentation is mainly composed of oscilloscopes and counters. Coarse measurements have been performed

7 http://www.accellera.org/downloads/standards/systemc
8 https://buildroot.org/
9 http://sevensols.com/index.php/products/white-rabbit-switch/
10 http://sevensols.com/index.php/products/wr-len/
11 https://www.xilinx.com/products/design-tools/ise-design-suite.html
12 https://www.xilinx.com/products/design-tools/vivado.html

using a PicoScope 3000 Series[13], and fine measures have been done with a Tektronix FCA3000 Counter[14].

2.5 ORGANIZATION OF CHAPTERS

Besides this introductory Chapter 2, which starts with a general overview of topics addressed in this thesis and its established scientific objectives, the organization of this doctoral thesis continues in Chapter 3 with an introduction to the state-of-the-art of critical distributed control systems. It includes the evolution of mixed-criticality systems from single-core to multi-core and its challenges, the main concepts of distributed control systems and their evolution to Smart Grid, followed by the relevance of the time distribution in these networks, and ending with a market survey regarding reliable timing devices for the industrial domain.

Chapter 4 exposes the methods to increase reliability in critical applications using multi-core platforms in terms of hardware, software and inter-core communication.

Chapter 5 presents a study of dependability issues for time and data distribution in industrial applications. This Chapter includes different mechanisms to increase industrial compatibility, timing reliability and data fault tolerance.

Chapter 6 proposes a real Smart Grid scenario as proof of concept for the developed reliable mechanisms presented in Chapters 4 and 5.

Finally, Chapter 7 summarizes the main contributions of our research and suggests several lines of future work.

13 https://www.picotech.com/products/oscilloscope
14 https://www.tek.com/datasheet/fca3000-and-fca3100-series

3

STATE OF THE ART

It's dangerous to go alone, take this!"

— Old Man, The Legend of Zelda.

INDEX

This Chapter presents the state-of-the-art (SoA) from which this thesis begins. It is mainly composed of three interconnected domains: time and frequency distribution, control systems and real-time critical applications. Section 3.1 unifies these three concepts, focusing on the need to coordinate the execution of distributed processes in critical time-aware applications and the mechanisms, techniques and protocols that must be used to guarantee system reliability. Section 3.2 presents the first step in the evolution of critical systems: the utilization of multi-core architectures instead of single-core, leading to coordination and isolation issues that must be addressed. Section 3.3 depicts the transition of DCS from the classical power grid concept, to its modern evolution known as Smart Grid, with particular focus on the need to synchronize the applications and inter-process communications that are taking place on the grid. Section 3.4 shows the different

protocols and schemes that are used in Smart Grid to synchronize devices, such as GNSS, NTP, PTP, etc. In addition to this, a full description of WR, main technology used within this thesis, can be found in this Section. This Section concludes with the description of different network mechanisms to guarantee timing reliability and fault tolerance avoidance.

Finally, Section 3.5 presents a market survey on different network devices capable of providing reliability features for timing and data distribution in industrial networks. Furthermore, it includes a full description of the main development platform used for this work, the WRS.

3.1 INTRODUCTION TO DEPENDABLE DISTRIBUTED PROCESSING

Integration and monitoring processes have evolved significantly during the last years, giving rise to the utilization of different components among these systems, starting with single chips, System on Chips (SoC), Network on Chips (NoC), and ending with distributed systems such as Wide Area Monitoring systems (WAMS) and 5^{th}-Generation Wireless Systems (5G). The basis of this evolution resides in the capacity of parallelizing and distributing processes in a coordinated and efficient manner. This fact requires the description of synchronous and asynchronous communication mechanisms that guarantee the best computation performance by sharing a common time reference. Added to this, the interaction with the environment and the distributed processing has added the need to synchronize the entire system with an accurate time reference.

These problems increase when real-time SC systems come on the scene. In this case, specific methodologies have to be developed, bringing new concepts, technologies and methods such as, Real-time Operating Systems (RTOS), isolation between critical and non-critical components, the utilization of hypervisors and the implementation of different communication buses like Time-Sensitive Networking (TSN). All this together leads to the deployment of a global dependable system that demands a variety of techniques that must be integrated in each level of the development chain. It is worth mentioning that the framework described is present in a large amount of European research projects under different funding programs and associations such as, ECSEL Joint Undertaking [13], ARTEMIS Industry Association [14] and Horizon 2020 [15].

The rest of the Sections in this Chapter describe these technologies, methods and concepts.

3.2 SAFETY-CRITICAL SYSTEMS: FROM SINGLE TO MULTI-CORE ARCHITECTURES

The simplification of the development process in SC systems has been the center of attention of many companies and research centers working on avionics, automotive and industrial applications. Safety requirements impose significant time and cost overhead over conventional design procedures which many companies strive to minimize. In the past, processor trends were dominated by the increase of complex feature sets, higher clock speeds, growing thermal envelopes and power dissipation (super-scalar micro-controllers). This solution is no longer scaling because of hardware limitations and power/thermal dissipation issues caused by frozen clock speeds. However, markets and applications demand performance increase, safety and low energy consumption, which is closely related to critical embedded systems [16].

Not many years ago, single-core was the most common hardware architecture for critical embedded systems but instead, there are many advantages of using multi-core/multi-processor systems: the single-core obsolescence and the lately business philosophy which aims to the increase of performance and the reduction of costs in the development process of these systems [16]. Additionally, the introduction of multi-core processing platforms into this kind of systems poses an important challenge at different levels specially related to the management of shared resources. These issues mainly affect system-level scheduling [17], interferences between the low-critical and high-critical tasks [18] as well as the architecture communication, which need to be solved at hardware and software levels to properly guarantee that multiple processes running on the multi-core can fulfill the real-time constraints required by the SC applications without affecting each other [19].

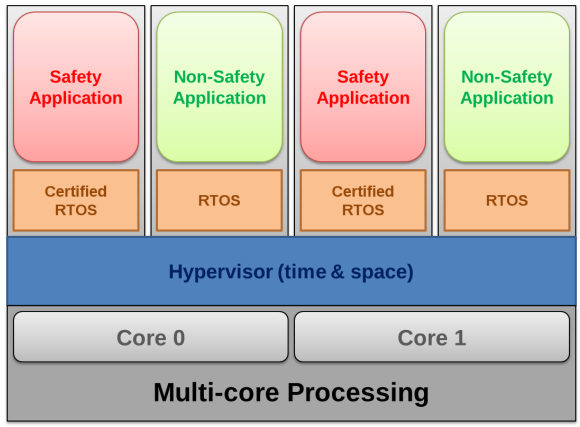

Figure 3.1
Mixed criticality application design for a multi-core architecture. Safety applications run on a certified RTOS while non-safety ones run on a standard RTOS. The execution and resources requested by the different RTOS are managed by a hypervisor. The hypervisor isolates processes and resources in time and/or space at the same time that it dispatches them between the two cores, thus guaranteeing the integrity of the safety application.

A trend solution lately proposed has been the utilization of mixed-criticality architectures on the same processor, in which different levels of reliability and criticality can be achieved [20, 21] as Fig. 3.1 depicts. While traditionally mixed-criticality systems were based on spatial separation of SC and non-safety-critical (NSC) tasks using different hardware processors, current trends aim towards using different mechanisms as temporal isolation [22] in order to allow sharing hardware resources. A goal is to find mechanisms that provide and prove isolation between NSC and SC parts. This allows the development of NSC applications of mixed-criticality systems in a simple and less expensive way and, additionally, to upgrade NSC parts of the system without requiring a re-certification process [23]. These problems become even more crucial in critical systems which require a strong validation and verification process.

Different mechanisms are available to increase and achieve safety properties, such as the utilization of identical or diverse redundancy concepts for both hardware and software [24, 25]. *Diverse redundancy* refers to using two or even more different subsystems, which are built with different components, algorithms, electronics, design methodology, etc., to perform the same task. One of the benefits derived from the utilization of diverse redundancy is the increased capability to reduce common mode and systematic failures, such as those caused by design flaws.

Software diversity was called into question for not being able to prevent system errors [26]. For this reason, the development of hardware-software architectures combining diversity techni-

ques reduces the errors that can be correlated from software-like diversity solutions [27]. This technique represents an effective defense against hidden dangerous faults, thus decreasing the probability of system failure in a safe state (Safe Failure Fraction, SFF). Logic solver technologies [28], which use internal diverse redundancy have been developed for applications up to Safety Integrity Level (SIL) 3. This mechanism is one of the methodologies recommended by IEC 61508 and IEC 61511 standards in order to increase safety integrity of programmable electronic systems [5].

The utilization of multi-core architectures involves the description of new requirements related to isolation and partitioning mechanisms that should be complied to achieve safety. First, a reliable communication connection (Core-to-Core communication) needs to be implemented to avoid losing the features that multi-core provides. In addition, a software system scheduling must be performed since different resources are shared by different processors, leading to system collisions and failures. As well as design methodologies, development tools should be adapted to these architectures since parallelism features and the possibility of concurrent processes are key points for multi-processors. The requirements and recommendations proposed for the development of avionics, industrial and automotive multi-core applications are detailed in the safety standards IEC 61508 [5], DO-178/C [29] , DO-254 [30] and ISO 26262 [31]. These documents are used to ease the certification process of any system deployed for these domains. Next sections describe the certification process, the concepts related to this complex process and a brief description of these standards.

3.2.1 *The certification process and concepts*

IEC 61508 defines the V-model for software safety integrity and the development life-cycle of SC systems. The design and development process starts with the specification of the software safety requirements and ends with the validation of the developed software on the software validation testing. Fig. 3.2 depicts this V-model.

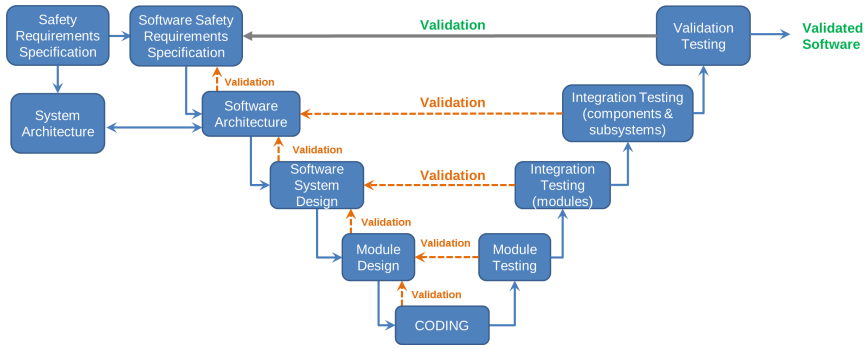

Figure 3.2
V-model for software safety integrity and the development life-cyle of SC applications defined by IEC 61508.

In order to understand the certification process and the entailed process that will be used among this thesis, some terminology must be introduced. These concepts and definitions have been extracted from the training course *Functional Safety Acc. IEC 61508 / ISO 26262* [32] provided by TÜV SÜD[1] during the course of the EMC^2 EU project.

3.2.1.1 *Hazard analysis techniques*

A hazard analysis is used as the first step to asses risk. The results of this process identify the different types of hazard. A hazard is defined as a condition or circumstance that could lead to an unplanned or undesirable event.

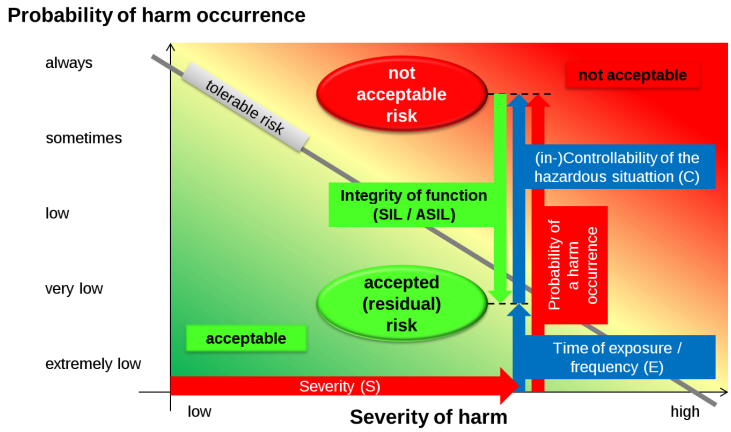

Figure 3.3
Hazard and risk analysis for industrial systems. Figure extracted from [32].

Safety and reliability risks can be eliminated in some cases, but normally, a degree of risk must be accepted, considering the

1 https://www.tuv-sud.es/

consequences and the probability of occurrence of these risks. Assessment of risk is carried out by the combination of the severity of a consequence with the likelihood of occurrence into a risk matrix (Fig. 3.1). Those risks that remain in the *unacceptable* category must be mitigated. Fig. 3.3 represents how a hazard and risk analysis is performed for the industrial domain, where the acceptance of a risk depends on its severity, the controllability of the situation (C), the time of exposure (E) to this risk and the probability of harm occurrence.

The results of the hazard analysis are summarized into the risk matrix. An example of risk matrix is shown below.

Table 3.1
Risk Matrix: Quality Management (QM) vs Safety Integrity Level (SIL)

	Probability: E x C				
Severity	0.0001	0.001	0.01	0.1	1
S0 (no injuries)	QM	QM	QM	QM	QM
S1 (light and moderate injuries)	QM	QM	QM	SIL 1	SIL 2
S2 (severe injuries)	QM	QM	SIL 1	SIL 2	SIL 3
S3 (fatal injuries)	QM	SIL 1	SIL 2	SIL 3	SIL 3

For the accomplishment of this thesis, two hazard techniques have been used: Fault Tree Analysis (FTA) and Hardware Analysis (FMEDA).

FAULT TREE ANALYSIS (FTA)

Fault Tree Analysis (FTA) is a top-down, deductive analysis which visually depicts a failure path or failure chain. FTA follows Boolean logic concept, thus allowing the creation of True / False statements. These statements form a logic diagram of failure. Events are arranged in sequences of series relationships (OR) or parallel relationships (AND). Results for each event are presented in a tree-like diagram as Fig. 3.4 depicts, using logic symbols to show dependencies among events.

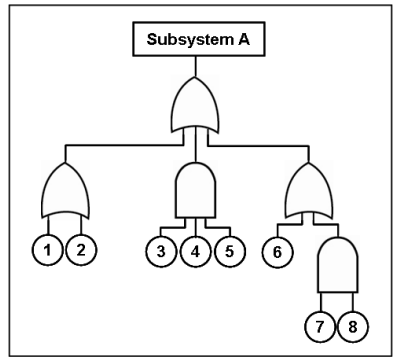

Figure 3.4
Fault tree schema example.

Events may be related to mechanical components, software and electronics used in the design of the product.

HARDWARE ANALYSIS (FMEDA)

FMEDA is the proof of the Safe Failure Fraction (SFF) and the Diagnostic Coverage (DC), the calculation of the probability of dangerous failures and its fault tolerance (HFT). It is a systematic analysis technique to obtain subsystem level failure rates, failure modes and diagnostic capability. It considers all component designs, the functionality of each component, the failure modes of each component, the effect of each component failure mode on the product functionality, the ability to any automatic diagnostics to detect the failure, the design strength and the operational profile (environmental stress factors).

Given a component database calibrated with field failure data that is reasonably accurate, the method can predict product level failure rate and failure mode data for a given application.

HFT is the ability of a hardware unit to go on executing a demanded function while faults or errors exist. When HFT is defined as N, it means that N + 1 hardware faults cause the loss of the corresponding safety function.

Example: Redundant channels of a control system with a mutual control: HFT N = 1.

$$\text{HFT} \quad = N + 1 \tag{3.1}$$

SFF regards to the relation between the safe failures, the dangerous detected failures and the dangerous undetected failures (3.2).

Safe failures ($\sum \lambda s$) represent those failures that do not affect the functioning of the system. Dangerous detected failures

($\sum \lambda DD$) describe those failures that, in spite of being harmful for the system, they are known. Finally, dangerous undetected failures ($\sum \lambda DU$) are those that are harmful for the system but they have not been detected.

$$SFF = \frac{\sum \lambda s + \sum \lambda DD}{\sum \lambda s + \sum \lambda DD + \sum \lambda DU} \tag{3.2}$$

DC (3.3) is used to classify the diagnostic coverage percentage of the safety function from relation between the sum of all dangerous failures (detected and undetected) and the ones detected.

$$DC = \frac{\sum \lambda DD}{\sum \lambda DD + \sum \lambda DU} \tag{3.3}$$

The DC results can be classified as *none* for measures below 60 %, *low* for those between 60 % and 90 %, *medium* for the 90-99 % range, and finally *high* for results above 99 %.

Hereafter, certification standards IEC 61508, DO-178/C, DO-254 and ISO 26262 are described.

3.2.2 *IEC 61508*

For the industry domain, IEC 61508 describes requirements to prevent failures caused by hazardous events and to control failures by ensuring safety, even when faults are present. Additionally, this standard provides requirements for product's overall safety life-cycle.

IEC 61508 specifies four discrete Safety Integrity levels (SIL) of safety performance for a safety function. SIL 1 is the lowest level of safety integrity, and SIL 4 is the highest level. Requirements to achieve safety integrity at the higher levels are more meticulous than at lower levels. Table 3.2 summarizes all SIL levels for IEC 61508 and their correspondent probability of failure associated to each SIL level.

Table 3.2
SIL probability of failure.

SIL	average probability of failure on demand	Probability of dangerous failure per hour
1	$\geqslant 10^{-2}$ to $< 10^{-1}$	$\geqslant 10^{-6}$ to $< 10^{-5}$
2	$\geqslant 10^{-3}$ to $< 10^{-2}$	$\geqslant 10^{-7}$ to $< 10^{-6}$
3	$\geqslant 10^{-4}$ to $< 10^{-3}$	$\geqslant 10^{-8}$ to $< 10^{-7}$ (1 dangerous failure in 1140 years)
4	$\geqslant 10^{-5}$ to $< 10^{-4}$	$\geqslant 10^{-9}$ to $< 10^{-8}$

SIL requirements for hardware safety integrity are based on a probabilistic analysis of the device. To achieve a concrete SIL,

the dangerous failure probability must be less than the one specified, and also it must be greater than the specified safe failure fraction. These failure probabilities are calculated for instance by performing a Failure Mode and Effects Analysis (FMEA) or any of its variations. The actual targets required vary depending on the likelihood of a demand, the complexity of the device(s), and the types of redundancy used.

3.2.3 *DO-178/C and DO-254*

Concerning avionics, DO-178B/C is a document used by the US Federal Aviation Administration (FAA) to determine the conditions in which software, that is required to be certified, is able to run, safely and reliable, in an airborne environment.

DO-254 standard is involved in the compliance for the design of complex electronic hardware of airborne systems. Complex electronic hardware includes devices like Field Programmable Gate Arrays (FPGAs) and Programmable Logic Devices (PLDs). This standard specifies the requirements for both design assurance and certification processes. Hardware design verification and validation need to be accomplished independently, which means that hardware designers should ensure that the design fulfills the defined system functionality and the verification team should verify that all of the derived requirements from the standard are met.

As SIL for the industrial field, the Design Assurance Level (DAL) is determined from the safety assessment process and hazard analysis by examining the effects of a failure condition in the system. The failure conditions are categorized by their effects on the aircraft, crew, and passengers from DAL A (highest level) to DAL E (lowest level which has no impact on safety). Table 3.3 presents DAL levels and their failure condition ranges.

Table 3.3
DAL Failure condition ranges.

Level	Failure condition	Failure Rate
A	Catastrophic	10^{-9}/h
B	Hazardous	10^{-7}/h
C	Major	10^{-5}/h
D	Minor	10^{-3}/h
E	No Effect	n/a

DO-254 standard is involved in the compliance for the design of complex electronic hardware in airborne systems. Complex electronic hardware includes devices like FPGAs, PLDs and ASICs. The hardware design and hardware verification need to be done with independence, which means that the hardware designers should work to ensure the design meets the defined requirement and the verification team should create a test that verifies all of the derived requirements. Rather than specifying how to implement the standard or which test should be completed, it specifies the requirements for a process of design assurance and certification.

3.2.4 *ISO 26262*

ISO 26262, which derives from IEC 61508, defines a framework and an application model as well as the activities, methods and results to guarantee safety and security for the automotive domain. It defines functional safety for automotive equipment applicable throughout the lifecycle of all automotive electronic and electrical safety-related systems.

Table 3.4
Cross-domain mapping of integrity levels for the industry, automotive and avionics domains.

Domain	Integrity Level	Domain-specific safety levels				
IEC 61508 (General)	SIL (Safety Integrity Level)	-	SIL-1	SIL-2	SIL-3	SIL-4
ISO 26262 (Automotive)	ASIL (Automotive Integrity Level)	-	ASIL-A	ASIL-B/C	ASIL-D	-
DO-178C/256 (Aviation)	DAL (Design Assurance Level)	DAL-E	DAL-D	DAL-C	DAL-B	DAL-A

The safety assessment and hazard analysis process for this standard involve the definition of the Automotive Safety Integrity Level (ASIL) for the safety function. Since ISO 26262 derives from IEC 61508, there is a correlation between SIL and ASIL levels. Table 3.4 presents the correspondence between SIL, ASIL and DAL.

Finally, dependable single and multi-core devices form part of largest distributed control systems (DCS) as leaf-nodes to perform actions, data acquisition and event control. Down below DCSs are introduced together with their evolution during the last years.

3.3 DISTRIBUTED CONTROL SYSTEMS

A DCS is a control system applied to complex industrial processes in large industries such as petrochemical, paper, metallur-

gical, generation plants, treatment plants of water, incinerators, etc.

DCSs work with a single integrated database for all signals, variables, graphical objects, alarms and system events. The DCS engineering tool programs the system and operates in a centralized way to develop the logic of its controllers. From this engineering position, the programs are loaded transparently to the system's computers. The figure below represents a typical DCS.

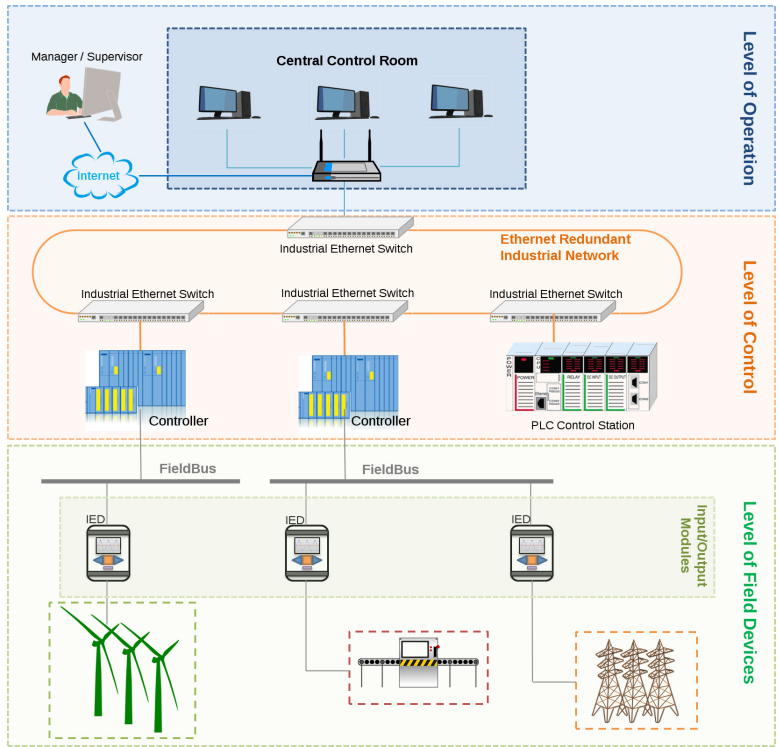

Figure 3.5
Industrial distributed control system example composed of the level of operation (blue shape), the level of control (orange shape) and the level of field devices (green shape). The upper communication layers include Ethernet protocols using, for example, Gigabit Ethernet over optical fiber while field devices are interconnected through field buses.

A DCS addresses the complexity of industrial processes by dividing its scope into four functional levels.

- **Level of Operation**. This level regards to the interaction of the system with the operators of the plant. It includes the computer systems for the monitoring of the information acquisition process in real time. It also stores the database, which is transformed into historical data for later analysis. This level also manages the exchange of information with other maintenance systems.

- **Level of Control**. Instead of centralizing all DCS functions and responsibilities in a single point, several parts of the entire process are assigned to different local distributed controllers. The controllers are connected to each other and to the operation stations through communication networks.
- **Level of Input / Output modules**. The input/output modules for wired signals are distributed by the installation (decentralized periphery). These input/output modules communicate with the controllers through specific protocols or fieldbus to ensure communication between controller and periphery in a minimum time (milliseconds). The most widespread field bus in Europe is the ProfiBus while the fieldbus Foundation is in America.
- **Level of Field Devices**. Since 2000, there has been a growing need to integrate directly the instruments and actuators in the field buses of the DCS, so that these devices feel as a natural extension of the previous level. These devices allow additional functions such as maintenance managing or configuration remotely from the operation level. The instruments of this level must be compatible with the chosen fieldbus.

All DCS must meet robustness and reliability characteristics at all levels described above: redundant computer equipment, redundant controllers, communication networks and redundant buses, redundant input/output modules, etc. Thanks to this reliability mechanisms, the availability factor of the services within the system can be guaranteed close to a 99.9999 % factor, higher than conventional control systems. [33] provides guidance on how to secure Industrial Control Systems (ICS), including Supervisory Control and Data Acquisition (SCADA) systems, DCS, and other control system configurations such as Programmable Logic Controllers (PLC), while addressing their unique performance, reliability, and safety requirements. The document provides an overview of ICS and typical system topologies, identifies typical threats and vulnerabilities to these systems, and provides recommended security countermeasures to mitigate the associated risks.

3.3.1 *From power to the Smart Grid. The role of a global time reference.*

In a classical power grid, a fixed price is charged to energy users. However, the cost of energy is higher during the daily peak load

operation. The classical power system operation has no control over the loads except in an emergency situation when a portion of the loads can be dropped as needed to balance the power grid generation with its loads. Therefore, most elements of the grid are only used for a short time during the peak power demand and they remain idle during the rest of the daily operation [34].

In order to offer an efficient service, the power grid is turning the centralized energy grid into an intelligent and interactive network changing the energy value chain. The Smart Grid is a complex infrastructure composed of many different systems and electronic devices. The complexity of Smart Grid is continuously increasing due to the arrival of new actors such as Distributed Energy Resources (DER), the electric vehicle [35], microgrids and prosumer concept [36]. Additionally, due to the critical nature of these infrastructures, providing essential services such as electrical energy, the Smart Grid must be safe, secure and reliable [1, 2, 37].

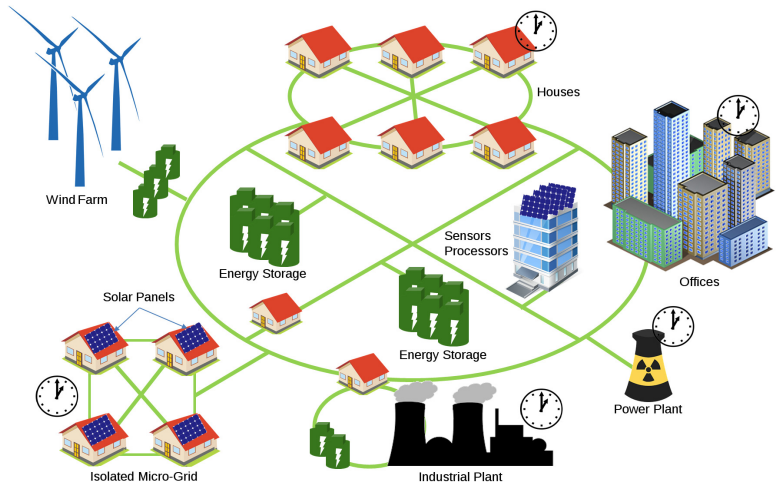

Figure 3.6
Smart Grid architecture including smart houses, buildings, power and industrial plants, and also renewable energies. This architecture takes into account the need for providing a precise timing service to the elements that compose the grid.

SAS systems such as SCADA (Supervisory Control and Data Acquisition), Remote Terminal Units (RTU) and Intelligent Electronic Devices (IED) are considered as core components of the Smart Grid at transmission and distribution levels (Fig. 3.6). Substation automation is a mission-critical task and, in addition, those systems are working under real-time conditions. Substation Automation Systems (SAS) provide reliable bedrock for future Smart Grid developments in electric facilities [38].

Due to the distributed and discrete nature of the different Smart Grid devices, the proper utilization of a network infrastructure is fundamental to provide a dependable communication channel for all the different actuators and sensors. On one hand, data communication allows sensing the status of the Grid properly and makes possible the optimization of the different operation parameters (towards an efficient energy consumption and distribution achievement). On the other hand, it is critical to guarantee the presence of communication features to enable failure contention (using for instance switches and breakers to isolate faults) or providing a global time reference for forensic analysis of the Smart Grid (requiring a network synchronization mechanism to work over large areas on a deterministic way). Both features require a dependable network infrastructure including reliable equipment, redundant network topologies with zero-time recovery, dependable time references and secure communications that are able to guarantee the proper Grid operation.

In this regard, IEC 61850 [6] suggests the implementation of redundancy protocols within the services offered in the network focusing in substation automation. There are several protocols suggested but SmartGrid and engineering networks use mainly two: the Parallel Redundancy Protocol (PRP) and the High-availability Seamless Redundancy (HSR) protocol. Both PRP and HSR were standardized by the International Electrotechnical Commission in Geneva, as IEC 62439-3 Clause 3 and 5 respectively [7]. Widely used in Power grid, PRP (3.8) and HSR (Fig. 5.9) are redundancy protocol for Ethernet (standardized as IEEE 802.3) networks. The former is used in tree topologies with separate LANs, and the latter in ring topologies and expandable to mesh topologies. Both of them provide zero-time recovery in case of failure of one component. It is suited for applications that demand high-availability and very short switch-over time, for example, protection for electrical substation automation. On Smart Grid the recovery time of commonly used protocols like the Rapid Spanning Tree Protocol (RSTP) is not acceptable and therefore solutions implemented at the physical network level like PRP and HSR are necessary. Both of them presents a profile for time distribution in [7] that specifies how time should be distributed using these protocols. A brief description of this PTP profile can be found in Section 3.4.6 of this Chapter.

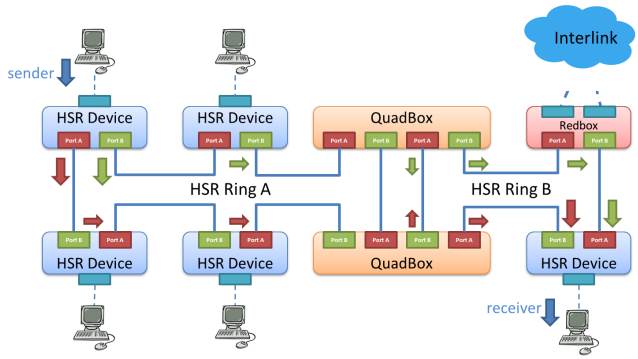

Figure 3.7
HSR network example.

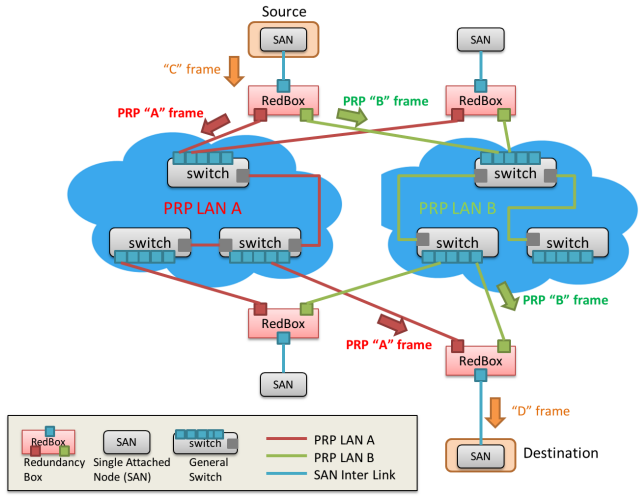

Figure 3.8
PRP network example.

In terms of synchronization requirements for Smart Grid, timing information is considered critical in these kind of environments since a proper synchronized communication must be guaranteed between the different elements that conform the grid. For this reason, high accurate, reliable and scalable technologies are required to distribute time for next generation SmartGrid networks. Range varies from one microsecond to hundreds of nanoseconds, and recently to tens of nanoseconds [39], within and across substations utilities in order to advance capabilities in real-time measurement and control. The proliferation of widely deployed smart sensors for Wide Area Monitoring Protection and Control (WAMPAC), distribution and energy management systems, along with the increasing need for fault detection and location as well as maintaining system stability in real time us-

ing spatio-temporal and temporal-frequency analyses, all require precision timing [40]. Table 3.5 summarizes several Smart Grid applications and their time accuracy requirements. It has been extracted from [40] and updated with [39, 41, 42].

Table 3.5
Wide area precision time requirements in current power and Smart Grid systems

Application	Time Accuracy Requirement
Digital Fault Recorder	1 ms
Substation Local Area Networks (IEC 61850 GOOSE)	100 μ to 1 ms
Line Differential Relays	10 to 20 μs
Substation Local Area Networks (IEC 61850 Sample Values)	1 μs
Wide Area Protection	Better than 1μs
Frequency Event Detection	Better than 1μs
Anti-Islanding	Better than 1μs
Droop Control	Better than 1μs
Wide Area Power Oscillation Damping	Better than 1μs
Traveling Wave Fault Detection and Location	100 to 500 ns
Synchrometrology (synchrophasors)	Better than 10 ns

The main standard timing sources used in power and Smart Grid are GPS for wide area time synchronization, NTP for utilities with accessible and available network communication lines, where some of them that require more accuracy integrates PTP. Within the substation the methods of time propagation include IRIG-B time signal [43] with time quality and leap second specifications based on IEEE C37.118 [44], NTP and PTP from a GNSS source [40]. Section 3.4 of this Chapter describes these timing technologies, their accuracies, advantages and disadvantages.

Another important element to be considered in the Smart Grid is safety, closely related with the previous features. The rapid growth of Intelligent Electronic Devices (IED's), Remote Terminal Units (RTU) and other components within electric networks is allowing Utilities to manage increased demand from users across the globe. However, the new technologies demand that safety standards be updated and modernized. Industry standards such as IEC 61508 provide a roadmap for organizations that wish to deploy and support the new technologies of the Smart Grid. This standard is widely used by electronic device manufacturers and suppliers when any part of the safety function contains an electrical, electronic, or programmable electronic component and where application sector international standards do not exist. The IEC 61508 standard specifies the risk assessment and the

measures to be taken in the design of safety functions for the avoidance and control of faults. In fact, IEC 61508 provides a complete safety life cycle that accounts for possible risk of physical injury and damage to the environment. Acceptable levels of risk are determined and procedures for residual risk management over time are established. In order to achieve the necessary SIL, the standard requires a proof of residual risk, which is based on the probability of dangerous failure. In this sense, the demonstrator will allow testing the impact of the integration of these intelligent systems in terms of safety and system latency in a network with high accuracy time synchronization. These concepts will be addressed in Chapters 4 and 6.

3.4 TIME DISTRIBUTION IN SMART GRID

Distributed systems and more precisely Smart Grid, involve a large number of interconnected nodes using different communication mediums for several purposes, such as data acquisition, parallel computing, control, event management, etc. These devices must exchange data by accessing the medium, and share the same notion of time in order to timestamp accurately the moment data is sent and/or received.

In order to solve this issue, several time synchronization technologies appeared into scene for distributed systems. Next Sections describe some of them in detail.

3.4.1 *GNSS*

GNSS[2] is a combination of navigation systems such as the US GPS[3], the Russian *transliteration Globalnaya navigatsionnaya sputnikovaya sistema*[4] (GLONASS) and Galileo[5]. These systems are capable of providing time and geo positioning anytime, anywhere. GNSS is a relatively innovative concept, since its point of departure is the US GPS system in the 70's. GPS was used exclusively by the American military and, in spite of its world coverage, it was far away of becoming the current *Global* time reference. In other words, GPS was under a strict control from the American Department of Defense. Nevertheless, by the end of 90's GPS starts to be used in commercial and civil applications after reach-

2 https://www.gsa.europa.eu/
3 https://www.gps.gov/
4 https://www.glonass-iac.ru/en/
5 http://galileognss.eu/

ing agreements among the US Government and various countries around the world.

Satellite navigation systems are based on the computation of the position on the surface of the Earth by measuring distances between a minimum of three satellites. Another one provides the altitude. The precision of the of the measurements is determined by the accuracy of the final location. In other words, a receiver captures the synchronization signal emitted by the satellites that contain the position of the satellite and the exact time in which it was transmitted. The satellite's position is transmitted in data messages that are used as the reference of the synchronization.

The precision of the position depends on the accuracy of the timing information. Only atomic clocks provide this required precision, in the order of nanoseconds. To this end, satellites use an atomic clock in order to be synchronized to all the satellites of the constellation. The receptor compares the diffusion time, which is encoded in the transmission together with the reception time measures by an internal clock, so that it can measure the *time-of-fly* of the signal from the satellite.

Nowadays, the GPS, GLONASS and Galileo are the only ones that belong to the GNSS concept. Other navigation systems that may be included for the civil aviation as part as GNSS are the Chinese BeiDou[6], Compasss[7], the Japanese *Quasi-Zenith Satellite System*[8] (QZSS) and the *Indian Regional Navigation Satellite System*[9] (IRNSS).

3.4.1.1 *Galileo satellite navigation project*

Galileo is the GNSS that is currently being created by the European Union (EU) through the European Space Agency (ESA) and the European GNSS Agency (GSA). This system provides the EU with an independent technology from the US GPS and the Russian GLONASS. In contrast to these ones, Galileo is intended for civilian use and it is planned to be fully functional for 2020.

Galileo intends to offer real-time positioning with a precision of 1 meter free of charge, and up to 1 cm for for the subscription model (payment required). A summary of the services that it will provide is described below:

- **Open Service**: Available without charge for use. Simple timing and positioning down to 1 meter.

6 http://en.beidou.gov.cn/
7 http://www.compasss.org/
8 http://qzss.go.jp/en/
9 https://www.isro.gov.in/

- **Commercial service (encrypted)**: Guarantees accuracies up to 1 cm with charge fees.
- **Safety of life applications**: Open service focused on transport applications where human lives might be injured in case the navigation system fails.
- **Public regulated service**: Encrypted robust and supervised service for governmental applications. This service will be used by customs and police departments.
- **Search and rescue service**: Real-time rescue messages reception from any point on Earth. Precise alert location (few meters instead of the current 5 km).

Regarding GNSS and GPS time accuracy, the results published in [45] states that the averaged RMS values of all GPS satellites are 0.034 ns. The GLONASS clock accuracy is worse than GPS with a RMS value of 0.066 ns. The Galileo and BeiDou clocks can achieve comparable accuracy to GLONASS clocks, which are 0.066 and 0.065 ns, respectively. The relatively large clock overlaps for Galileo may be caused by the sparse amount of available tracking data. Although the GEO orbits are worse than other satellite types, the corresponding clock accuracy is also generally better than 0.1 ns as the satellite clocks are mainly correlated with the radial orbit component. However, such precise results are not transferable to users, presenting accuracies around 100 ns and, at best, around 10 ns after additional rigorous calibration procedures. In addition, the utilization of GPS for time synchronization may display a number of vulnerabilities, which might jeopardize the reliability of satellite navigation systems.

3.4.1.2 *Satellite positioning systems vulnerabilities*

The most notable vulnerability of GNSS is the possibility of signal interference. Over the past years, spoofing and jamming media attacks and their consequences have been echoed in a variety of media, such as the *The Scary Threat to GPS That Could Paralyze U.S. Businesses*[10], *GPS Disruptions: Efforts to Assess Risks to Critical Infrastructure and Coordinate Agency Actions Should Be Enhanced*[11] and the one presenting the greatest impact: *What Hap-*

10 https://www.kiplinger.com/article/business/T057-C000-S010-thethreat-to-gps-that-could-paralyze-businesses.html

11 https://www.gao.gov/products/GAO-14-15

pens If GPS Fails?, published by The Atlantic[12], The New Yorker[13] and Forbes[14], among others.

There are several possible sources of interference to GNSS, both within and outside the band, but particularly by point-to-point terrestrial microwave links allowed by several states (1559-1610 MHz). GNSS signals are vulnerable due to the relatively low power of the received signal because they come from satellites and each signal covers a significantly large fraction of the Earth's surface. It is worth to mention that a jamming device is within the reach of all, costing below 30€[46].

The probability and operational consequences of this interference vary with the medium. It is not considered a major risk as long as states exercise an adequate control and protection of the electromagnetic spectrum, both for existing and new frequency allocations. In addition, the introduction of new GNSS signals into new frequencies will ensure that unintended interference does not lead to the complete loss of service, even if it experiences some performance deterioration.

It has been determined that most of the reported GNSS interference cases come from on-board systems and experience with GNSS installations have identified several sources of unintended interference. Portable electronic devices may also cause interference to the GNSS and other navigation systems.

Ground sources of interference currently include mobile and fixed very high frequency communications, point-to-point radio links in the GNSS frequency band, television station harmonics, certain radar systems, mobile satellite communications systems and military systems. Large cities with significant sources of radio frequency (RF) interference, industrial sites, etc., are more prone to involuntary interference than remote regions, where interferences are very unlikely.

Regarding intentional attacks, spoofing is able to corrupt navigation signals so that aircrafts might deviate and follow a false flight path. Although signal simulation interference can theoretically induce an aircraft into navigational errors, it is most likely to be detected by normal procedures. Apart from military or avionics issues, there are common jamming and spoofing attacks that may also be performed to bypass certain laws, such as the avoidance of tracker information for trucks, fishing in restricted zones, etc. In order to solve these issues, ground proximity warn-

12 https://www.theatlantic.com/technology/archive/2016/06/what-happens-if-gps-fails/486824/

13 https://www.newyorker.com/tech/elements/what-would-happen-if-gps-failed

14 https://www.forbes.com/consent/?toURL=https://www.forbes.com/sites/quora/2017/10/20/what-happens-if-the-gps-system-on-a-plane-fails/

ing systems (GPWS) and on-board collision avoidance (ACAS) give additional protection against collisions. Furthermo-re, new satellite signals and constellations will greatly reduce the vulnerability of GNSS. The use of stronger signals and the various frequencies planned for GPS, GLONASS and Galileo will effectively eliminate the risk of unintended interference, as it is very unlikely that a source of such interference will simultaneously affect more than one frequency.

Stronger signals and new GNSS frequencies make it more difficult to intentionally interfere with all GNSS services. In addition, larger satellite constellations, such as IRIDIUM [47], reduce the risk of system failure, operational errors or service interruptions. Administration and a strong system funding are essential for the continued operation of GNSS services and to mitigate the system vulnerabilities mentioned above.

3.4.2 *Inter-range instrumentation group time codes*

IRIG codes are standard formats for transferring timing information. Atomic frequency standards and GPS receivers designed for precision timing are often equipped with an IRIG output. This standard was created by telecommunication working group of the US military. The standards were created by the TeleCommunications Working Group of the U.S. military's Inter-Range Instrumentation Group (IRIG).

The different timecodes defined in the Standard have alphabetic designations. A, B, D, E, G, and H are the standards currently defined by IRIG Standard 200-04. The main difference between codes is their rate, which varies between one pulse per minute and 10,000 pulses per second. The different IRIG formats are shown in Table 3.6.

Table 3.6
IRIG Time Code Formats

Format	Pulse Rate (or Bit Rate)	Index Count Interval
IRIG-A	1000 PPS	1 ms
IRIG-B	100 PPS	10 ms
IRIG-D	1 PPM	1 minute
IRIG-E	10 PPS	100 ms
IRIG-G	10000 PPS	0.1 ms
IRIG-H	1 PPS	1 second

IRIG and more precisely IRIG-B is commonly used in the power industry and Smart Grid, being implemented in data acquisition and control devices, such as RTUs. The utilization of a solution combining GNSS and wired technologies such as IRIG-B has become widely popular in this field during the past years [11].

3.4.3 *Network time protocol*

NTP is an Internet protocol for synchronizing clocks of computer systems through UDP packet routing in networks with variable latency. NTP is one of the oldest internet protocols that are still in use (since before 1985).

NTP uses the UTC time scale, including support for features as second interleaves. NTPv4 offers a synchronization accuracy of 10 ms over the Internet, and can reach up to 200 μs in local area networks under ideal conditions.

In order to get the best NTP performance, it is important to have a standard NTP clock with phase tracking loop implemented in the operating system kernel, instead of just using the intervention of an external NTP daemon: all current versions of GNU Linux and Solaris support this feature.

NTP uses a clock stratum hierarchy system, where layer 1 systems are synchronized with an external clock such as a GPS clock or some atomic clock. NTP stratum 2 systems derive their time from one or more of stratum 1 systems, and so on (note that this is different from the stratum used in telecommunication systems).

The timestamps used by NTP consist of a 32-bit second and a 32-bit fractional part, giving a 2^{32}-second scale (136 years), with a theoretical resolution of 2^{-32} seconds (0.233 ns). Although NTP time scales are rounded every 2^{32} seconds, implementations should disambiguate the NTP time using the approximate time from other sources. This is not a problem in general use since this only requires a time close to a few decades.

There is a less complex form of NTP that does not require storing the information regarding previous communications known as Simple Network Time Protocol or SNTP. It has gained popularity in embedded devices and in applications where high accuracy is not required.

3.4.4 *Precision time protocol*

PTP is a protocol used to synchronize clocks in computer networks. It achieves an accuracy range below 1 µs. For this reason, it has been using lately in control and measurement systems.

PTP was originally defined in the IEEE 1588-2002 *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, published in 2002. Later in 2008, IEEE 1588-2008, also known as PTPv2, improved the protocol accuracy, precision and robustness. The price to pay for this enhancement was the incompatibility with the first version of the protocol. PTP appears into scene to fill a niche not fully served by NTP or GPS.

PTPv2 describe a hierarchical master-slave architecture for clock distribution. Following this architecture, four types of clocks can be defined:

- **Ordinary clocks (OC)**: an OC is a device with a single network connection that can act as master of slave of a synchronization reference.
- **Boundary clocks (BC)**: a BC is a device that has multiple network connection (2..n), using one of them as the slave of a synchronization reference, while the rest acts as master.
- **Transparent clocks (TC)**: a TC is a device able to forward PTP messages from the master to the slave. It modifies these messages as they pass through the device to compensate the delay between them. TCs do not use the synchronization information to their own clocks.
- **Hybrid clocks (HY)**: a HY is a device able to forward PTP messages the same way a TC does. It also modifies these messages to compensate the delay between the master and the slave node. In contrast to TCs, HYs use the timing information from PTP to synchronize its clock.

PTP synchronization is achieved by the exchange of messages through the communication channel. PTP uses the following type of messages:

- *Announce* messages are used to built a clock hierarchy and select the best clock of the network (Best Master Clock Algorithm (BMCA)).
- *Sync*, *Follow_Up* and *Resp_Follow_Up* are used to compute the synchronization delay between the master and the slave clock.

- *Delay_Req* and *Delay_Resp* are used by OC and BC to compute the delay between two timing devices.
- *Pdelay_Req*, *Pdelay_Resp* and *Pdelay_Resp_Follow_Up* are used by TCs and HYs to measure the delay of the entire communication channel between a master and a slave device.
- *Signaling* messages are used for non-time-critical communications between clocks. These messages were included in IEEE 1588-2008.

PTP messages can be categorized as *event* and *general* messages. *Event* messages, such as *Sync*, *Delay_Req*, textitPdelay_Req and *Pdelay_Resp* are time-critical since the timestamp associated to the transmission affects the clock distribution accuracy. On the other hand, *Announce*, *Follow_Up*, *Delay_Resp*, *Pdelay_Resp_Follow-_Up* and *Signaling* are considered general messages. General messages are important for the PTP protocol but their transmission and receipt timestamps are not.

In order to perform the synchronization process, slave clocks determine the offset between themselves and their master reference. For a given slave device (s), the $\mathrm{offset}_{ms}(t)$ from the master (m) at a time *t* is defined by:

$$\mathrm{offset}_{ms} = s(t) - m(t) \tag{3.4}$$

where *s(t)* represents the time measured by the slave clock at *t*, and *m(t)* represents the same for the master clock.

The computation of the offset_{ms} is performed using four timestamps: t_1, t_2, t_3 and t_4. As Fig. 3.9 depicts, t_1 represents the moment *Sync* is sent from the master, t_2 to the moment *Sync* is receipt on the slave, t_3 represents the moment a *Delay_Req* is sent from the slave to the master, and t_4 is the moment the *Delay_Resp* arrives to the slave clock. Finally, the offset_{ms} is determined as:

$$\mathrm{offset}_{ms} = ((t_4 - t_1) - (t_3 - t_2))/2 \tag{3.5}$$

It is worth mentioning that PTP can work following two operation modes: one-step and two-step. Two-step includes the dissemination of a *Follow_up* message right after sending the *Sync* message, to transfer t_1 from the Master to the Slave. Besides that, one-step inserts t_1 within the *Sync* message, thus making unnecessary the use of *Follow_ups*. Configuring any of these modes depends on the PTP device hardware specs: one-step clocks require hardware capable of on-the-fly updates. Two-step clocks require that the software remember the dwell time of a *Sync* message and match it to the corresponding *Follow_up*.

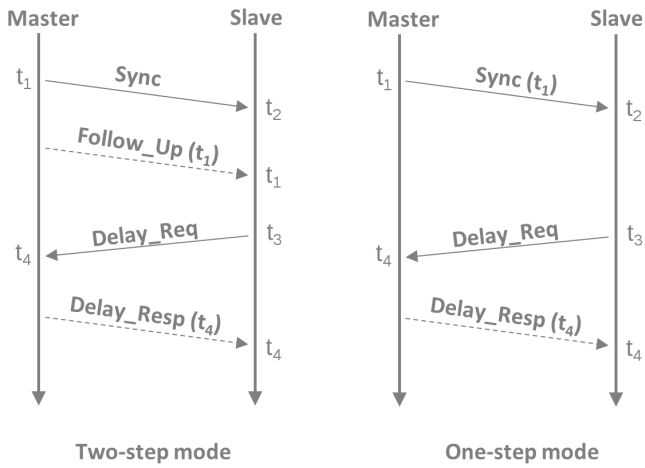

Figure 3.9
PTPv2 standard message exchange with four times-tamps. These time-stamps can be exchanged following two operation modes: one-step and two-step. Two-step introduces the *Follow_up* message that transfers t_1 from the Master to the Slave, whilst one-step includes t_1 within the *Sync* message.

3.4.5 *White Rabbit technology*

WR was born at the European Organization for Nuclear Research (CERN) as an Ethernet technology to synchronize devices with a accuracy of less than one nanosecond in scientific facilities such as accelerators and colliders. It is based on three elements: an extension of IEEE 1588 PTPv2, the distribution of frequency using a Layer 1 (L1) syntonization mechanism similar to Synchronous Ethernet (SyncE), and the recovery of the signal phase using Dual Digital Mixer Time Difference (DDMTD) components in order to improve the time-stamps accuracy.

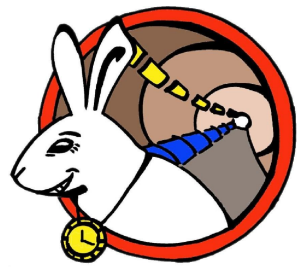

Figure 3.10
White Rabbit Technology Logo

The White Rabbit Precision Time Protocol (WR-PTP) implements a hierarchical master-slave architecture where the master uses a two-step synchronization mechanism to send its time to the slave.

WR is distributed under an open hardware license and it is available at the Open Hardware Repository and it has been proposed as the new high-accuracy profile for the IEEE-1588 standard [48]. WR main features are:

- Sub-nanosecond synchronization
- Connecting thousands of nodes
- Typical distances of 10 km between nodes but extensible beyond 100 km
- Ethernet-based Gigabit rate reliable data transfer
- Fully open hardware, firmware and software

WR devices are nowadays implemented as OC and BC, which performs the estimation of the link delay and the synchronization hop by hop using a two-step end-to-end (E2E) mechanism to propagate the clock, and *Delay-Request* messages to estimate the delay between them. Each device recovers the clock from its immediately before master frequency reference using a L1 frequency distribution approach, and after estimating the delay to the master, it computes the offset to the master using PTP frames. E2E is meant to be the best solution in scientific infrastructures, where there is not a complete knowledge of the network topology and PTP-like and non-PTP-like devices may share the network. Regarding scalability, E2E studies have stated that this mechanism increases both jitter and skew of Pulse per Second (PPS) signals as soon as the number of hops in the network increases. Alternatively, other implementations integrate a Peer-to-Peer (P2P) approach instead of the common-used E2E one. This approach has been developed for WR within this thesis. Chapter 5, Section 5.2.2 focuses on this development and the results obtained, being significantly better than the standard E2E WR implementation.

WR basis for syntonization, phase difference measurement and synchronization are presented below.

3.4.5.1 *WR layer 1 frequency distribution*

SyncE uses the physical layer to transmit timing the same way SONET/SDH. It provides a mechanism to transfer frequency over Ethernet networks.

SyncE provides a frequency transmission hierarchy formed on a link-by-link basis. The syntonization performance is evidently

immune to variations of the traffic load and packet delay, because clock recovery works on the physical layer independent of data transmission [49].

In contrast to standard SyncE, WR devices do not propagate the received clock immediately, they use their local oscillator to transmit the frequency reference. The clock of a sending node is only propagated to the directly attached opposite node of the link.

It is known that the distribution of a L1 frequency accumulates phase noise that degrades the performance of the synchronization between master and slave nodes. This is normally the main factor of synchronization lost on timing networks [49]. The WR approach is able to improve this issue by measuring periodically the phase difference between the two frequencies. For this reason, there is almost no phase noise accumulation, thus improving packet-switching synchronization protocols in terms of clock stability and reliability.

3.4.5.2 *WR phase recovery: improving hardware time-stamps*

In order to improve the accuracy of WR and avoid phase noise issues, a phase recovery mechanism is included within WR thanks to this L1 frequency distribution system similar to SyncE.

On the slave side, after recovering the clock from the master (L1 syntonization), the slave locks its PLL to the master reference. From this point, the slave is able to measure the phase difference between its local oscillator frequency and the one that is being received from the master using two DDMTD components. This phase difference is then included in WR PTP frames' correction field in order to include the phase difference in the final $\mathrm{offset}_{\mathrm{ms}}$ computation.

This solves the loss of performance caused by the phase noise that the L1 syntonization process introduces, thus improving the final WR accuracy.

3.4.5.3 *White Rabbit Precision Time Protocol (WR-PTP)*

WR-PTP is implemented as an Open Source PTP daemon called PPSi[15]. PPSi standard version develops 2-step end-to-end (E2E) clocks and measures the delay between two clocks using the *Delay-Request* mechanism. By default, WR clocks are OCs and BCs.

15 https://www.ohwr.org/projects/ppsi/wiki/wiki

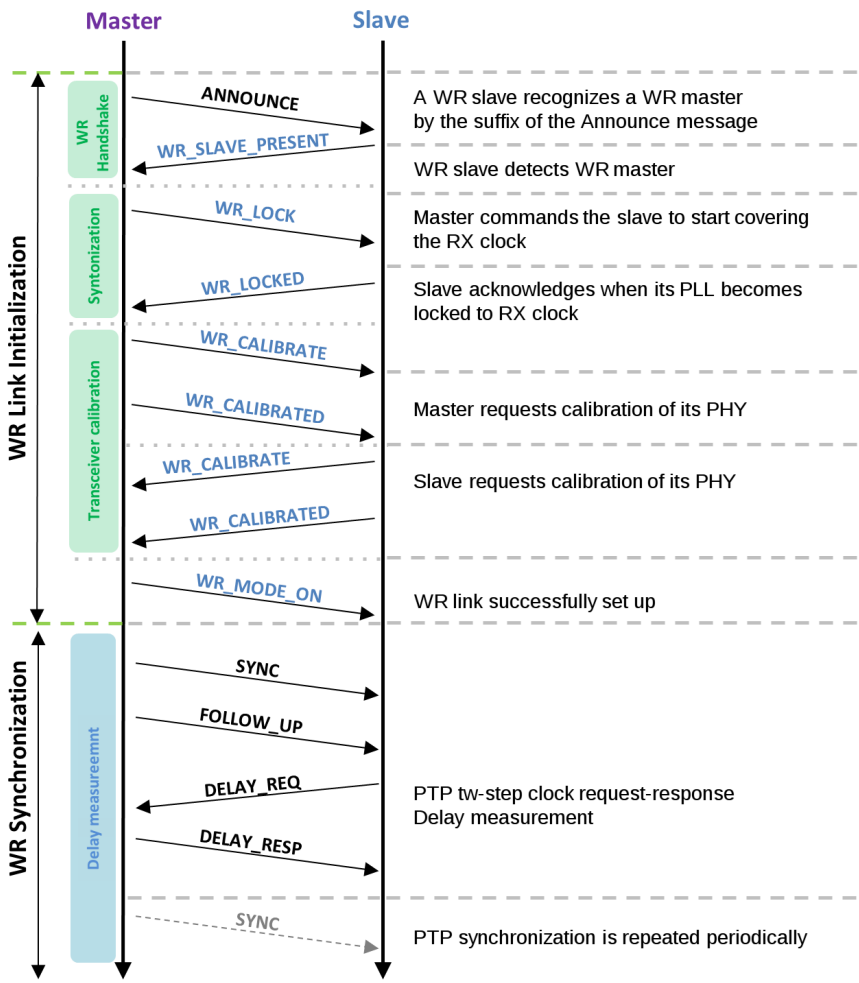

Figure 3.11
Complete PTP message flow during WR synchronization. Figure inspired in [50].

$$\begin{aligned} asymmetry &= \Delta_{tx_m} + \Delta_{rx_s} - \frac{\Delta - \alpha\mu + \alpha\Delta}{2+\alpha} && (3.6)\\ \mu &= ((t_4 - t_1) - (t_3 - t_2))/2 && (3.7)\\ delay_{ms} &= \mu + asymmetry && (3.8)\\ offset_{ms} &= t_2 - (t_1 + delay_{ms}) && (3.9) \end{aligned}$$

Fig. 3.11 depicts a complete WR-PTP message exchange between a master and a slave node. The first step is to perform the syntonization of the reference clock using the L1 WR link initialization. Due to the asymmetric transmission medium, this process includes the exchange of the different asymmetries on both sides, master and slave (3.6). Once the syntonization process is over, and the asymmetry of the link calculated, starts the

PTP synchronization process that includes the phase difference thanks to the utilization of the previously described DDMTDs.

PTP synchronization is carried out constantly, adjusting the slave oscillator by tracking the changes on the phase ($\mathtt{offset}_{\mathrm{ms}}$).

To summarize, Table 3.7 contains a compilation of the accuracies granted by each protocol described in this Section.

Table 3.7
Synchronization technologies features summary

Technology	Transmission Medium	Time Format	Accuracy
GPS	RF	GPS	~ 10 ns
IRIG-B	Serial Bus	IRIG-B	10 ms
NTP	Ethernet	UTC	< 10 ms
PTP	Ethernet	TAI	< 1 µs
White Rabbit	Ethernet	TAI	< 1 ns

3.4.6 *Dependable time transfer*

The development of reliable mechanisms for time transfer has become mandatory for critical time-sensitive applications. The loss of synchronization or accuracy during functioning may lead to an undesirable behavior that can affect to the entire system [51]. For this reason, HSR standard [7] Section A.5 describes how to map HSR and PRP protocols to PTP. Since this thesis focuses on HSR, only the HSR details are described below. For more details please refer to [7].

HSR devices send the same frame in both directions of the ring after inserting an HSR tag with a sequence number. Each device therefore receives two data frames, uses the first and discards the duplicate. This operating mode does not apply to PTP frames.

HSR PTP frames are sent duplicated through the ring, but each of them presents a different delay when they arrive at each node. This is due to the fact that they are modified on each hop of the ring. Fig. 3.12 represents the dissemination method for HSR PTP frames.

HSR PTP implements a 1-step or 2-steps *Sync* and P2P delay mechanisms as Fig. 3.13 depicts. Each node implements HY (TC + OC) and *peerDelay* to compute the delay between two adjacent nodes. The delay computed on each port of a node is used to estimate the *residence_time* of *Sync* messages on each hope. The sum of all *residence_times* are used to compute the final $\mathtt{offset}_{\mathrm{ms}}$ on each port. The BMCA of each HSR node is in charge of selecting which PTP data shall be used to synchronize to the best clock quality.

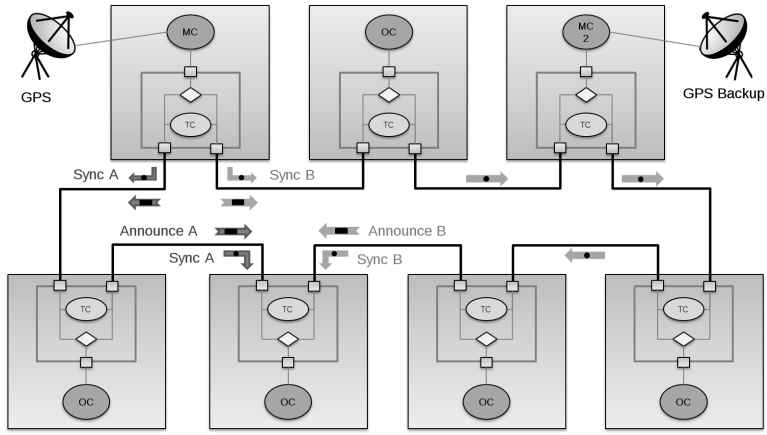

Figure 3.12
HSR with one GM. Inspired in IEC 62439-2 [7]

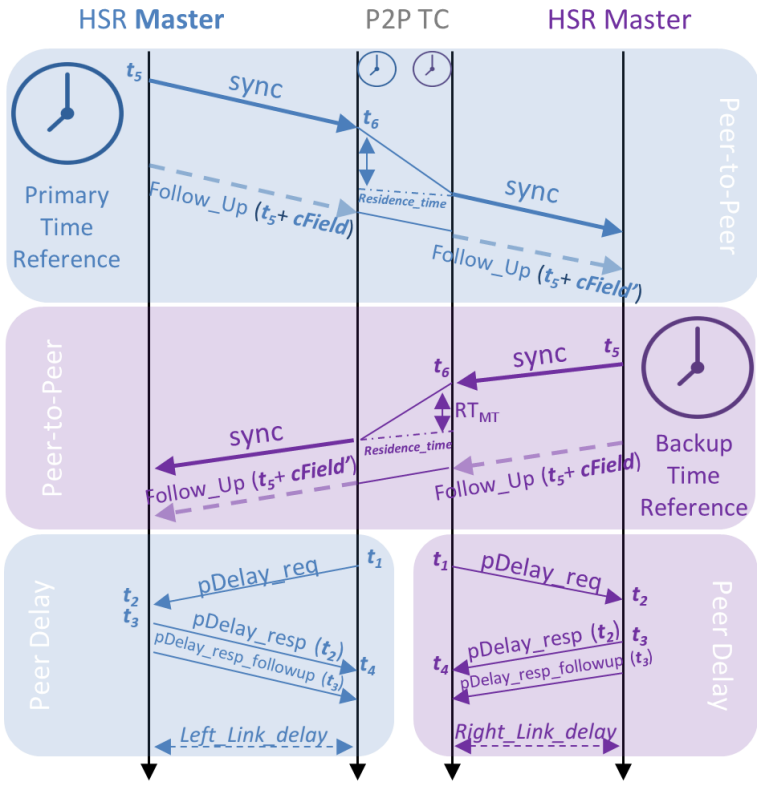

Figure 3.13
PTP messages sent and received by an HSR node (2-step). Inspired in IEC 62439-2 [7]

The reference that a node follows is considered the primary one. The other one is used as a backup reference, and, in case the primary is lost, each node shall start using the backup to synchro-

nize its local oscillator. This mechanism is known as *switchover* and it must be performed as soon as possible, so that the impact in the synchronization is seamless.

One of the contributions of this doctoral thesis resides on the implementation of a high-accurate version of HSR PTP in White Rabbit Switches (WRS) provided by Seven Solutions S.L.[16]. In addition, this development includes a special switchover mechanism that reduces considerably the impact on the WR synchronization accuracy after primary time reference failure. Chapter 5 describes this contribution in detail.

3.5 MARKET SURVEY

A market survey has been realized in the framework of reliable time and data transfer for industrial facilities. This survey focuses on features and prices of different devices that are capable of disseminating time over Ethernet networks (fiber and copper links) and their synchronization accuracy meeting the requirements from the industry standards like IEEE 1588 and IEC 61580. On the other hand, the redundancy capabilities detailed in IEC 62439 for both data and timing frames have been part of this study, too.

The study addresses different current products in the market from several companies (Fig. 3.14) such as ABB[17], Cisco[18], Flexibilis[19], Grid Solutions[20], MOXA[21], Siemens[22] and SoCe[23]. The characteristics analyzed are listed as follows:

- HSR and/or PRP compatibility
- PTPv2 compatibility
- Synchronization outputs
- Synchronization accuracy
- Price

The following Table 3.8 summarizes this study. Since there does not exist many devices with this features in the market, HDL IP Cores and small prototyping boards have also being considered.

16 http://sevensols.com
17 http://new.abb.com/
18 https://www.cisco.com/
19 http://www.flexibilis.com/
20 https://www.gegridsolutions.com/
21 https://www.moxa.com/
22 http://w3.siemens.com/
23 http://soc-e.com/

Table 3.8
HSR timing devices with HSR features characteristics and prices in the market

Device	Company	Timing Compatibility	Timing Output	No. Ports	Accuracy	Price
White Rabbit Switch	Seven Solutions S.L.	WR, PTPv2	PPS & 10 MHz	18	< 1 ns	3000 $
IE-4000-4GC4GP4G-E	Cisco	PTPv2	none	12	< 1 µs	6170 $
HPS - HSR-PRP Switch IP Core	SoCe	PTPv2	PPS	12	< 1 µs	20000-25000 $ (license)
ARWSCBRD-XRS7004E (Single chip)	Flexibilis	PTPv2	PPS	3	< 1 µs	825 $
PT-G503-PHR-PTP Series	MOXA	PTPv2	none	3	< 1 µs	2299 $
RUGGEDCOM RS950G	Siemens	PTPv2	none	3	< 1 µs	2077 $
ABB AFS660 Switch	ABB	PTPv2	none	3	< 1 µs	3000 $
Reason H49 PRP/HSR Redbox Switch	Grid Solutions	PTPv2	none	4	< 10 µs	3000 €

Figure 3.14
HSR devices/platforms with timing compatibility. From left to right: Flexibilis XRS7004E, Cisco IE-4000-4GC4GP4G-E, Moxa PT-G503-PHR-PTP, Siemens RUGGEDCOM RS950G, ABB AFS660 Switch and Grid Solutions Reason H49 PRP/HSR Redbox Switch.

From Table 3.8 it can be subtracted that the average price for Ethernet switches supporting HSR/PRP features together with timing compatibility seems to be around 3000 $. If we also consider single HSR/PRP timing nodes, the Flexibilies' XRS7004E presents the cheapest solution available (825 $). The synchronization accuracy of these devices is mainly provided by PTPv2, which are below 1 µs. In this context, the WRS is noteworthy for reaching an accuracy below 1 ns thanks to the WR technology.

Regarding timing outputs, in spite of being PTPv2 compliant devices, not many of them offer the possibility of outputting synchronized signals like PPS. In this respect, Flexibilies' XRS7004E, SoCe's HPS - HSR-PRP Switch IP Core and the WRS stand out against their competitors.

Finally, the WRS seems to be the best solution with the best timing performances. For this reason, the WRS has been used as the main development platform for the methods and implementations later described in this thesis. WRS hardware and firmware are briefly introduced in next Section 3.5.1.

3.5.1 *The White Rabbit Switch*

The WRS is the main element of a WR network, allowing multiplexing of high-precision timing and control data in single fiber connections. It is a full-duplex, non-blocking Gigabit Ethernet switch (IEEE 802.1D bridge) with 18 SFP ports, supporting fiber and copper connections.

Figure 3.15
White Rabbit Switch manufactured by Seven Solutions S.L.

The WRS provides the following features:

- Sub-nanosecond time accuracy
- 18 SFP Gigabit ports
- External inputs: 1-PPS, 10 MHz, allowing GM configuration.
- Outputs: 1-PPS, 10 MHz and 125 MHz
- Full routing latency determinism
- 1 uplink slave port
- 17 downlink master ports
- 1 copper management port running SNMP daemon
- Thousands of nodes
- 80 km fiber link distance
- PTPv2, SyncE supported
- Robustness and redundancy
- Dynamic calibration
- Open hardware

As it has been already stated, the WRS has been the main development platform for the work related to time synchronization and data redundancy features described in this thesis.

A summary of hardware, FPGA gateware and software details of a default WRS are described below. The complete description and schematics of the WRS can be found at CERN's Open Hardware Repository[24].

3.5.1.1 *Hardware*

The WRS is composed of two boards, the Switch Core Board (SCB) and the Mini Back-Plane (Mini-BP). The SCB is the main PCB and contains all the FPGA/CPU logic, together with the WR clocking system. Fig. 3.16 shows the board and also points out the different hardware elements of the platform as follows:

1. Virtex-6 FPGA LX240T
2. ARM processor (AT91SAM9G45)
3. DDR2 32M x 16
4. 256 MB NAND Flash
5. Temperature sensors
6. Power supply
7. SMC CLK/PPS in/out electronic buffers
8. Dataflash (bootloader memory)
9. WR clocking system (PLL AD9516, XO, DAC, etc)
10. DMTD clocking system

The mini-BP is considered an extension of the SCB board to add 18 SFP ports to the final device. Fig. 3.17 depicts the schema of the Mini-BP connected to the SCB.

24 https://www.ohwr.org/projects/white-rabbit/wiki/switch

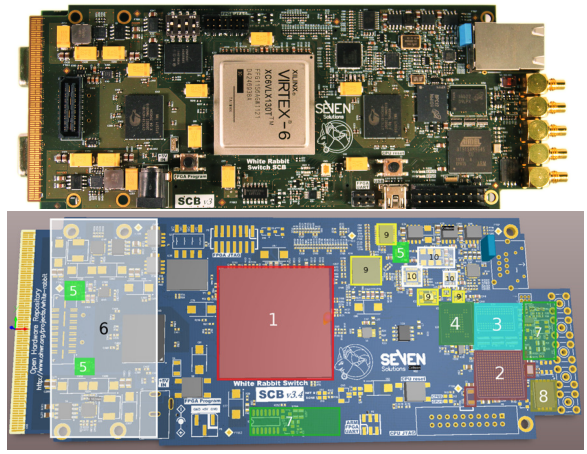

Figure 3.16
Switch Core Board picture (top). Switch Core Board schema with main components numbered (bottom)

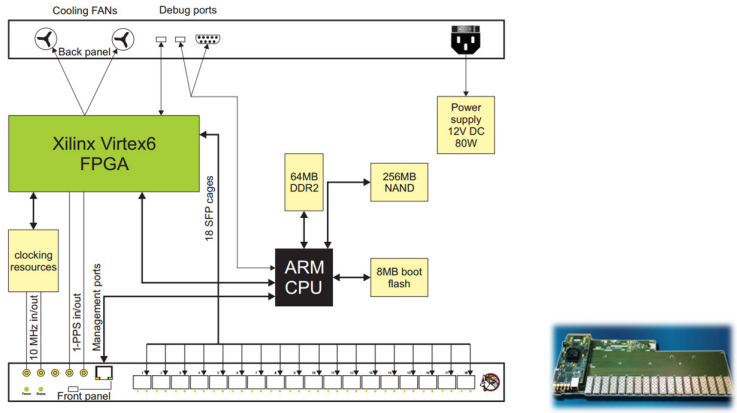

Figure 3.17
White Rabbit Switch Hardware Schema (left). SCB connected to Mini-BP (right)

3.5.1.2 *FPGA gateware architecture*

The FPGA of the WRS contains all the logic related to the WR technology regarding timing, and also the logic to commute the traffic of the network. Fig. 3.18 presents the main HDL IP cores that can be summarized as follows:

- **Endpoint**: It implements 1 Gigabit Ethernet MAC functionality. It receives and sends Ethernet frames with the capability to generate a timestamp for both Tx and Rx frames. Each endpoint is able to classify the incoming traffic in terms of type and priority (VLAN). The WRS contains 18 instances of this module.

- **Real-Time Subsystem**. It contains modules responsible for the time synchronization. It is composed of the Lattice Micro 32 (LM32), the softPLL, a DPRAM, a debug UART, a SPI and the 1-PPS generator.

  LM32: Soft-core processor that executes the software implementation of the PLLs. The instructions and data are stored in the DPRAM.

  SoftPLL: Hardware (HDL) part of the softPLL implementation. It is controlled over Wishbone from the LM32 software.

  1-PPS generator: Module responsible for generating the PPS signal output.

- **Tx Timestamp Unit**. This module collects all Tx timestamps for Ethernet frames sent through all 18 ports. Those timestamps are stored inside a FIFO queue and can be retrieved by the CPU.

- **Port Statistics**: It provides statistics for each port of the WRS.

- **Switching Core**: It is the switching logic for data in the WRS. It stores and forwards the traffic following the rules provided by the Routing Table Unit.

- **Routing Table Unit**: It is responsible for deciding where the frames must be forwarded after reaching the WRS.

- **Topology Resolution Unit**. This module provides hardware support for the topology resolution protocols implemented in software (e.g. RSTP and LACP).

- **Network Interface Controller (NIC)**: It forwards Ethernet frames between the Linux kernel running on the ARM processor to the 18 ports of the WRS.

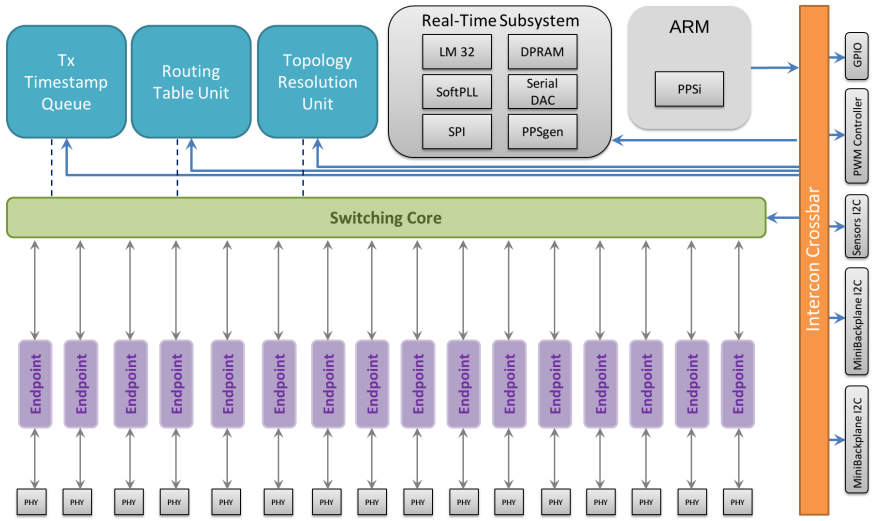

Figure 3.18
White Rabbit Switch FPGA hardware architecture.

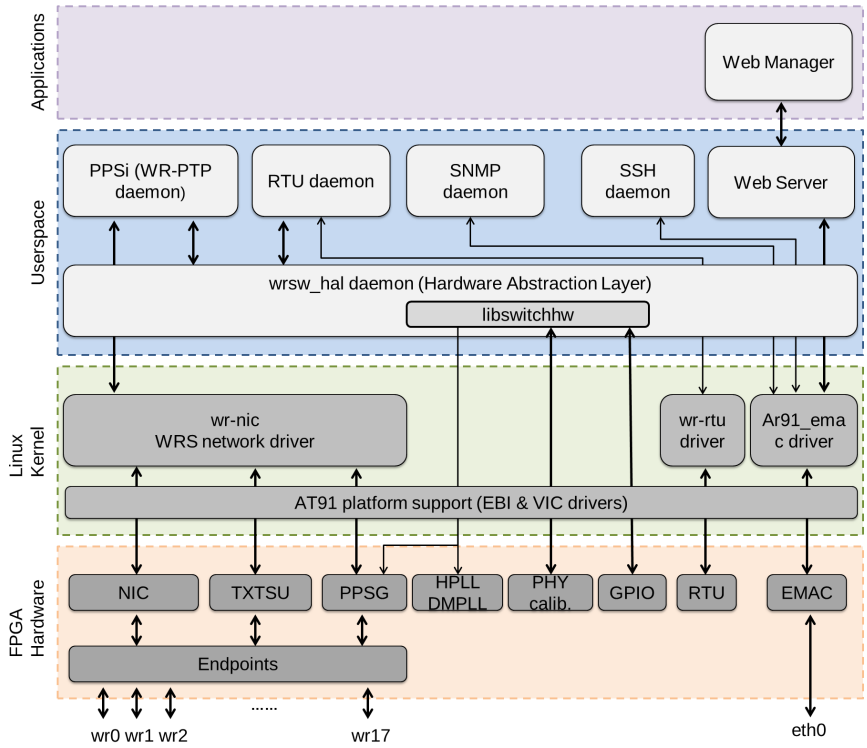

Figure 3.19
White Rabbit Switch software architecture.

3.5.1.3 *Software architecture*

The WRS software architecture is composed of several daemons that run on an ARM processor (AT91SAM9G45). These daemons interact with the FPGA components through an External Bus Interface (EBI) bus to perform the following tasks: time synchronization, data distribution and remote management. The communication between the ARM and the FPGA is carried out by the Hardware Abstraction Layer (HAL) daemon. Fig. 3.19 describes the interaction between the FPGA elements and the ARM daemons.

The main daemons and functionalities are described below:

- **PTP daemon**: it performs the WR synchronization. It retrieves Tx timestamps through the WRS NIC. Rx timestamps reach the ARM at the end the received PTP frames.
- **RTU daemon**: it creates a routing table to decide to which ports an incoming frame must be forwarded.
- **SNMP daemon**: it is used for management purposes. It replies SNMP queries using SNMPv2.

Part II

METHODS AND DEVELOPMENTS FOR DEPENDABLE SYSTEMS

4

METHODS AND DEVELOPMENTS TO INCREASE MIXED-CRITICALITY SYSTEMS RELIABILITY

Be wise. Be safe. Be aware.

— Dr. Wallace Breen, Half-Life 2

INDEX

The adaptation of the certification process from single to multi-core has been forced to develop new mechanisms to reduce the certification cost of multi-core systems. Furthermore, the utilization of multi-core platforms has made possible to implement SC and NSC critical applications in the same processors. For this purpose, this Chapter presents different mechanisms to increase reliability for mixed-criticality multi-core industrial systems without increasing the certification costs that, together with the reliability network mechanisms described in next Chapter 5, are capable of covering the dependability features required by any DCS. These multi-core dependable features are summarized as: providing isolation mechanisms between SC and NSC applications, enabling the re-certification process with no additional cost and integrating reliable communication channels to share data between different processors safely.

The work described within this Chapter was carried out in the framework of the RECOMP project as a result of the collaboration between University of Granada, Åbo Akademi and Seven Solutions. The main contribution of this thesis to this join effort resides in the development of the multi-core SC application.

This Chapter is structured as follows: Section 4.2 introduces the possible methods and methodologies used to increase reliability for mixed-critical systems. Section 4.3 describes the base of this work that involves the description of a case study: a safety emergency stop for industrial machinery. Section 4.4 presents the developments and architecture carried out to meet the requirements in terms of hardware and software of these applications. Section 4.5 integrates these methods and mechanisms to develop a new design with safety features for the WRS to increase its integrity level. Finally, results and conclusions of this work are presented in Sections 4.6 and 4.7, as well as the future work that derives from the presented implementation.

4.1 MOTIVATION

The main issue related to multi-core is that certification guides are clearly defined for single-core but not for multi-core, leading to new challenges and problems that must be addressed. In most of the cases, critical functions must run independently on a single-core with dedicated resources but in other cases, they are exposed to coexist together with non-critical functions, known as mixed-critical applications (Fig. 4.1). When this occurs, allocation technologies, partitioning and isolation methodologies must be extended from single-core standards to justify the spatial and temporal independence between each partition in multi-core systems [19].

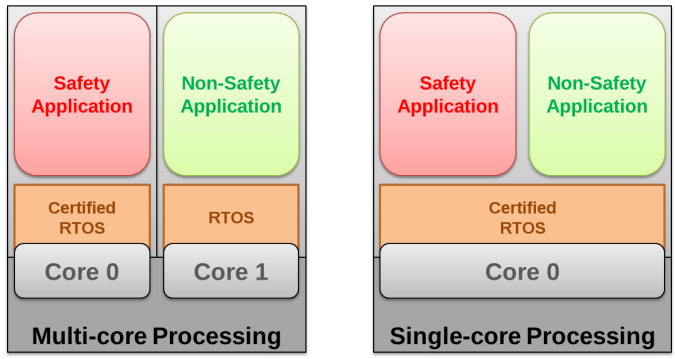

Figure 4.1
The left image represents a mixed-criticality application design for a multi-core architecture, running the SC and the NSC applications separately on two cores. The picture on the right shows a pure mixed-criticality design, both SC and NSC applications runs *simultaneously* on the same core.

The use of multi-core technologies is also a challenge for mixed-critical systems. The main issue is to make possible the coexistence of different criticality levels on the same computing platform. In these systems, low-critical and high-critical applications must share processing resources and time maintaining criticality properties. Unfortunately, this makes even more expensive and complicated the certification process, including the adaptation of certification standards, since it requires less criticality components to be certified at the highest criticality level. This problem can be extended from isolated multi-core processors to larger systems composed of several multi-core processors scattered over wide areas. In this case, new challenges such as network latency and processes synchronization play a key role in the certification process. These challenges are described in Chapter 5. Under this heading, only isolated multi-core processing and its certification process are described in this Chapter.

In terms of costs, the certification process of SC applications is an expensive and complicated issue. This process normally increases development costs anywhere from 25 percent to 100 percent [52] unless isolation between NSC and SC parts becomes proven. NSC parts can offer user experience, flexibility and dynamism in software (graphical user interfaces, Ethernet broadcast messages, etc), which are highly valued from a user's point of view. Such functionalities focused on system monitoring can be implemented as NSC software. As long as this NSC software is isolated from the SC part, there is no risk for injury to the users or environment [53], thus certification is not required for NSC. This leads to important cost savings in the certification process (0 percent for the isolated NSC part) of the whole system in case only the NSC application requires modification.

For this reason, the development of monitoring features as NSC applications is a low-cost solution compared to monitoring SC solutions. Moreover, developers may focus on the application features without taking care of safety limitations. The work described in this Chapter has been performed by following this alternative for the development of the NSC part of the mixed-criticality system using a multi-core architecture.

This Chapter presents a mixed-critical multi-core architecture in which different approaches have been developed to satisfy certification standard requirements by means of hardware, gateware and software. It is based on three elements:

- A multi-core open hardware platform capable of isolating fault propagation from the NSC part to the SC part while still providing communication [54]. It includes a specific gateware specially developed for the work described in this Chapter.
- A SC application with compliances according to SIL3 level in the IEC61508 standard.
- A NSC application allowing communication with the SC part and ability to update software during runtime.

A diverse architecture implementation for hardware and software has been designed following the recommendations from certification guides for the implementation of the SC part of the mixed-critical system. The NSC component is based on a sensing application with run-time capabilities that is not crucial for correct execution of the system. To further increase the flexibility of the NSC software, it implements a runtime updating mechanism. Using this mechanism, software developers are able to patch the running software without restarting the system or application. This is a desired feature, for example, in complex machinery, pulping plants, mills etc. [55] since a system reboot can be very time consuming.

For the development described in this Chapter, different requirements have been taken into account. These requirements were extracted from the work accomplished in an Artemis JU project, RECOMP. This implementation complies with the development recommendations of three certification standards: IEC 61508, DO-254 and DO-178C. These standards were previously described in Chapter 3.

4.2 METHODS TO INCREASE MULTI-CORE RELIABILITY

Mixed-criticality exists in several forms: certification wise (as in this contribution), application wise (mapping based priority levels), processor wise (mapping based on processor type), etc. [56]. The fundamental commonality is, however, to secure the execution for the *higher priority* part independent of the low priority behavior. Wasicek et al. [57] present a SoC platform for executing mixed-criticality applications in which a Trusted Computing Base (TCB) is used to isolate a critical part from misbehaving components outside the TCB. Complimentary to this work, the mixed-criticality architecture can be set-up to manage for example data isolation with trusted memory spaces [58]; this is done rather than having a dedicated software part to intercept faulty behavior due to increased speed and less resource use.

To schedule mixed-criticality applications Mollison et al. [59] suggested a multi-level scheduling mechanism for multi-core systems. The very scheduling technique is dedicated to a certain criticality level of the application used. High criticality applications are, for example, set to use only local static scheduling, while the lowest criticality levels use global best effort scheduling. A problem when using multi-core SMP scheduling in critical systems is to guarantee resources for critical applications in form of CPUs, OS resources, memory bandwidth when using inter-core locking.

One of the solutions proposed for the shared resources for multi-cores is virtualization. It consists in assigning access to shared resources by means of time or space. A shared source can be owned by a process for a slice of time or can be mapped only to a certain region of memory [53]. The most popular solution for memory management in RTOS is the utilization of hypervisors. Fig. 4.3 represents a hypervisor distributing resources in time slots and Fig. 4.2 a hypervisor allocating resources to a specific application.

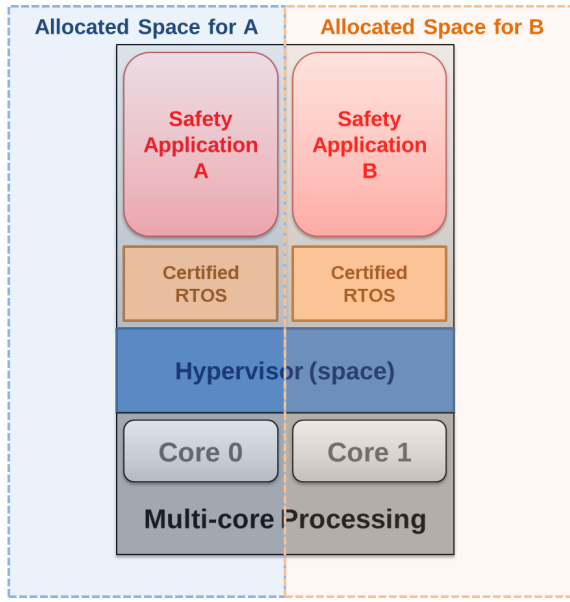

Figure 4.2
Space hypervisor. It allocates resources to application or threads during their entire execution. Other applications are not able to use the resources even if they are not being used.

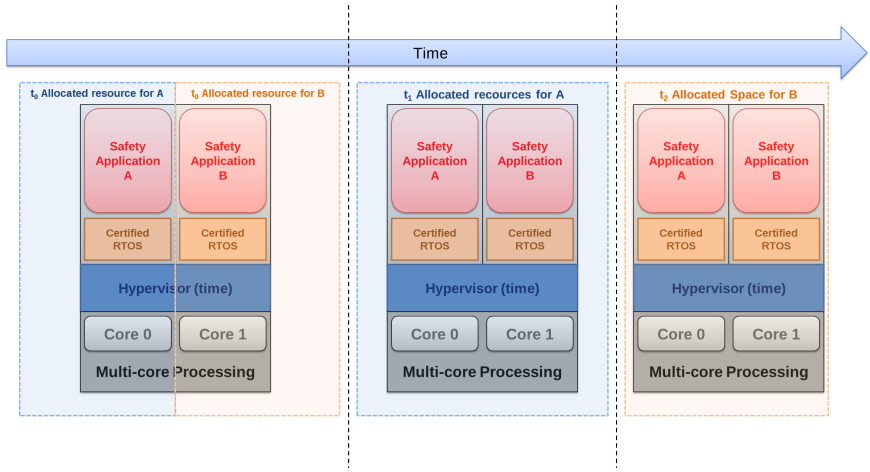

Figure 4.3
Time Hypervisor. It grant access to the resources for determined time slots, during this time, only one application (or thread) is capable of using the resource.

In this architecture, using a hypervisor was not necessary because of the utilization of an asymmetric multi-core OS (AMP) which maps one time sharing scheduler on each core in separate memory blocks (FPGA). No problems caused by inter-core locking is therefore present in the OS since all OS resources are requested from the local-core OS and inter-core communication is done explicitly via two mailboxes.

A SoC design for mixed-criticality applications, in which hardware and functional isolation mechanisms are used to guarantee correct execution for critical applications in a pacemaker, is presented in [20].

The CPU provides memory protection for shared scratchpad memories and functional isolation is provided by online monitoring. In contrast to this platform, the shared resources developed are directly controllable by the safe FPGA and all non-safe applications are running on the external ARM9. The utilization of this ARM9 processor together with the AMP FPGA architecture provides this platform with diversity, which is helpful to increase reliability in terms of fault-avoidance [60]. In addition to this, this design provides isolation by design and allows the safe FPGA to operate completely independent of the external ARM9.

Having introduced the motivation related to this contribution, the application scenario developed for the design of a mixed-criticality application is presented below.

4.3 DANFOSS CASE STUDY: A SAFETY-CRITICAL USE CASE APPLICATION

The Danfoss case study [61] was presented as a basic, complete and real (rich in safety-critical and certification concepts) example design in the framework of the RECOMP project. The SC application of the mixed-criticality system described in this Chapter follows the basis of this case study.

It consists in the removal of the torque from an industrial motor. A misbehavior of the machinery could put human lives at risk. To prevent any further damage, these systems are provided with an emergency button that generates an *Emergency Stop* (ES) signal, which must be monitored and implemented following the industrial standard IEC61508. It describes the necessity of using a redundant architecture to process the ES signal and control the status of the system. IEC61508 recommends using a dual channel *1 out of 2* (1002) as the safety control architecture for this type of machinery. It minimizes the effect of dangerous failures using two independent processors.

Its implementation develops a 1002 channel using two MicroBlaze soft-processors that perform a cross-comparison diagnose of the ES signal as IEC61508 states (Fig. 4.4) for SIL3 applications. This process results in the activation of the safety function that performs the removal of the torque, the Safe Torque Off (STO) function. After the activation of the STO from any of the pro-

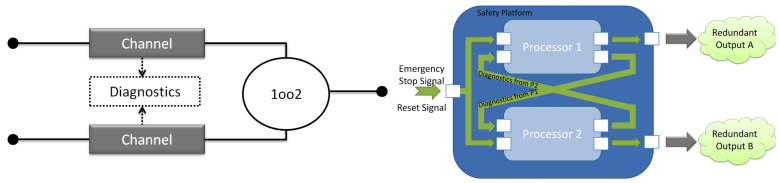

Figure 4.4
Redundant and cross-comparison methodology used in a safe channel architecture 1oo2. Left: Concept design of a safe channel architecture 1oo2 extracted from IEC 61508. Right: Design of a safe 1oo2 channel for this SC application using two processors.

cessors, the industrial motor removes the torque and the system halts.

The two processors of the 1oo2 architecture receive the same inputs: an ES signal and the data related to the STO function from the other processor. The data go through two diagnose processes, a cross-comparison function for local and external STOs, and another for the ES input and non-STO related variables as seen in Fig. 4.5. The diagnostics module includes also a liveness check routine implemented as a counter. The two cross-comparison modules together with the liveness check routine conform the failure detection system on both processors.

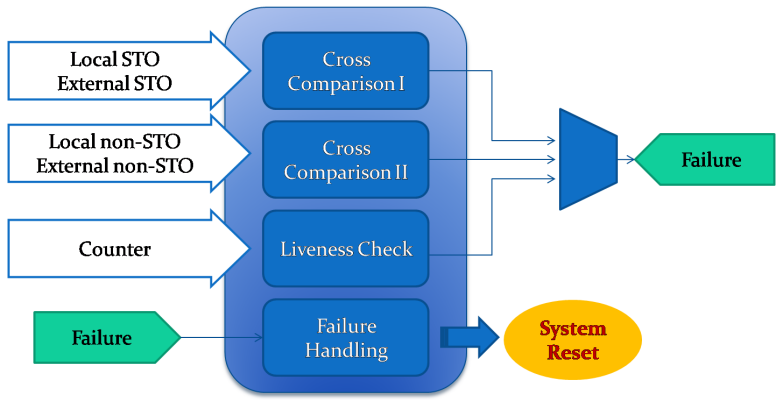

Figure 4.5
Processor's diagnostic module. Each processor implements a diagnostic module to detect system failure. It consits in four components: two cross-comparison functions, one for the local and external STOs and another for non-STO related signals, an external counter to check that the other processor is alive and a fourth function that detects when any of the other components rises an error signal.

The functioning of the SC application starts with a power-up stage that checks the availability and correctness of the peripherals. After a successful power-up phase, the system runs normally. It executes several diagnostic tasks used to ensure the correct operation of the system and then, the sensing application captures

and evaluates the data for monitoring. The behavior and measurement treatment of the NSC sensing application are described in Section 4.3.1.

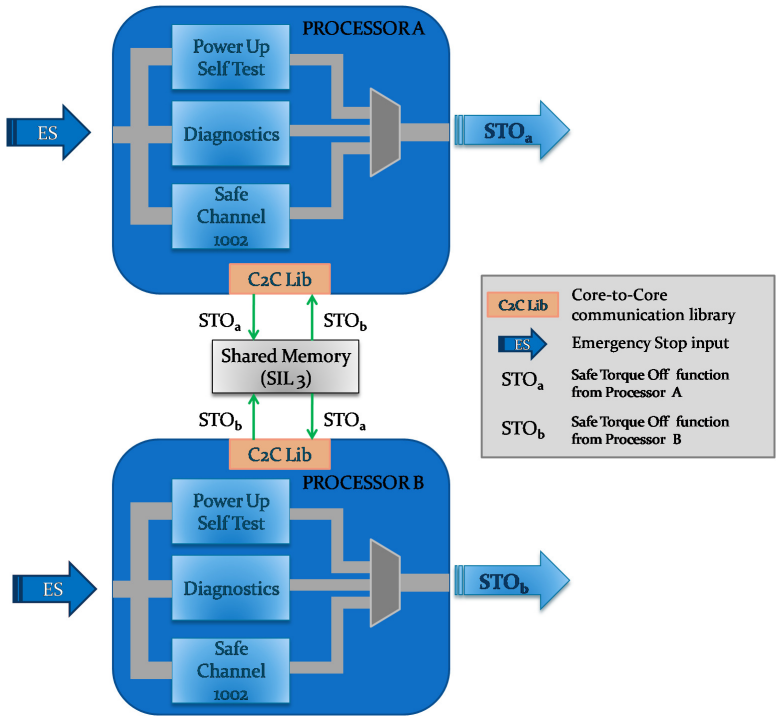

Figure 4.6
Core-to-Core communication architecture. The ES signals are connected to each processor in which the STO activation is evaluated. The STO related values are sent and receive from the other processor using a C2C library developed at SIL3 to perform the cross-comparison diagnose phase.

In case of emergency, the ES button is pressed by a machine operator. Immediately after, each processor activates the STO function so that the torque of the motor is removed. Each processor evaluates the ES input, its local STO, and the external STO from the other processor. As Fig. 4.6 shows, the exchange of data between both processors is accomplished using a Core-to-Core (C2C) library guaranteeing SIL3 for the communication [62]. After the cross-comparison diagnose processes the STO from both processors are activated.

Finally, both STO outputs are connected to a signal analyzer through a CAN bus system, which is in charge of removing the torque of the industrial motor when the STO is activated. The implementation details of the SC application are described in Section 4.4.4. It should be noted that, CAN bus and the signal analyzer deployments are not in the scope of this work.

4.3.1 *Integrating a non-safety-critical sensing application to the Danfoss case study*

NSC software is used to enhance the user experience and increase NSC functionality in a system. Since no certification is needed for this part due to the isolation methodology followed, its development and update do not increase certification costs.

This case study demonstrates a NSC sensing application for visualizing internal values of the SC FPGA platform. The implementation of the isolation mechanism is presented, which enables communication between the SC FPGA and the NSC ARM9 while guaranteeing isolation from fault propagation. The sensing application runs on the external ARM9 and contains a functionality for presenting various measurement values from the sensor devices on the FPGA to the user. Values such as *temperature*, *voltage*, *safety function signals* and *error values* can be read by the FPGA and sent to the ARM9.

To stress the meaning of maintenance costs, the sensing application includes an update mechanism which can modify software during run-time, which enables the system to change the program code of the executing tasks. Note that this updating mechanism enables the modification of the behavior of the sensing application avoiding the necessity of powering the system off nor restarting the application [63]. Similarly, the SC applications or the safe FPGA do not require a reboot when updating NSC software; which is beneficial for timing purposes. Moreover, no re-certification of the SC applications are needed when updating the NSC software since the platform guarantees correct behavior of the SC software regardless of the NSC software.

In summary, the case study represents a mixed-criticality system that is composed of two isolated applications: a SC part which evaluates the emergency stop of an industrial machine, and a NSC part that corresponds to the sensing application used for displaying measurement value and is able to insert new user necessities at run-time. Next Section presents the implementation that was necessary for the development of this system in terms of hardware, firmware, operating system and application.

4.4 IMPLEMENTATION OF A MIXED-CRITICALITY EMERGENCY STOP SYSTEM

This section describes the implementation and mechanisms developed to meet the requirements of the described case study in terms of hardware, FPGA gateware, and software.

4.4.1 *System architecture*

Certification standards stress that there are two procedures to include NSC elements in a SC application. One of them resides in certificating the NSC at highest priority level of the SC part, and the other is to isolate the NSC application from the SC part. The latter methodology reduces certification costs as no certification is required for the development of the NSC part. Updating only the NSC part would be exempt from certification too. For this reason, the design of this work focuses on isolating SC and NSC applications.

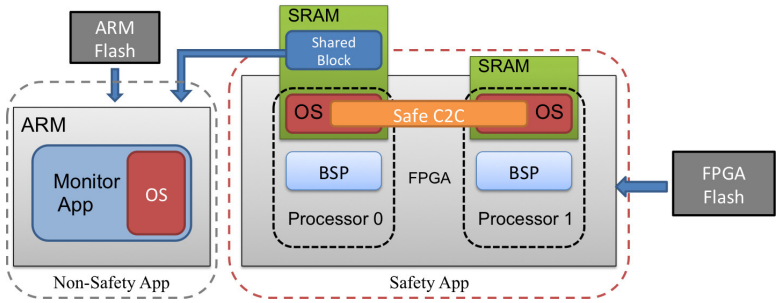

Figure 4.7
System architecture of the complete application. Each FPGA processor runs an instance of a RTOS (OS) with its correspondent Board Support Package (BSP) configuration. Moreover, both processors are connected via a safe core-to-core (C2C) communication channel. The monitor application (ARM) runs another independent RTOS (OS) and reads data from the FPGA through a SRAM shared block.

The isolation design starts with the separation of hardware elements and mapping shared devices independently to ensure a correct access behavior to them. Hardware isolation is performed by allocating the SC and the NSC parts in different components of the platform. The SC application is allocated on two FPGA soft-processors and the NSC part runs on an external ARM9 processor with its private memory and OS as Fig. 4.9 shows.

With this configuration, isolation is assured by means of hardware but this property needs to rise up to software [17]. This ARM-FPGA solution provides the system with heterogeneous and diverse elements, since it is running on two different isolated processors interconnected with a shared memory [60]. The FPGA hardware architecture (Fig. 4.9) has been developed as a prototype for the first development stage. Next step is the migration of this design to a new implementation with physical processors.

The OS used for this system is a Real-Time Operating System (RTOS) for embedded systems, called OpenRTOS [64]. OpenR-

TOS is a RTOS developed by Wittenstein High Integrity Systems [1]. The benefits of using this RTOS regards to the possibility of parallelizing functions/tasks in addition to a priority system that stands out for its simplicity and predictability. It is used for both SC and NSC parts and contains a simple scheduler which shares the execution time of tasks on the local CPU core.

Using a scheduler is necessary to parallelize access to shared memories and of course to make the tasks work "concurrently" for the emergency stop of the system, which involves up to 13 different OpenRTOS tasks at the same time. The utilization of a scheduler is also useful for parallelizing access to FPGA devices and the tasks that are involved in the core-to-core (C2C) communication between the two soft-processors that conform the SC part of the system and must be executed concurrently. The main objective of this application is the diagnostic of a safety function and the capture of system monitoring measurements. These values are later processed by the NSC part of the system, the sensing application.

Besides software, the hardware platform is also an important element in this system. The hardware platform on which it has been decided to implement the mixed-critical multi-processor architecture, is called Avionic Computing Platform (ACP), and has been developed by Seven Solutions[2] in the framework of RECOMP project. This development platform is described in detail in Section 4.4.2. Section 4.4.4 describes the SC and NSC parts of the system in terms of design and specification in more detail.

4.4.2 *Hardware platform*

The ACP platform (Fig. 4.9) is composed of two different boards connected through an external interface connector: the core board, in which processors and memories are included and the architecture is developed. This board is called AION. Below, it is connected to a peripheral board called RECOMP Sensor Board (RSB), which implements the required peripherals to fulfill safety-critical requirements and connections [54]. The AION board is a dual-processor that provides dual and diverse processor-devices: an ARM9 single-processor and a Virtex-6 FPGA. Along with these processors, independent memory chips (two QDRII chips for the FPGA device, DDR2 for the ARM9 and flash memory chips for each), safety peripherals like watchdogs and isolated

1 http://www.highintegritysystems.com/
2 http://sevensols.com

oscillators, are available for each processor (Fig. 4.9). More details about the platform development can be found in [65].

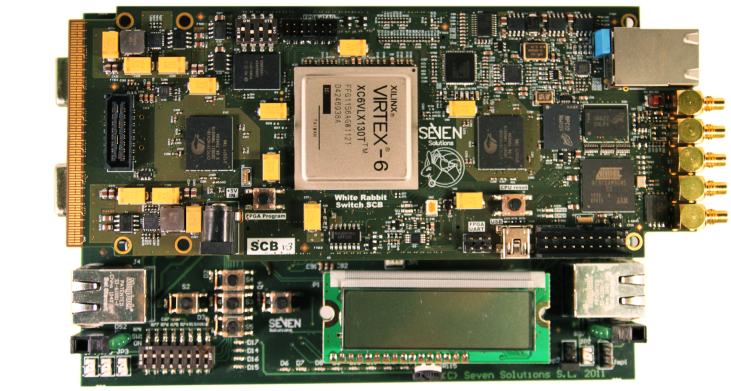

Figure 4.8
Avionics Computing Platform developed by Seven Solutions S.L. for RECOMP project.

This platform covers the hardware requirements of this mixed-criticality system since it requires independent and duplicate system peripherals in order to implement an AMP architecture inside the FPGA as the safety part, whereas the non-safety part is implemented inside the ARM9 processor (Fig. 4.9) [56]. Furthermore, according to the case of study, it was decided to implement an AMP dual processor architecture inside the FPGA provided within the ACP, whereas the ARM9 single-processor is used to run the NSC part of the system. Thanks to the utilization of the FPGA, the hardware architecture can be implemented (and modified if required) using soft processors in order to evaluate first the correct behavior and system functionality instead of being restricted to a single hardware architecture.

A heterogeneous AMP system is used, in which each software process is locked to a single core. This provides an execution environment similar to single-processor systems, allowing simple migration of legacy code. Moreover, it also allows developers to manage each core independently, implementing different OS and architecture in each one (memories, peripherals) if necessary. Nevertheless, the isolation of processors forces developers to implement a communication channel and its correspondent protocol for data transmission. This communication channel has also been implemented in the FPGA.

4.4.3 *FPGA gateware*

The FPGA implements two independent Microblaze processors. The Microblaze processors are connected to the board peripherals and external memories through a PLB bus which will be the main bus of the system. IP-cores (the commonly used) and the Microblaze processor are implemented in a safe and certifiable way by Xilinx in collaboration with third-party partners in the frame of critical projects. Therefore, although commercial versions of the IP cores are used, the migration of them to the qualified ones is very simple, which makes re-certification in future projects easier.

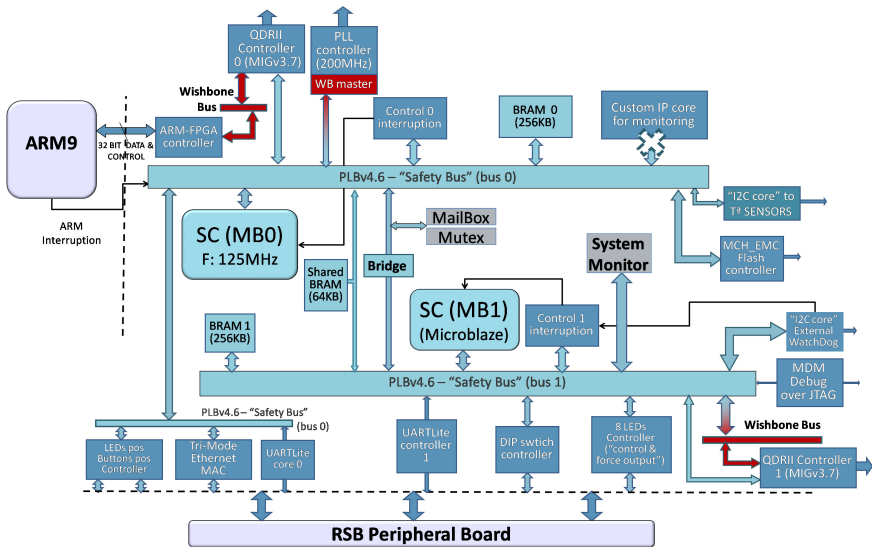

Figure 4.9
ACP On-chip FPGA architecture (adapted from [65])

Due to the utilization of an AMP architecture, each processor must run independently from each other and only some peripherals are shared. So the architecture can be seen as two isolated soft-processors with different accessible peripherals that are connected as Table 4.1 describes. The processors are connected to each other through a mailbox and a mutex IP-core (Fig. 4.9) to share packets and instructions and a 64KB shared memory to exchange largest amount of data and information. These are the components that are used to provide the system with communication features.

Due to the criticality nature of the SC part of the system, a controller for the shared QDRII memory has been developed between the ARM9 and the FPGA to ensure that the NSC application does not interfere with the SC one. This QDRII controller

AION Peripherals	MB0	MB1	Shared
QDRII External Memory	X	X	
Interruption Controller	X	X	
Internal Memory & Mailbox/Mutex Controller			X
External WatchDog Controller		X	
Flash Memory Controller	X		
SW Configuration Memory		X	
Temp. Sensors and FPGA Monitor	X		
LEDs and buttons	X	X	
JTAG Debug Controller			X
Serial Ports Controller	X	X	
LCD Controller	X		

Table 4.1
MicroBlaze processors connected peripherals.

has been modified in order to avoid shared memory problems that may cause a wrong behavior of the SC part of the system. Hence, these changes ensure that the ARM9 is just able to read the data that the FPGA writes in the QDRII from a specific region address. Any other operation from the ARM9 is denied by the QDRII controller.

Note that the industry certification standards states that the safe channel architecture 1002 needs to be implemented using two cores in two different platforms with different power supplies in order to maintain the required SIL3 level. The implementation here described uses two isolated processors in the same platform inside the FPGA, which means that SIL3 cannot be ensured as IEC61508 describes. Nevertheless, this FPGA platform is firstly used to develop and perform the isolation and the communication mechanism that the hardware architecture and the application need. Once this development is successfully accomplished and as future work, the design of both processors will be separated into two different platforms assuring the required SIL3 level.

The software implementations mapped to the FPGA and the ARM9 are described in next Sections *SC Software Implementation* and *NSC Software Implementation* respectively.

4.4.4 *Software implementation*

The developed software consists of to two different elements as Fig. 4.10 shows: the SC and the NSC applications. The SC software is mapped on the FPGA multi-core soft-processor. It con-

tains the logic required to satisfy the case study requirements previously introduced in Section 4.4.1. The application requires the exchange of information regarding each processors' diagnose results compared to the other processor. In addition to this, the utilization of an AMP architecture requires of a safe mechanism to send and receive the data. The reason for this is to avoid inconveniences of using shared memories and to guarantee the robustness of the SC system. For this purpose, a C2C communication library has been integrated to provide processor communications with SIL3 [62].

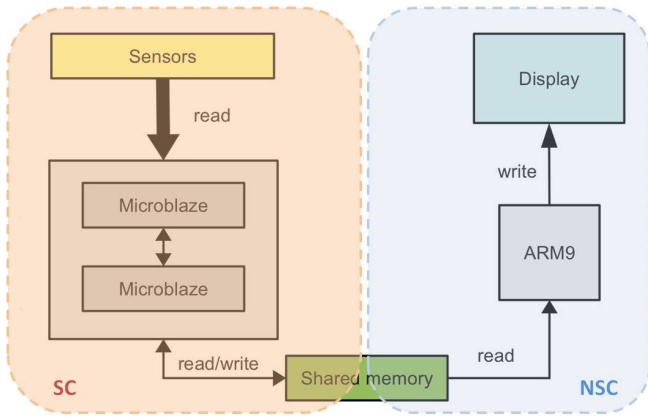

Figure 4.10
SC-NSC system data-flow. Represents the flow of data from temperature and system monitor sensors (SC) connected to the FPGA to the display connected to the ARM9 (NSC)

As previously mentioned, the chosen OS for the safety system is OpenRTOS and the programming language is C. Wittenstein provides a free license for FreeRTOS [66], which is a successfully small and efficient embedded kernel and compatible with OpenRTOS. FreeRTOS and OpenRTOS are not certifiable operating systems as such, but a certifiable version of these RTOSes is available for SC systems: SafeRTOS [67]. SafeRTOS is a certified pre-emptive RTOS that maintains the same features as the mentioned RTOSes and, in addition, contains additional features required for certification, such as a complete isolation system for SC tasks by the definition of Memory Protection Unit (MPU) regions per task.

By using a OpenRTOS, the execution order of tasks can be priorized depending on the relevance of their work. In addition, this RTOS includes an specific C2C communication library called WC2C library (WC2C) [62], which has been also developed by Wittenstein in the context of RECOMP. This library offers the possibility to exchange data between several processing elements

connected through a shared memory and a mailbox. In addition to this, the WC2C library ensures safety features for the critical data exchanged between processors without putting at risk the reliability of the entire SC application as direct access to shared memories may cause. Fig. 4.11 displays the final implementation design of the mixed-criticality emergency stop system presented within this Chapter.

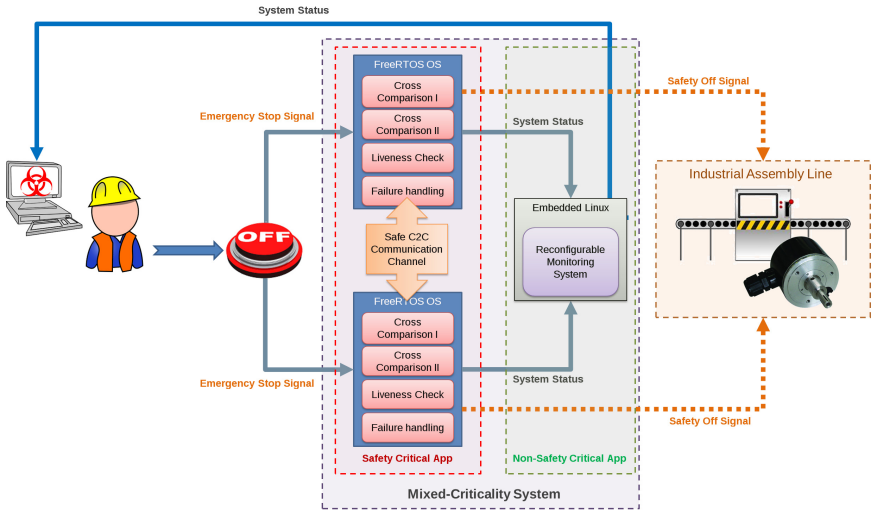

Figure 4.11
Final design of the mixed-criticality emergency stop system including the Danfoss case study concepts together with a non-safety-critical monitoring system.

4.4.4.1 *SC software implementation*

The most relevant part of the SC application is performed by each processor due to the tasks that it executes. Each processor runs different tasks in parallel to guarantee the correctness of the system. Three different tasks can be defined: STO diagnostic, system monitoring and communication. The STO diagnostic functions handle all tasks that evaluate and diagnose the system status. The SC application decides whether the STO must be activated or not. Moreover, these tasks must determine whenever the system reaches an undesirable state caused by an anomalous behavior, causing the system to a power cycle reset in order to regain the safe state.

The STO diagnostic tasks can be grouped into the following categories:

- Power up tasks: This task starts with a complete checking procedure during the power-on of the system. Once the initial test has been successfully accomplished, it remains

activated waiting for the moment the safety function is activated, in order to contribute to the maintenance of the system.

- Diagnostics tasks: This is the main element among the diagnostic process. It processes both local and external STO function status after the power-on self test stage during the whole execution of the platform. The diagnostic of the state of the platform is transmitted to the other core through the C2C channel.
- Failure detection tasks: this task updates consecutively the system watchdog. When a failure occurs, it stops updating the watchdog causing the FPGA to reboot. The FPGA is then re-programmed with the initial content of the flash memory. The flash contains the FPGA firmware as well as the two OpenRTOS applications.
- Communication tasks: each processor runs also two tasks that are used to send and receive data from the safe C2C channel. One task is in charge of sending the local STO status values to the other processor, whereas the other one is reading.

These categories are composed of a certain number of tasks for each processor (Figs. 4.5 and 4.6). Table 4.2 summarizes the software implementation of these tasks.

Task Name	Type	Description
vPowerUpSelfTests()	Power Up	Performs a peripheral check at start-up
vCrossComparisonIAnalysis()	Diagnostic	Verifies the both local and external STOs
vCrossComparisonIIAnalysis()	Diagnostic	Verifies internal and external data different from STOs
vLivenessCheck()	Diagnostic	Checks whether the other process is alive or not
vFailureHandling()	Failure Handling	Diagnostics task. Launches a reboot command in case of failure
vSTO()	Diagnostic	Handles the local STO function for each processor
SafeChannel1oo2Write()	Communication	Uses the WC2C library to send data to the other processor
SafeChannel1oo2Read()	Communication	Uses the WC2C to read data from the other processor

Table 4.2
Main function tasks implemented in each processor. Tasks are divided into three type of functions: power-up, diagnostics and communication.

In order to guarantee the correct behavior of the AMP platform and the schedulability of these tasks executing on each processor, FreeRTOS and OpenRTOS offer a routine that ensures the atomic execution of critical sections.

These routines are `portENTER_CRITICAL` and `portEXIT_CRITICAL` and are described in [64, 66]. Inside a critical region the scheduler will never extract the task from the processor during the execution of these lines by disabling hardware interrupt signals, avoiding undesirable and unpredictable reads/writes on peripherals and shared devices.

The monitoring tasks are used to measure values from internal sensors and registers for runtime monitoring. The SC application periodically reads all the sensor measurements, such as temperature sensors and system monitor values, as well as both local and external STO states, and writes values to a pre-defined region in the shared memory as unsigned 32-bit integer values. These values can be used to detect over heating of components or under voltage brown/blackouts. Furthermore, the SC part writes the STO signal for each processor, and error values for the STO function inconsistency in the shared memory. The STO signal is inconsistent if, for example, one processor signals STO:high while the other signals STO:low.

4.4.4.2 *NSC software implementation*

All NSC software containing the sensing application and the runtime updating mechanism is mapped on the external ARM9 processor. The goal of the sensing applications is to read measurement values from sensors connected to the SC FPGA and present the values on a display. The obtained values are partly derived from hardware sensors connected to the FPGA and partly from the STO applications described in previous sections. Since the sensing application is completely without certification, no actions apart from displaying values are taken from this part of the system.

The ACP platform allows communication between the FPGA and the ARM9 via a shared QDRII memory. Moreover, the NSC application has no guarantees for correct execution, and must be assumed to unexpectedly generate faults. These faults cannot propagate back to the shared memory and interfere with the SC part. The challenge is therefore to successfully isolate the SC part from the NSC part and to prohibit fault propagation to the SC part while allowing communication.

The sensing application is mapped to the ARM9 CPU described in Section 4.4.2. It is running as a task on top of a FreeRTOS [66] port created for the ARM26EJ-S. Fig. 4.10 illustrates the data-flow from sensors to display via the shared QDRII memory. Initially, the sensor values are read by one of the MicroBlaze cores and written into shared memory. The ARM9 then polls the shared memory periodically to read the stored values. A read from the shared memory is performed simply by reading a 32-bit pointer value from the memory address associated with the shared memory. When a read is issued by the ARM9, a memory controller managed by the FPGA is called. It fetches the data from the memory block and sends it back on the bus connecting the ARM9 and

the FPGA. Currently, the available data from the SC written in the shared memory block of the QDRII are:

- Three external temperature sensors
- On-chip temperature sensor for the FPGA
- Internal and auxiliary voltage sensor for the FPGA
- STO function status from processor 0.
- STO function status from processor 1.
- Failure status errors generated by processor 0 undesirable behaviors.
- Failure status errors generated by processor 1 undesirable behaviors.

The ARM9 reads the shared memory completely autonomously and, in real-time, translates the values to relevant unit such as Celsius degrees (°C) and millivolt (mV) depending on which value is read. These values are then displayed through the serial port to a terminal for the machinery operator.

– *Isolation mechanisms:*
One of the key features in mixed-criticality systems is the guaranteed isolation [57] between the safe and the non-safe part. As the software on the ARM9 is not certified and thus assumed unsafe, a miss behavior in the ARM9 cannot propagate to the safe FPGA. This means that the only communications channel – the shared memory – must be protected against misuse. Misuse can, in form of unintended faults, be originated at the non-safe ARM9 in form of:

1. Data overwrite: The ARM9 overwrites critical data in the shared memory.
2. Resource locking when writing: The ARM9 locks a memory space for an unpredicted time when writing.
3. Read flooding: The ARM9 floods the bus with reads and blocks the FPGA from writing.

Isolation can be guaranteed with the respective solutions:

1. ARM9 has read only access: A write will not change the value of the content.
2. ARM9 has read only access: A write will not lock the shared memory since the resources will never be accessed.

3. The QDR memory ensures the scheduling between read and write. This is provided at hardware level by the memory IP core controller developed in the FPGA.

The mentioned solutions for preventing fault propagation is possible since the FPGA is able to set read/write permissions for the shared QDRII memory. The ARM9 is not able to interfere with the FPGA in any other way, since the bus connecting the shared memory is the only physical connection between the two processors.

– *Run-time updating of NSC software:* To further improve the dynamism and flexibility of software, a runtime updating mechanism for NSC software has been implemented originating from the thesis [68]. Run-time update of software is a process of replacing an existing part of software on a running platform with another part without shutting down the system or restarting the application. Reasons for updating software are usually due to version updates, bug fixes, algorithm optimization, and to keep the software more up-to-date with the users.

This procedure is fairly trivial as long as no SC software is running on the same platform and as long as the update is not executed at runtime. The shut down process and start up of a SC system can be very time consuming in complex machinery, mills, etc. This leads to NSC software updating possibilities only during complete system maintenance in which the whole system is brought to a stop. Since the uptime of large machinery is a crucial part of its efficiency, complete system shut downs should occur as seldom as possible. Updating software online – on the other hand – does not require a reboot of the running system, nor does it require a restart of the application the update was performed on. A runtime updating mechanism will also enable remote updating of systems via the Internet, which reduces personnel expenses significantly. Bug fixes and other patches to NSC software could easily be distributed to systems in remote locations from one single location without the restart of the SC nor NSC part.

Run-time updating of NSC software on a mixed-criticality platform is therefore an interesting use-case since a) both the updated software and the updating mechanism itself are allowed to interfere with the SC part b) the updating of NSC software on a mixed-criticality platform has the potential to dramatically reduce the development costs for NSC software and its integration.

The runtime updating mechanism is created for lightweight embedded systems running on FreeRTOS. It is capable of transferring the task state of any FreeRTOS task into the updated task version without system reset. As previously said, this NSC part including the sensing application and the updating mechanism, has been implemented and mapped onto the ARM, which is by hardware isolated from the safe FPGA. Hence, users can add new features to the system without interfering the correct behavior of the SC application. To this end, the modification performed on the shared QDRII block memory controller (described in the previous hardware Section) consolidates the avoidance of undesirable access from the NSC to the SC. This ensures the complete isolation between SC and NSC applications, thus demonstrating that the developed mixed-criticality system complies with the industrial certification standards that industrial machinery requires.

4.5 EXTENDING SAFETY-CRITICAL CONCEPTS TO THE WRS ARCHITECTURE

As previously discussed in Chapter 3, one of the boards that compose the WRS is the ACP, which has already been described in Section 4.4.2. Due to the fact that the WRS does not implement any SC feature itself, this Section introduces a concept design that enables increasing the integrity of this device. This enhancement is based on the the duplication of the PPSi daemon that runs on the ARM and the softPLL process that is executed on the LM32 inside the FPGA. This duplication allows the implementation of a cross-comparison stage for the execution of the PID controller similar to the one described in Section 4.4.1, in order to ensure a correct system functioning. The cross-comparison phase monitors the adjustment results of both PI controllers and their applicability to the PLL, considering the latter as a shared resource.

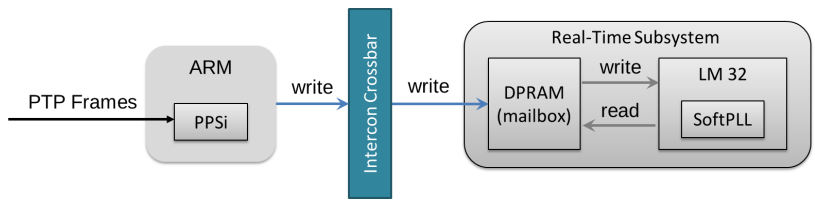

Figure 4.12
WRS PPSi/LM32 non-reliable communication architecture. PPSi runs on the ARM processor, which computes the $offset_{ms}$ and sends it to the softPLL that is running on the LM32 through a non-reliable single mailbox. Then, the softPLL adjust the oscillator using this offset.

Fig. 4.12 represents the standard non-reliable PPSi-softPLL architecture. The PPSi daemon receives PTP frames and computes the $\mathtt{offset}_{\mathtt{ms}}$, which is sent through a mailbox to the softPLL that is running on the LM32. The PID controller converts and applies this $\mathtt{offset}_{\mathtt{ms}}$ to adjust the local oscillator to maintain the WR synchronization. This architecture does not develop any redundancy feature, therefore, wrong offsets computed by PPSi or softPLL will be likewise applied to the PLL, causing synchronization performance loss.

In order to solve this issue, we have designed a new PPSi-softPLL architecture. In this schema, we have duplicated all resources related the PPSi-softPLL workflow. We have defined two instances of PPSi and the softPLL, a cross-comparison algorithm to evaluate the measurements of the two processors that must be applied to the shared PLL, and a safe communication channel between them as Fig. 4.13 depicts.

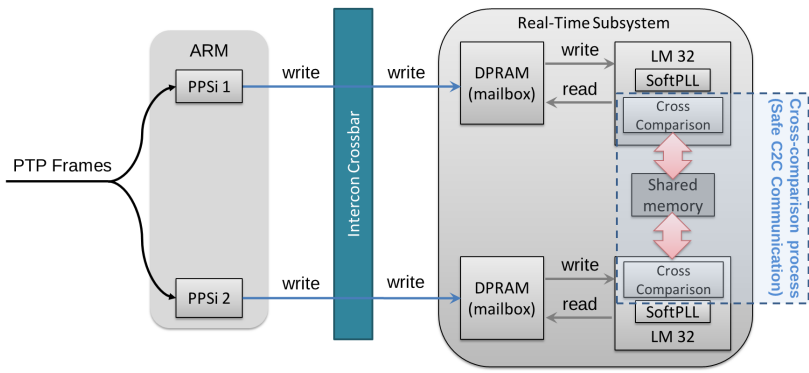

Figure 4.13
WRS PPSi/LM32 reliable communication architecture. Two PPSi instances run on the ARM processor, which computes two $\mathtt{offset}_{\mathtt{ms}}$ with the same PTP information received (duplicated). These offsets are sent to two softPLL instances, which adjust the local oscillator after executing a cross-comparison algorithm. In case a wrong value is computed by PPSi or the softPLL, it will be discovered at the cross-comparison stage.

PTP frames are received duplicated on each PPSi instance. Both of them compute and send the $\mathtt{offset}_{\mathtt{ms}}$ to the softPLLs running on each LM32. Then, the cross-comparison algorithms exchange through a safe communication channel the values measured by each softPLL. The difference between these two values determines whether the received PTP data and computed offsets are valid for the PLL or not. Three possible scenarios are defined:

- Both processors have measured the same value for the PLL. In this case, the value is used to adjust the PLL.

- Both processors have measured different values, but the difference is negligible (below a pre-defined threshold). Any of them might be applied to the PLL.
- Both processors have measured different values, but the difference is considerable (above a threshold). None of the measured values are applied to the PLL and an alert is sent to the WRS management daemon. In this scenario, it is better to wait for the next valid measured value[3].

This design improves the resilience of the standard synchronization process of the WRS. The duplication of resources such as PPSi and the LM32 covers the redundancy suggestions from IEC 61508. In addition, the utilization of a safe communication channel between the two LM32 enables the possibility to perform the cross-comparison phase, adding an additional reliability layer. Unfortunately, due to time constraints, this design has not been implemented within the course of this thesis. Due to its technical possibilities, it has been included in the Future Work Section 7.3.

4.6 RESULTS

This Section describes the results of the implemented system, which demonstrates the correct behavior of the heterogeneous multi-core mixed-critical system in which the SC application performs critical tasks correctly under all possible circumstances, and the NSC part does not interfere with it due to the isolation mechanisms here developed. Moreover, NSC-SC (ARM9-FPGA) isolation reduces significantly the certification cost. It is also included in this case study an example of on-the-fly update software in the sensing application.

4.6.1 *Safety-critical results*

In order to verify the correct behavior of the SC system application, several tests have been performed. These tests evaluate the multi-core issues related to the certification process of this application: hardware and software redundancy, diversity, C2C communication channel architectures and isolation mechanisms. The testing methodology used is the following:

- Fault injection in the ES signal activation of the redundant STO activation process.

3 Under normal temperature conditions, a WR slave node is able to follow the changes on the retrieved clock even without computing the PTP $\texttt{offset}_{\texttt{ms}}$.

- Fault injection in the exchange of the STO signal through the C2C channel architecture.
- Fault injection in the cross-comparison functionality.
- Isolation controller to ensure the NSC part is not interfering the SC one.

Fault injection has been the evaluation technique to check redundancy correctness, detect errors and measure the response time of the case study. This fault injection method consists in three different experiments. The fist one is the simulation of loss of information between hardware and software components. This loss should not hamper the system because of the redundant hardware and software architecture which guarantees the reception of the signal from two different paths. To simulate the loss of information, a subroutine disables the ES read function in one of the processors as i.e., a physical wire cut. When the ES button is activated, one of the processors' ES signal remains always inactive while the other one will read the correct ES activation and thus, generating the STO signal which will remove the torque. The reception of at least one of the ES signals is guaranteed by the redundant hardware architecture.

The second one injects wrong STO values to the C2C communication system to simulate that one of the processors is not processing the ES signal properly. This leads to a fake STO announcement from one of the processors to the other or the lack of it. In both cases this situation must be detected by the cross-comparison diagnostic functions (see Figure 4.5 and Table 4.2) in each OpenRTOS instance and start with the safety system halt. This halt has to ensure the removal of the torque from the motor although only one processor activates the STO signal. The activation of the STO signal from any of the processors is guaranteed thanks to the utilization of the diverse software architecture and the 1002 communication channel architecture.

Fig. 4.14 presents the complete state diagram of the fault injection error scenario previously described and the response of the system to injected faults. When the ES button is pressed, each processor receives the order to start the diagnostic process for the STO execution. After the activation of STO in one processor, this signal is sent to the other processor to perform the cross-comparison diagnostic. In case both STO signals are activated with no failure, the system removes the torque normally and it is halted (green lines in Figure 4.14). In case the two STO signals differ from each other, the system will remove the torque, but it automatically starts with the reboot sequence to guarantee a

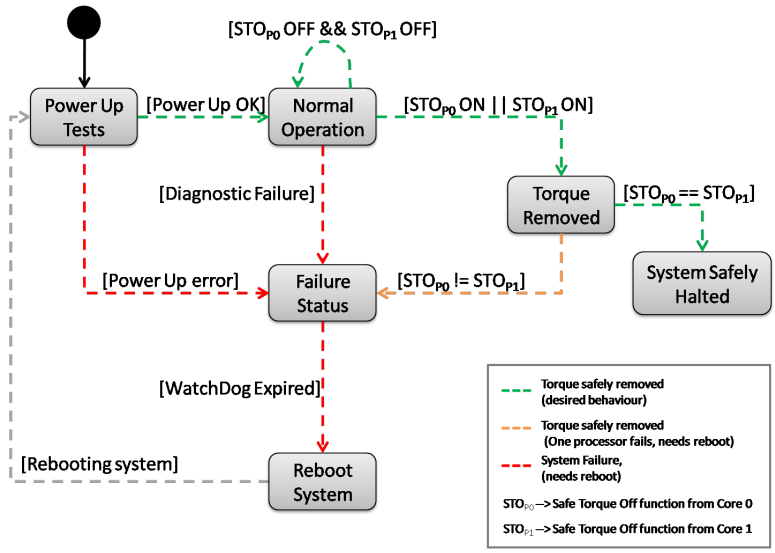

Figure 4.14
State diagram of the SC system behavior. The green lines represent the correct status of the application. It ends with the expected removal of the torque preceded by the activation of both STO functions from processor 0 and 1 (STO_{P0} and STO_{P1} are ON). The yellow line represents the correct removal of the torque but the system needs to reboot since STO_{P0} and STO_{P1} are not equal (safe state but undesirable). Red ones represent the expected actions that the system performs when errors occur (failure)

normal system behavior again (green lines). This information is stored in the shared memory for the NSC part.

The reboot sequence consists in the expiration of a watchdog. When the processors detect a system failure, they stop updating the watchdog and, two seconds after the watchdog expires, the system reboots, it resets the FPGA and loads both firmware and RTOS from the flash memory. This response time is completely customizable and depends on the system requirements. By this, the correct behavior of the SC part is guaranteed in case of hardware connection faults.

The third fault injection simulates a software error that affects internal variables directly related to the local and external STO diagnostic phase. This method has been implemented as a function that modifies the local STO-dependent values randomly inside each of the OpenRTOS applications. The cross-comparison functions for both local and external STO data discover an incongruence data exchange (red lines). In this case, both processors request a hardware reset. Once again, the tasks updating the watchdog stop and the watchdog expires prompting a system reboot.

In terms of isolation, the variables that depend on the functionality of the mixed-criticality system are written in the shared

memory in order to be read from the sensing system. The controller (Fig. 4.15) that was developed for the shared memory restricts the NSC part allocated on the ARM9 any possibility of writing. Moreover, read functions are performed by using a different HW bus (Wishbone) that the one used by the FPGA (PLB) to write in the memory. By this it is ensured that the SC part is completed isolated from the NSC application preventing from any undesired behavior and thus, reducing the necessity of developing the NSC part to SIL3 (SC part) that leads to cost saving in the certification process.

The three fault injection tests together with the isolation controller show how the error prevention mechanisms work and, at the same time, how the system response flow is restricted as described in Fig. 4.14. The SC implementation satisfies the following certification challenges for multi-core mixed-criticality architectures:

- Isolation of SC and NSC parts.
- Certification/re-certification of NSC updates with no additional cost.
- Individual memory mapping for both processors.
- Safety communication channel developed at SIL3.
- Scheduling capabilities using a RTOS.

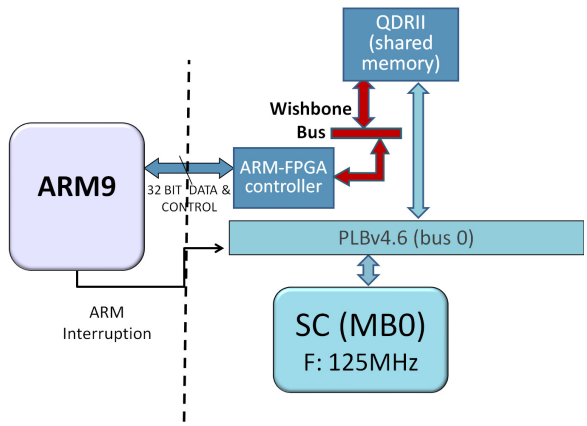

Figure 4.15
HW isolation of the memory shared by the NSC and SC parts. The ARM-FPGA controller isolates and avoids any interference between the shared data from the SC part to the NSC application. The QDRII shared memory is accessible from the SC part (MB0) through the PLB bus and is able to write and read. The NSC part (ARM9) is only allowed to read the QDRII memory and it accesses the memory through the Wishbone bus.

4.6.2 *Non-safety-critical results*

As presented in Section 4.4.4.2, the NSC sensing application periodically reads sensor values from the shared memory provided by the safe FPGA. This Section presents the results of the implemented system by showing the right behavior of the SC application and no interference of the NSC part when performing runtime updating. For this case study, it has been chosen to update the software capable of displaying the measurement values on a display. This means that a failure in the display software is not critical for human safety, but degrades the user experience significantly. The reason for using runtime updatable software in display software is the ability to restructure the display layout, add more features, increase/decrease accuracy of measurement values, modify the update period, etc.

The first version of the software included two measurement values: On-chip temperature and internal chip voltage. The software also included two derived values: Max chip temperature and average chip temperature. These values are based on previous sensor values and is therefore part of the state context. This is shown by updating the software that transfers the state of the application to the updated software.

A timer was used as decision maker for the runtime update. At a pre-defined time the timer called a callback function in the display task to signal its update request. The display task then entered the safe state and its context (max and average temperature) was saved. The newer version (Version 2.0) of the display software restored the context, and was able to continue registering maximum and average temperature based on the previously collected data. Fig. 4.16 shows both versions of the display software.

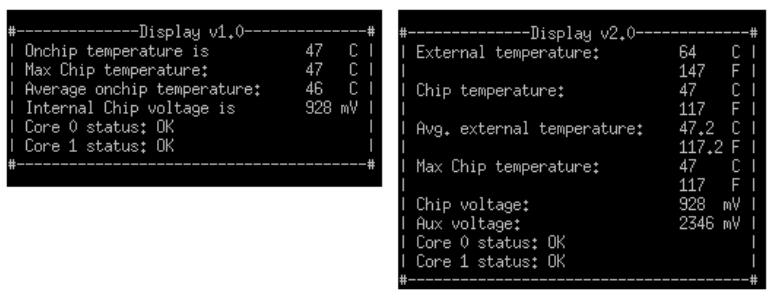

Figure 4.16
Runtime update of display software. Version 1 to the left and Version 2 to the right

The updated version of the display software allows more sensor values such as: external temperature, and auxiliary voltage levels. It also displays all temperatures in both Centigrade and Fahrenheit degrees – a feature added by the new software. It is worth mentioning that both display and run-time update mechanisms do not interfere in the SC system behavior because of the previously stated isolation between the ARM9 and the FPGA. In addition, any update/upgrade of the NSC application would not derive in any additional certification costs.

4.7 CONCLUSIONS

A complete and reliable mixed-criticality system has been presented in this Chapter which involves safe execution of an emergency stop button which removes the torque from a motor of an industrial machine. Furthermore, a NSC sensing application with run-time upgrade capabilities has been included in the final implementation. The utilization of multi-core has made possible to develop specific isolation mechanisms for NSC and SC and also to implement a diverse and redundant solution for this industrial problem.

The utilization of a reliable hardware platform have been implemented by following certification standards. This provides a system with isolated and redundant peripherals. A dual-core architecture has been used for an AMP application with two MicroBlaze processors on a Virtex-6 FPGA to achieve the duplication of hardware required by the certification standard. These processors run two isolated instances of OpenRTOS which communicate via a safe C2C communication channel that has been implemented based on requirements from IEC61508 SIL3. The utilization of the WC2C communication library, together with the isolated FPGA, has enabled the fulfillment of one of the main requirements for this kind of SIL3 systems: the redundant channel architecture (1002), which is necessary for the cross-comparison diagnose stage of the system. This diagnose stage represents the redundant component which ensures the correct behavior of the system in response to the activation of the safety function in which human lives rely on.

In addition to this, it has been included a complete sensing system application and a runtime updating mechanism which provides the possibility of upgrading the NSC part with new features depending on the user experience and needs at runtime. Due to the isolation between the FPGA and the ARM9 created in hardware, the NSC part can be upgraded without interfering

with the correct behavior of the SC part, and thus, reducing certification costs.

As already stated, several parts of the system have been implemented following certification standards. The complete isolation between both SC and NSC parts of the system at hardware and software levels even when the updating mechanism is running has been demonstrated. For the intercommunication between the processors the hardware itself provides the necessary isolation while still allowing inter-processor communication. Nevertheless, this inter-processor communication is supervised at software level and performs a safe communication system over shared resource inconveniences. Hence, hardware, firmware and the communications channel are close to a certifiable system.

It is worth mentioning that the flexibility of the platform in terms of reconfigurable hardware and software, leads to an important point of our research, the introduction of Open Source approaches regarding the certification process as open-boxes or systems. This helps to save time-to-market and development costs but the main feature obtained is that open platforms help to improve the reliability of the overall system since reviewers, source code and safety evidences correctly documented, can be completely examined and verified by a wide engineering community [54].

To fully achieve a certified product, the next required steps to cover all the safety standards requirements for every hardware and software components are listed below. In further versions of the system, the feasibility of migrating current IP cores will be evaluated to qualified ones suitable for FPGA design due to their high cost, only affordable for the industrial domain and business cases. Additionally, other proper methods would be used in order to achieve a final DO-254 certified gateware. In the same manner, the safety in the execution of the SC application must be improved. Due to the fact that OpenRTOS is not certified, it is needed to integrate the safety version of this RTOS, SafeRTOS. By this, the required safety properties for hardware, firmware, OS and communication between local processors that are necessary for certifying a complete mixed-criticality system are completely covered.

From this point forward, guaranteeing reliability of distributed inter-process communications in DCS will be the main topic of next chapters.

5

METHODS AND DEVELOPMENTS FOR HIGHLY DEPENDABLE TIME AND DATA DISTRIBUTION IN INDUSTRIAL NETWORKS

Alice: But I don't want to go among mad people
The Cat: Oh, you can't help that,
we're all mad here. I'm mad. You're mad
Alice: How do you know I'm mad?
The Cat: You must be, or you wouldn't have come here.

— Lewis Carroll, Alice in Wonderland

INDEX

This Chapter presents the methods and mechanisms implemented to increase time and data reliability in industrial networks as it has been covered in the state-of-the-art Chapter 3, Section 3.3. These mechanisms complement the ones described in previous Chapter 4 but, in this case, the focus is on the distributed nature of the system. The combination of these approaches

makes possible the development of a robust and dependable distributed control system. In addition, industrial compatibility and scalability have been also studied within this Chapter.

The rest of this Chapter is structured as follows: Section 5.1 presents the motivation of this work, Section 5.2 the mechanisms implemented to improve the scalability of the timing solution, Section 5.3 describes the developments carried out to increase timing fault tolerance for the WR protocol and Section 5.4 presents the FPGA implementation that guarantees reception of critical data in case of network failure. Section 5.5 shows the results for these three features, and Section 5.6 states the conclusions related to the scalability and dependability of these implementations.

5.1 MOTIVATION

Industrial facilities and more particularly Smart Grid, combine the integration of different electrical engineering, energy storage and advances in new information and communication technologies within the electrical energy domain, from the generation process to its commercialization. This allows the interconnection between the control, instrumentation, measurement and energy administration in a global management system in order to promote rational and efficient use of energy [1].

This evolution on the global management system imposes the utilization of reliable and widespread communication network infrastructures like Ethernet and a desirable improvement in network services availability, security and safety due to their critical nature [2, 3]. For this reason, these systems need to incorporate different mechanisms to increase fault tolerance and the avoidance of single point of failure so that control decisions and actions can be disseminated as soon as possible and with the minimum latency possible [4].

The information and control data that are transmitted through Smart Grid networks are considered critical since a non-received control event message could lead to personal injuries and also the loss of enormous sums of money. In addition, the correct synchronization of the devices that form the grid is crucial [8] to maintain a common notion of time of when and what occurred on the network as already discussed in Chapter 3 Section 3.3.1.

In terms of time synchronization, the highest accuracy requirement (tens of nanoseconds) comes from PMUs [9] in Smart Grid. GNSS, in spite of being globally used, are subject to lose synchronization performance. Furthermore, the use of GNSS as the unique time source is considered vulnerable in terms of security

(due to spoofing, jamming, etc). IEEE recommendations focus on providing an alternative method to GNSS by using terrestrial systems [10]. The utilization of a solution combining GNSS and wired technologies such as IRIG-B and recently, PTP based on Ethernet networks [12], is becoming widely popular in this field [11]. In this regard, the integration of the WR technology in Smart Grid communication networks resolves the security problems GNSS solutions present (wired technology), and covers the forthcoming strict synchronization needs introduced by the utilization of PMUs (sub-nanosecond accuracy) [39, 41, 42].

Regarding availability and fault tolerance, IEC 61850 [6] suggests the implementation of redundancy protocols to meet these Smart Grid requirements. Several protocols can be developed for these fields but, PRP and HSR protocols are the most commonly used [7].

PRP and HSR include the development of different mechanisms to provide zero-time recovery in case of a network component failure. This is due to the fact that this recovery time or network reconstruction requirements are even tougher for Smart Grid. For example, the performance of the commonly used Rapid Spanning Tree Protocol (RSTP) is not acceptable anymore (30-60ms recovery time) [69].

Following these recommendations, the aim of this work resides on reaching the best possible synchronization accuracy for industrial applications together with the possibility of recovering from a system failure to increase system fault tolerance. Moreover, duplication of timing references and also data becomes mandatory for Control Systems and Smart Grid as H. Kirrmann et al. present in published contributions [70–72]. This leads to the implementation of redundancy protocols as [6] suggests. Currently, only few products that combine all these features together based on standard PTP have been identified, the maximum accuracy reached lies between 30 ns and 100 ns [73–76].

The rest of the Section focuses on providing methods to increase scalability and industrial compatibility, timing and data reliability in network infrastructures.

5.2 EXTENDING WR TOWARDS SMART GRID INTEROPERABILITY

WR default implementation lies on BCs, which performs the estimation of the link delay and the synchronization hop by hop using a two-step E2E mechanism to propagate the clock, and *Delay-Request* messages to estimate the delay between them. Each

device recovers the clock from its immediately before master frequency reference using a L1 signal recovery approach similar to SyncE, and after estimating the delay to the master, it computes the offset to the master using PTP frames. E2E is meant to be the best solution in scientific infrastructures, where there is not a complete knowledge of the network topology and PTP-like and non-PTP-like devices may share the network. Regarding scalability, E2E studies have stated that this mechanism presents worse results for jitter and offset skew measurements of PPS signals as the number of hops in the network increase [77, 78]. In addition, although E2E can be used in redundant protocol implementations like PRP and HSR, [7] suggests the utilization of P2P instead of E2E for engineering networks. Fig. 5.1 depicts both E2E and P2P delay models.

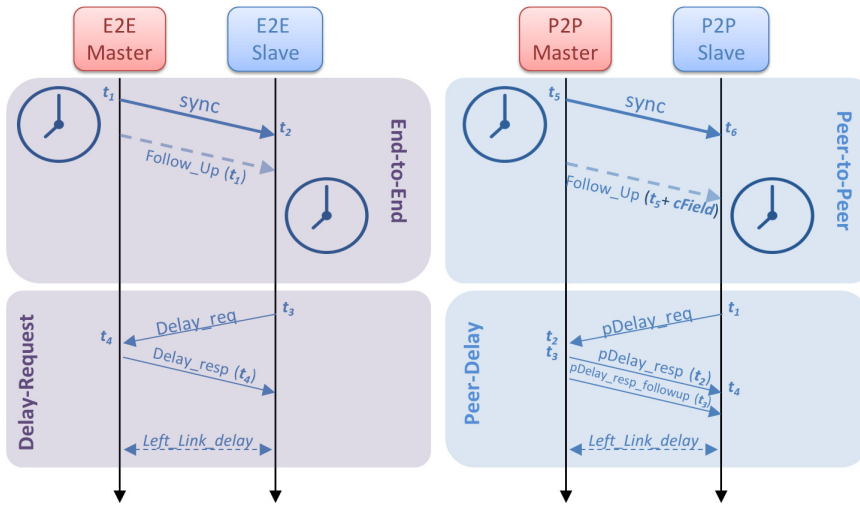

Figure 5.1
The E2E delay model uses four times-stamps and the *Delay-Request* mechanism to compute the offset between the master and the slave node (left image). The P2P delay model computes this delay using six time-stamps and the *peerDelay* mechanism (right image).

For this reason, the development of P2P clocks becomes almost mandatory to use WR in SmartGrid networks as the main time provider, leading to the utilization of TCs and HYs instead of BCs for the middle nodes that compound the network. P2P is used in engineering networks where all nodes are known to be IEC 1588 compatible and able to distribute time frames. In this type of networks PTP frames are sent from the master to the slave node and forwarded by intermediate nodes like switches and routers, considering the entire network as a simple fiber link.

The development of WR TC/HY involves the implementation of mainly two mechanisms: a P2P mechanism to send *Announce*,

Sync and *Follow_Up* frames from the master to the slave, and a *peerDelay* mechanism to estimate the delay between neighbor devices. Furthermore, WR requires the dissemination of the frequency over L1, too. Next Sections 5.2.1 and 5.2.2 present the differences and similitudes compared to the default WR implementation.

5.2.1 *Syntonization on WR P2P clocks*

In spite of the utilization of TC or HY in P2P networks, the distribution of the frequency from the master to the slave must also be forwarded through intermediate nodes. This is due to the fact that final P2P nodes (HYs) need master's frequency to reach the WR sub-nanosecond synchronization. For this reason, the WR syntonization for TC/HY has been realized in the same way it is carried out for BCs. WR BC link syntonization model has been already described in Chapter 3, Section 3.4.5.1.

5.2.2 *WR Synchronization on WR P2P clocks*

In contrast to E2E WR clocks, P2P devices send PTP frames from the master to the final slave through TC or HY middle nodes. These nodes forward *Announce*, *Sync* and *Follow_Up* frames instead of generating them locally. In addition, they compute the link delay using the *peerDelay* mechanism instead of the one used by BCs (*Delay-Request*). These concepts are detailed below.

PeerDelay measures the delay of the link between two adjacent nodes using four timestamps (t_1, t_2, t_3 and t_4) as Fig. 5.2 shows. It uses three type of messages: *PDelay_Request*, *PDelay_Response* and *PDelay_resp_followup*. First, t_1 corresponds to the moment a node sends a *PDelay_Request* to its adjacent node, t_2 is generated in the other node as soon as *PDelay_Request* is received. This second node responds with a *PDelay_Response* message and generates t_3, which is received in the requester at t_4. Previous t_3 is received immediately after in a *PDelay_Resp_Followup* message. The receiver uses the timestamps to calculate the delay as it follows (5.1):

$$delay_{adj} = (t_2 - t_1 + t_4 - t_3)/2 \tag{5.1}$$

In P2P, in order to measure the clock offset, *Announce*, *Sync* and *Follow_up* frames are sent from the master to the slave, considering all intermediate nodes of the network as TCs or HYs, where these frames are forwarded as Fig. 5.2 shows. A Slave node is not

aware of the delay from the master since its delay is estimated with the current neighbor nodes using *peerDelay*, for this reason it is necessary to keep track of the delay accumulated by all the links/nodes of the network path. This is performed using the *Correction_Field* (cField) of *Follow_Up* frames.

Right after the master sends a *Sync* message, t_5 is generated and sent in the next *Follow_Up*. When a *Sync* message is received in a TC, a time-stamp $t_{sync\_ingress}$ is generated and the message is immediately forwarded to the other active ports generating a t_{sync_egress} time-stamp. These two timestamps are used to calculate the *residence_time* (5.2) of each *Sync* message on each of the outgoing ports. In addition, *cField* might also add the link delay computed in the incoming port.

$$\text{residence\_time} = \text{sync\_egress} - \text{sync\_ingress} \tag{5.2}$$

$$\text{cField} = \text{residence\_time} + \text{delay}_{adj} \tag{5.3}$$

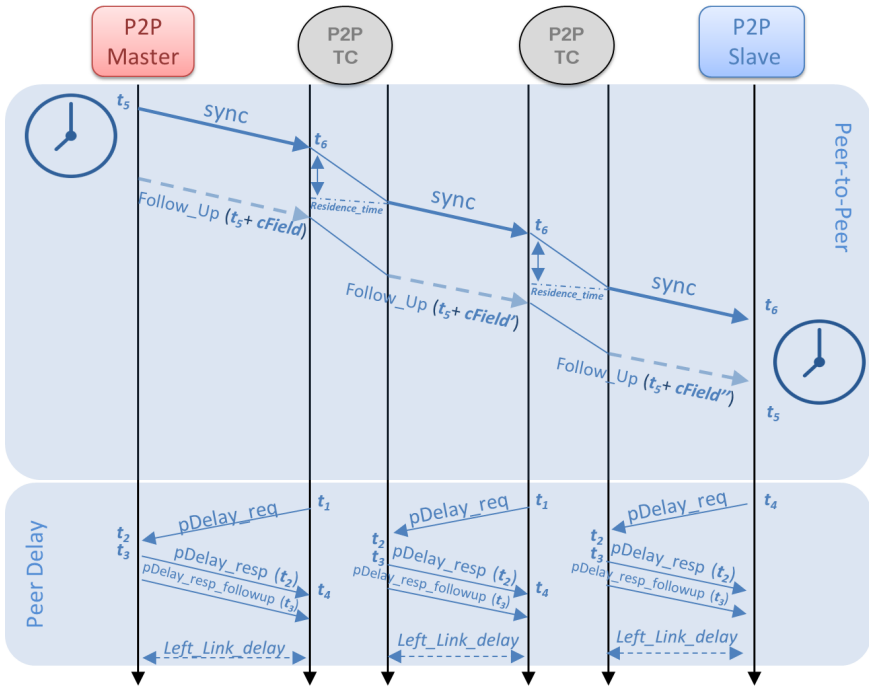

Figure 5.2
Clock offset measurement using WR TC/HYs. Master sends *Annonce*, *Sync*, and *Follow_Up* frames to the slave through TC/HYs. Each time a *Sync* goes through a TC/HY, its residence time inside the device is measured and, together with the link delay of the incoming port, it is added to the Correction Field of the next *Follow_Up*.

When the end-node receives a *Sync* message, it generates t_6 and waits for the *Follow_Up*, which contains t_5 and the total link

delay in the *cField*. By applying (5.4), the slave adjusts its local clock to the master reference.

$$offset_{ms} = (t_6 - t_5 + cField + delay_{adj} \tag{5.4}$$

Thanks to the development of these mechanisms, two new type of WR clocks are now available: P2P TCs and P2P HYs. The only difference between both of them is that a TC does not compute the $offset_{ms}$ to adjust its local oscillator to the master reference, while a HY does. A HY computes the $offset_{ms}$ after forwarding the received PTP frames to the next node and applies the changes to its local oscillator. In this way, HYs syntonize and synchronize to the master reference while TCs only syntonize.

Regarding precision problems caused by timestamps generation, note that WR uses hardware time-stamps that are generated immediately before/after a PTP frame is sent/received so that the uncertainty that could be introduced by the utilization of these time-stamps at higher levels of the OSI model can be reduced to negligible values.

Section 5.5.1 presents comparison results using both P2P and E2E approaches with BC, TC and HY in order to evaluate the stability of the synchronization performance in large-scale WR networks and how this affects to the scalability of the timing system.

Finally, the utilization of P2P TC/HYs is the first key step to adequate WR to SmartGrid, enhancing PPS jitter and offset skew results and also opening doors to the development of redundancy protocols such as PRP and HSR as [6] suggests for communication networks for power utility automation to improve fault tolerance.

5.3 WR-HSR: A SUB-NANOSECOND FAULT-TOLERANCE TIMING IMPLEMENTATION

This Section presents the implementation of the HSR protocol for WR devices to increase time distribution reliability. This implementation requires a ring network topology and is based on four features:

- the syntonization of the entire ring using the L1 frequency distribution approach.
- the synchronization of the entire ring using WR-PTP.
- the distribution of two time references at the same time (for both L1 frequency and WR-PTP).

- the development of a switchover mechanism able to change from an active to a backup time reference.

It is important to highlight that due to the fact that TCs do not synchronize [12] and thus, there is not synchronized output signal to be compared to the master reference, it might be more enlightening to use HYs to cover and evaluate the synchronization accuracy and precision of the HSR implementation.

Next Sections 5.3.1.1, 5.3.1.2 and 5.3.1.3 focus on the HSR timing implementation in detail.

5.3.1 *Redundant time distribution*

As previously described in 3.4.5, WR relies on two bases: the distribution of the clock frequency using a L1 distribution technology similar to SyncE, and the utilization of an extension of PTPv2 (WR-PTP) to compute the offset between the slave and the master clock. The former is related to the initialization of the WR link over the physical layer, and the latter to the synchronization itself. These processes have been modified to make them compatible with redundant ringed-topologies, as Fig. 5.3 depicts. Next Sections describe in detail these mechanisms together with the switchover mechanism.

5.3.1.1 *WR redundant link initialization model*

As previously mentioned, WR devices implement a slightly different version of SyncE from the standard. This custom version of SyncE differentiates WR from other PTP implementations [79] imposing a master-slave model to perform the syntonization process. In turn, this evokes a modification at the time of performing the syntonization for HSR protocol as explained in this Section. The adaptation of L1 WR syntonization process from tree to ring topologies has been one of the most challenging developments carried out within this thesis.

The standard process (Fig. 5.4, left side) starts with the dissemination of the clock reference over L1 from the master to the slave. When the slave receives this reference it locks its frequency to the recovered clock. Then, the slave also sends its clock to the master to estimate the asymmetry (5.5) of the communication channel [79]. This process is carried out using the following Ethernet frames: *WR_SLAVE_PRESENT*, *WR_LOCK*, *WR_LOCKED*, *WR_CALIBRATE*, *WR_CALIBRATED* and *WR_MODE_ON*.

If we now concentrate attention on technical details, as soon as a WR slave receives an *Announce* message from the master,

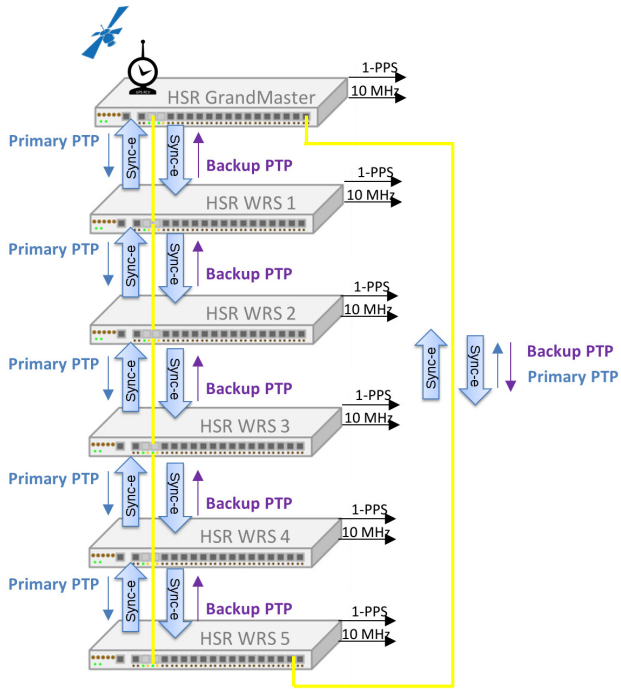

Figure 5.3
WR HSR setup using the L1 Synchronous Ethernet approach, two-step Hybrid Clocks exchanging PTP frames with *peerDelay* to measure the $\mathtt{delay}_{\mathtt{adj}}$ and Peer-to-Peer (P2P) to measure the $\mathtt{offset}_{\mathtt{ms}}$. Six WRSs form the ring where the one on the top of the figure is the *GM*. PTP is duplicated and sent over the two ports of the *GM* to the rest of the nodes so that slaves receive two PTP copies (primary and backup references). The frequency of the slaves is also locked to the adjacent nodes frequencies thanks to the L1 frequency distribution technology. WRSs synchronized outputs are 1 Pulse per Second (1-PPS) and 10MHz signals.

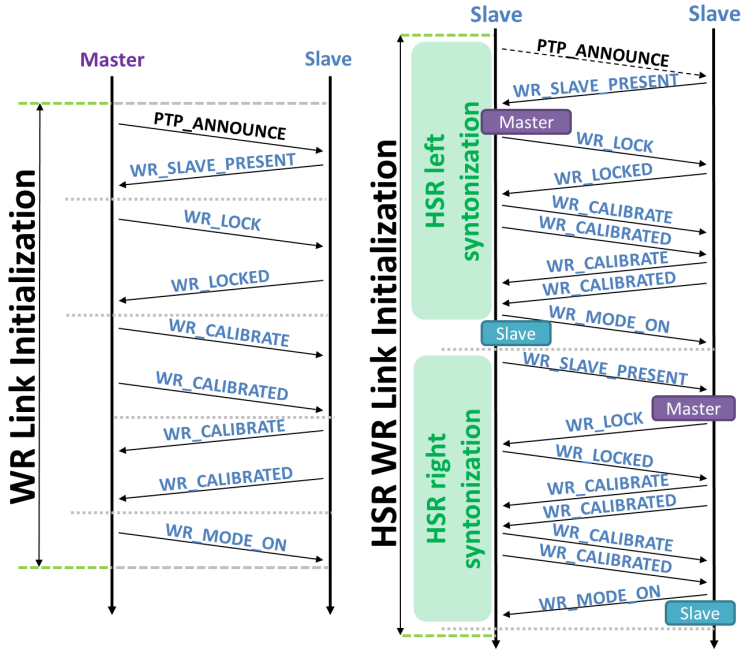

Figure 5.4
WR link initialization. Standard WR link initialization (left) is performed between a WR master and a slave, however, since the HSR WR link initialization (right) is carried out between two slaves, when a slave receives a *WR_SLAVE_PRESENT* it turns into master state to make possible the initialization of the slave. Once the first slave is syntonized, they exchange their master-slave roles to syntonize in the other way round.

it replies immediately with a *WR_SLAVE_PRESENT* in order to start the syntonization process. The master then sends a *WR_LOCK* frame to the slave to force its local oscillator to lock to the retrieved clock. Once the slave locks to the master frequency, it sends a *WR_LOCKED* to the master so that they start with the calibration parameters exchange (Δ_{tx_m}, Δ_{rx_m}) by sending *WR_CALIBRATE* frames. When the slave sends the *WR_CALIBRATED* frame to the master, this replies a *WR_MODE_ON* that means that the WR syntonization process has been successfully accomplished.

The main difference between the standard version (tree topologies) and HSR (ring topologies) is that all ports of the ring are slaves except for the two GM ports. Moreover, each device's frequency must be locked to its right and left clock references to guarantee L1 syntonization redundancy. For this reason, the syntonization process has been modified to make possible slave-slave links and also to be locked to two clock references at the same time.

In this approach (Fig. 5.4, right side), when a slave-slave link is established, one of the slaves becomes master and starts to send its local clock to the slave. Once the WR initialization process is done for the first slave, they exchange their roles in order to perform this process in the other way round. When this procedure is accomplished for the entire ring, all devices are doubly syntonized (clockwise and counterclockwise) to the GM clock frequency and hence, they all share *the same notion of time*. After this, the synchronization process starts to compute the clock offset between the master and the slave using WR-PTP to arrive the same notion of time.

5.3.1.2 *WR redundant P2P synchronization model*

Clock synchronization process is carried out by PPSi, the WR-PTP daemon. WR-PTP is used to compute the delay between two adjacent nodes, and the offset from the slave to the GM of the ring. The HSR implementation of WR-PTP has been developed following the recommendations suggested by the PTP profile attached to [7] for PRP/HSR. These recommendations can be summarized as follows:

- clocks can be configured as TC or HY.
- P2P is the recommended propagation delay mechanism to calculate the $\mathtt{offset}_{ms}$ (5.10).
- *peerDelay* shall be used to measure the delay between two adjacent nodes (5.7).

- PTP event frames sent from the GM (*Announce*, *Sync* and *Follow_Up*) must include a six-bytes HSR tag composed of the HSR Ethertype *ox892f*, ring path ID, LSDU and sequence number.
- PTP event frames sent from the GM (*Announce*, *Sync* and *Follow_Up*) are duplicated and sent in both directions of the ring. Each node shall compute them independently in order to follow two time references all the time.
- PTP frames are received on all nodes and forwarded to the next one. TCs/HYs do not generate their own PTP frames except for the *peerDelay* ones.
- The time that event frames spent in any device is call *residence_time* (5.8) and it is updated in every hop of the ring. The *residence_time* also includes the $\mathtt{delay}_{adj}$ (5.7) from the incoming port and it is used to compute the final $\mathtt{offset}_{ms}$ (5.10).
- *peerDelay* frames do not need to include any HSR tag, but the HSR implementation described in this thesis also inserts HSR tags in *peerDelay* messages, that are *Pdelay_Req*, *Pdelay_Resp* and *Pdelay_Resp_Follow_Up*.
- *peerDelay* is performed on both ports attached to the ring to compute the left and the right delay of the fiber link since these frames pass through the ring in both directions.

$$\mathtt{asymmetry} = \Delta_{tx_m} + \Delta_{rx_s} - (\Delta - \alpha\mu + \alpha\Delta)/(2+\alpha) \quad (5.5)$$

$$\mu = (t_2 - t_1 + t_4 - t_3)/2 \quad (5.6)$$

$$\mathtt{delay}_{adj} = \mu + \mathtt{asymmetry} \quad (5.7)$$

$$\mathtt{residence\_time} = \mathtt{sync\_egress} - \mathtt{sync\_ingress} \quad (5.8)$$

$$\mathtt{cField} = \mathtt{residence\_time} + \mathtt{delay}_{adj} \quad (5.9)$$

$$offset_{ms} = t_6 - t_5 + cField + delay_{adj} \tag{5.10}$$

$$\Delta_{offset_{ms}} = curr\_delay_{adj} - prev\_delay_{adj} \tag{5.11}$$

The first step to perform the ring synchronization is to form a cascade of HY WRSs. They all compute $offset_{ms}$ till they achieve sub-nanosecond accuracy. From this moment, they only need to follow the changes on the clock phase ($\Delta_{offset_{ms}}$) to remain synchronized.

Once they are synchronized, a last fiber link is attached from the last WRS of the cascade to the GM and thus, closing the ring network. At this point, all devices will receive PTP frames on their right port, which is considered the backup time reference. These PTP frames are also used to measure the $\Delta_{offset_{ms}}$ by taking into account only the clock phase variations the same way the primary reference does.

Fig. 5.5 shows how PTP is disseminated through the ring. *Announce*, *Sync* and *Follow_Up* event frames are sent from the GM to the all the slaves through its two ports. As previously indicated, the PTP received on the left port will be considered by default the primary time reference, and the right side one will be considered the backup time reference. In the meantime, each node computes the delay (5.6) between its adjacent nodes on both ports using the *peerDelay* mechanism based on 4 timestamps (t_1, t_2, t_3, t_4) by exchanging 3 messages: *pDelay_req*, *pDelay_resp* and *pDelay_resp_followup*. In order to compute this delay, the asymmetry of the channel estimated in the WR link initialization should be included in this measurement (5.5). Finally, the link delay can be represented as (5.7).

This delay is used to compute the *residence_time* (5.8) of each *Sync* message in every WRS. This *residence_time* is calculated as the difference between the time *Sync* was received (*sync_ingress*) and the time it was forwarded to the other port of the ring (*sync_egress*). Moreover, the delay of the incoming *Sync* port must be included in this *residence_time* and added to the *Follow_Up*'s *correction_field* (5.9) to accurately compute the $offset_{ms}$ (5.10).

Finally, t_5 and t_6 are used to compute the $offset_{ms}$. t_6 corresponds to the time *Sync* was received, and t_5 to the moment *Sync* was sent from the GM. The difference between these two times, the *Follow_Up*'s *correction_field* and the $delay_{adj}$ conform the difference between the master and the slave ($offset_{ms}$).

It is necessary to clarify that when a slave node is synchronized, the entire $offset_{ms}$ is not applied to the local oscillator.

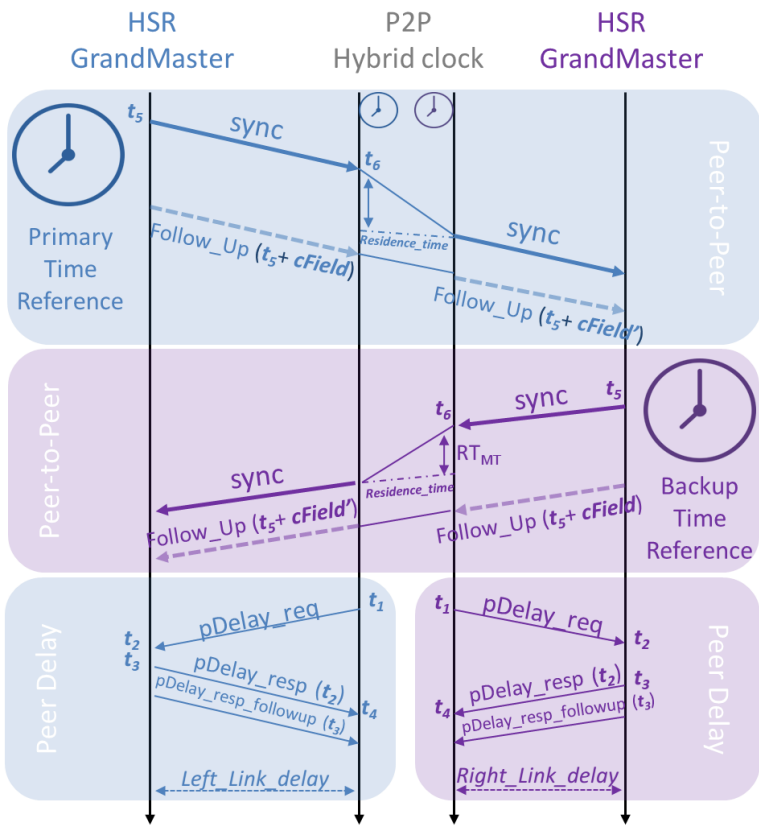

Figure 5.5
HSR P2P WR synchronization. The synchronization in a HSR WR ring network is carried out using P2P to measure the offset_{ms} of the clocks and *peerDelay* to compute the delay between two devices. Due to the utilization of HY, *Announce*, *Sync* and *Follow_Up* frames are forwarded from the GM to all the nodes of the ring, adding the *residence_time* of the *Sync* into the *Follow_Up correction_field*. Each node receives two copies of these frames (with different *correction_field*) that must be handled separately as primary and backup time references.

In this case, only the difference between the previous and the current $\mathtt{offset}_{\mathtt{ms}}$ ($\Delta_{\mathtt{offset}_{\mathtt{ms}}}$) computed with the primary time reference is applied to the local oscillator. On the backup side, $\Delta_{\mathtt{offset}_{\mathtt{ms}}}$ is computed too, being used to estimate an average that will be applied to the local oscillator in case the primary time reference is lost. When this happens, the switchover is triggered and the backup reference becomes the active one, thus applying the backup $\Delta_{\mathtt{offset}_{\mathtt{ms}}}$.

Next Section describes the recovery mechanism implementation and its behavior.

5.3.1.3 *Recovery mechanisms and components*

Switchover is the mechanism developed to change from a primary to a backup time reference. This procedure must occur as soon as possible since timing devices are only able to keep their oscillators synchronized to a timing source for a short period of time after losing the reference. After this time, synchronization is lost and it must be accomplished again from the beginning. Power plants standard states that this process must be carried out in less than 10 ms [6].

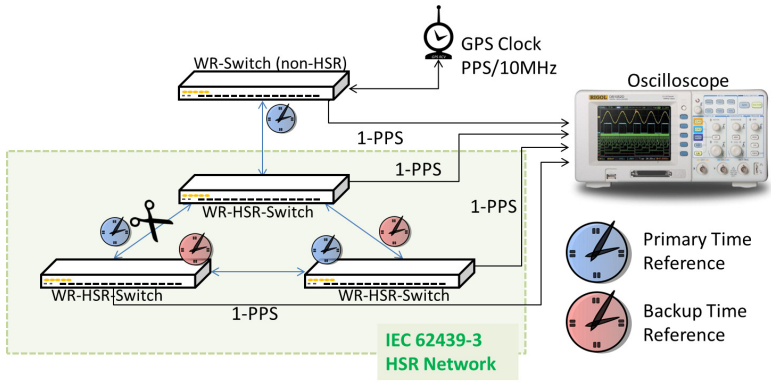

Figure 5.6
Switchover concept for redundant time distribution in ring topologies. Each node receives two copies of the same time information, one is considered the primary time reference (blue) and the other is used as the backup one (red). In case the primary time reference was lost due to for example, a link down, each node affected by this failure must switch over the backup reference. This procedure is called *switchover* and must occur in the minimum amount of time possible.

The time an oscillator is able to maintain the clock reference before becoming a free-running clock is called *holdover*. The switchover time must always be below the *holdover* of the oscillator's timing device. The *holdover* of a WRS is approximately 100 ms [80].

An initial version of this mechanism was previously developed for parallel networks at CERN [80]. The switchover process that changes from the primary to the backup time reference is done in the rage of µs [80]. This ensures that the switchover execution time never exceeds *holdover* and hence, synchronization is never lost. The basics of this mechanism reside in the development of the WR SoftPLL multi-channel approach, described below. Deeper details and results of this mechanism for parallel topologies can be accessed in [80].

MULTI-CHANNEL WR SOFTPLL FOR PARALLEL NETWORKS

The WR SoftPLL, previously introduced in Chapter 3, Section 3.5.1, has been extended to support multi-channel operation, fault detection based on phase error and majority voting, as well as seamless switchover.

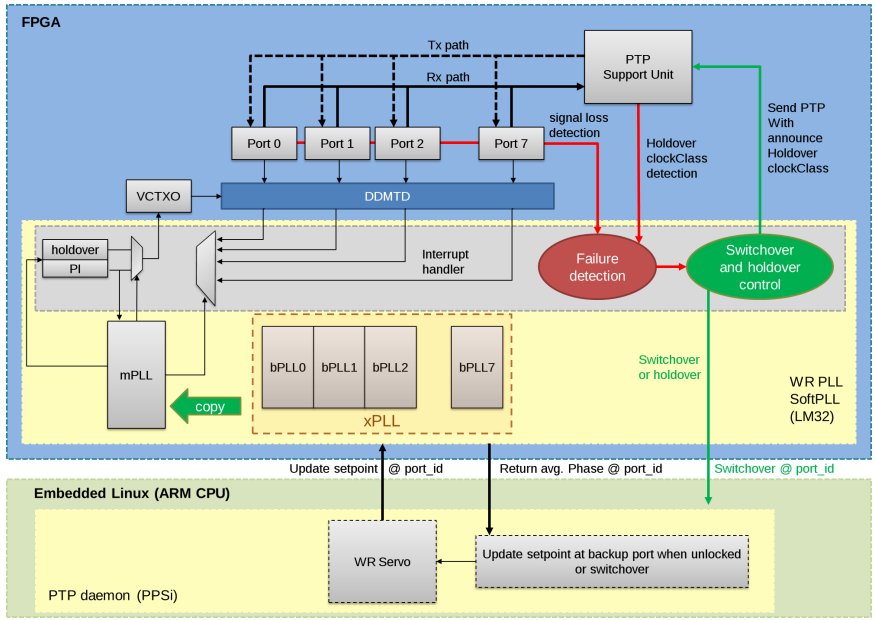

Figure 5.7
Switchover architecture developed for parallel networks. Extracted from [80].

The multi-channel WR SoftPLL is connected to the recovered clock signals from each port, as depicted in Fig. 5.7. Each input port is represented by a Backup PLL (bPLL) structure. This structure stores the necessary information to act as the Main PLL (mPLL), such as phase error, setpoint, source port number (port_id) and whether the port is locked and enabled. The structure also keeps track of short and long-term moving averages of

phase errors. These average values are also kept for the mPLL, as they are use to detect failure in multiple backup configurations.

In case of failure, switchover is executed as follows:

1. CPU Interrupts are disabled.
2. An intermediate holdover is applied to the mPLL and to the Helper PLL (hPLL).
3. The reference of the hPLL is switched to the new one and any history of phase error is cleared.
4. The relevant data from the bPLL is copied to the mPLL structure and the bPLL is disabled.
5. A flag is set to notify PPSi that switchover occurred.
6. The intermediate holdover is exited and interrupts are enabled.

MULTI-CHANNEL WR SOFTPLL FOR HSR NETWORKS

The adaptation of this mechanism to ring topologies maintaining the same features and functionalities is part and one of the objectives of this thesis. For the HSR implementation of this multi-channel mechanism only three channels are necessary as depicted in Fig. 5.8: the mPLL and two bPLLs. This is due to the fact that there are only two ports attached to the ring. The rest of the switchover process is similar to the one exposed before.

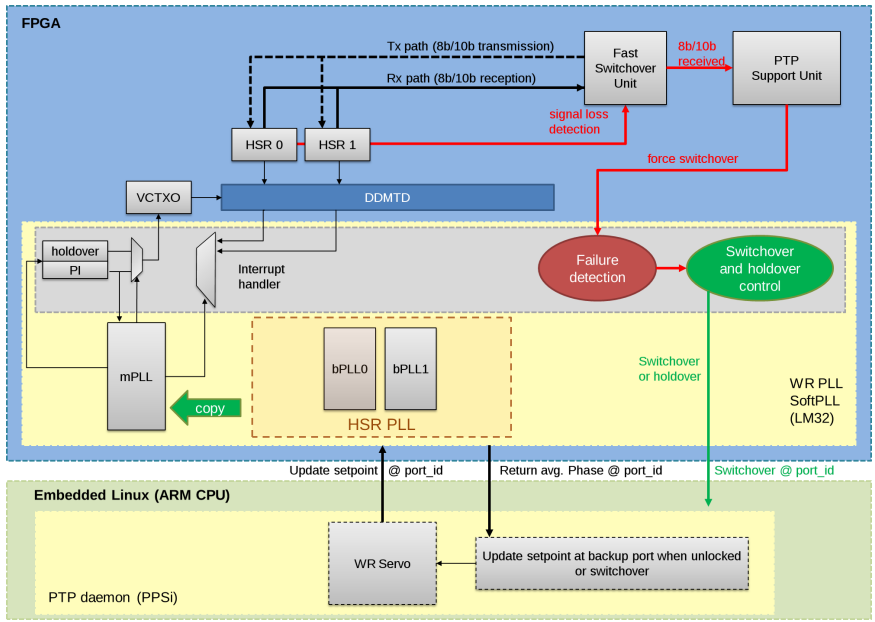

Figure 5.8
Switchover architecture developed for parallel networks. Adapted from [80].

As previously indicated, when the ring is closed all nodes will keep the track of the changes of both $\Delta_{offset_{ms}}$, but only the one from the primary reference port will be actually applied to the local oscillator. Regarding L1 syntonization, the procedure is quite similar, when the primary frequency reference is lost, the switchover mechanism will also start to discipline its local oscillator to the backup recovered clock. We must not forget that all the frequency distributed through the network is exactly the same, since the network topology is a ring.

The ring implementation of this protocol imposes the propagation of the switchover mechanism to all nodes as soon as possible. This is due to the fact that only the nodes directly attached to the device/link that is down will take notice of the failure. The rest of the nodes would start to run into *holdover* and after 100 ms, they will start to drift unless they receive a switchover alert, thus forcing the execution of the switchover process. For this reason, three dissemination procedures were evaluated to solve this issue:

- Sending multicast raw Ethernet frames from the CPU over the two HSR ports to all nodes.
- Setting a time threshold for PTP frames on each node, forcing switchover when a node stops receiving PTP frames over the time threshold.

- Sending Ethernet control symbols following the 8b/10b encoding system directly from the FPGA to the two HSR ports.

The first method was dismissed since it is closely related to CPU latency and raw sockets handling, exceeding the WRS *holdover* in all cases (hundreds of milliseconds). The second option was also discarded since by default, the WRS PTP daemon sends event frames every second, meaning that the minimum time to realize about a switchover situation leads to two seconds periods of holdover, which is completely unacceptable for the WRS. By increasing the PTP event frame-rate the switchover detection system was improved, obtaining switchover times of hundreds of milliseconds, but still over the *holdover* time. Finally, the implementation of the third method achieved noteworthy results, being hundreds of nanoseconds the time switchover was propagated and detected on each node. This is because the complete development was carried out inside the FPGA, thus preventing CPU and sockets handling. This feature is carried out by the Fast Switchover Unit (FFU), VHDL module explained in more detail in the hardware Section 5.4.3. The utilization of this module was a major change in the adaptation of the parallel version of the switchover mechanism previously described.

5.4 WR-HSR: A RELIABLE LOW-LATENCY DATA TRANSFER IMPLEMENTATION

Data distribution is considered critical in distributed control systems since a big transmission latency or the no reception of control data could lead to emergency situations. For instance, research facilities such as the GSI's particle accelerator control system, imposes very strict requirements for their control system, being only one frame the maximum number of frames that can be lost per year. Something similar occurs with industrial infrastructures, requiring similar results for the control system.

This Section describes mechanisms to increase data reliability for distributed critical applications. These methods solve and meet the strict requirements for data distribution in both industrial and research infrastructures thanks to the implementation of the HSR protocol also for data. All developments related to data dependability have been carried out in the WRS's FPGA, as VHDL IP Cores.

5.4.1 *HSR data transmission implementation*

HSR data networks are composed of three elements: Doubly Attached nodes (DANs), Redundancy-Boxes (RedBoxes) and QuadBoxes. DANs represent single nodes doubled attached to the ring, QuadBoxes connect several rings guaranteeing the redundancy features and RedBoxes connect non-HSR networks to HSR networks and vice-versa. Fig. 5.9 presents a full HSR network. The details of the functioning of the HSR are described below.

A DAN sends its data over both network interfaces and the destination DAN receives two copies of the same frame. The destination consumes the first frame and discards the duplicate.

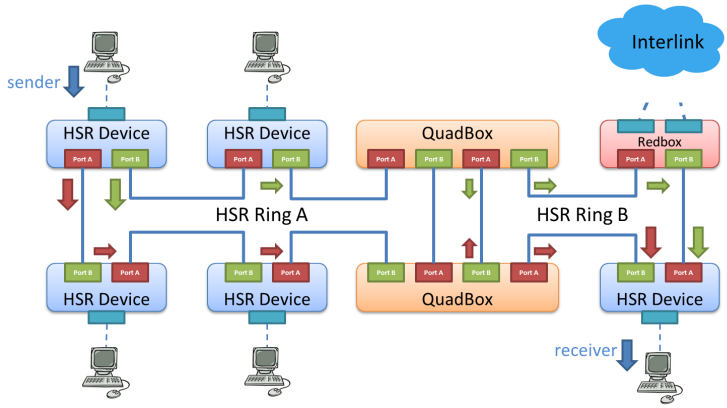

Figure 5.9
HSR network example, composed of DANs, RedBoxes and QuadBoxes. DANs are single nodes double attached to the ring topology, Quadboxes connect two ringed networks and Redboxes converts non-HSR into HSR frames and vice-versa.

Frames are identified by the combination of their source MAC address and their HSR tag. Duplicates can therefore be discarded by checking a 6-byte tag that must be part of data frames within the ring. The tag is inserted before the frame original Ethertype and is composed of the following fields: a 16-bit Ethertype identifier (0x892F), a 4-bit path identifier, a 12-bit frame size field and a 16-bit sequence number. The DAN is also responsible for removing the HSR tag and delivering the data frame to the upper layers.

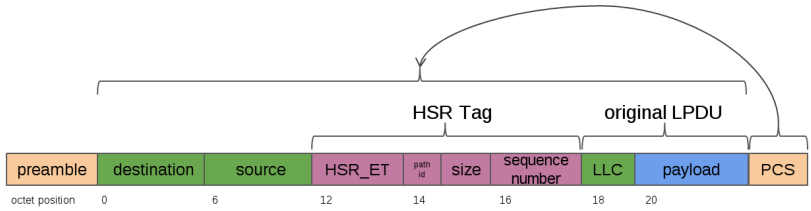

Figure 5.10
A HSR frame with a six-bytes HSR tag. A HSR tag is composed of the HSR Eth-erype (0x892f), the path id that identifies the port that transmits the frame, the size of the frame with the HSR tag and the sequence number.

Nodes that receive a multicast/broadcast frame or a frame for which they are not the destination will forward the received frames from one of the ring ports to the other, unless they are the node that injected it into the ring. This implies that if one of the devices or links of the network fails, data frames will still reach their destination through one of the halves of the broken ring.

The aforementioned IEC documents in [7] define a RedBox as a device that *acts as a proxy* for devices that only have one network interface and cannot interpret HSR tags. The HSR data implementation carried out in the WRSs will act as a RedBox, sending and receiving frames on behalf of a device that is not directly connected to the ring. The implementation of the HSR protocol has been carried out in the WRS.

From the outermost components towards the core, the most relevant parts of the switch architecture related to this design are the endpoints, the routing table unit and the switching core. The endpoints are a component that provide Ethernet medium access control capabilities. The set of 18 endpoints are all connected to the switching core. This module receives all data frames from the 18 endpoints and the CPU and routes them to their adequate destination. The information of where each frame should be routed is provided by the RTU, a component that is in charge of composing a table indicating to which port every MAC address belongs. This is accomplished by checking the source addresses of data coming from all endpoints. The operating system running in the CPU hosts the control software for these components and also runs the timing-related software.

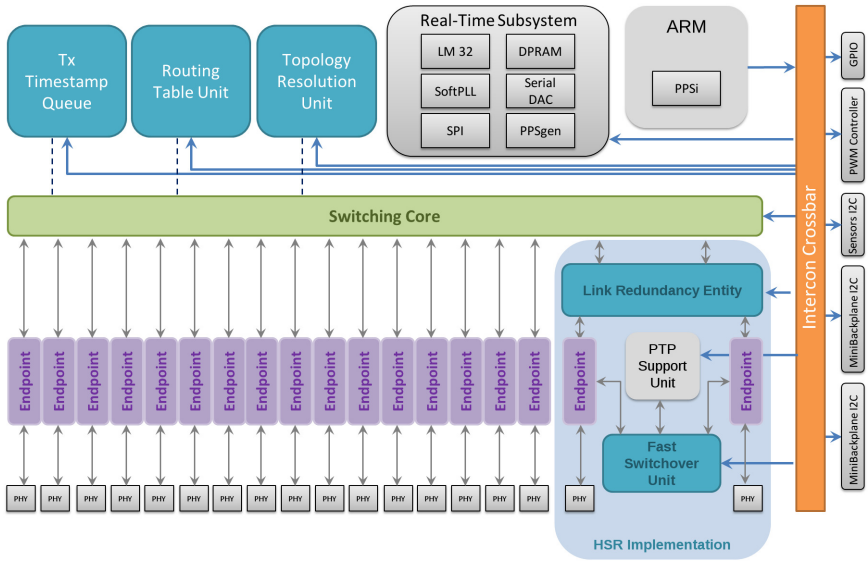

Figure 5.11
White Rabbit Switch Hardware-Gateware Architecture. This design includes the IP cores developed for the HSR protocol: the Link Redundancy Entity (LRE) and the Fast Switchover Unit (FFU).

The HSR data components that hold all the HSR data activities have been implemented as a VHDL core placed transparently in the preexisting switch architecture (Fig. 5.11). The core is called Link Redundancy Entity (LRE) according to [7]. Since it is only relevant for the operation of the ring, it only affects two of the eighteen ports of the WRS.

5.4.2 *Link redundancy entity IP core*

In the next Section the different units found inside the LRE (Fig. 5.12) are described in terms of function. The parts that are related to the incoming traffic (*Fast Forwarding Unit (FFU), dropper* and *untagger*) are firstly enumerated, while the ones that are related to outbound traffic (*tagger* and *arbiter*) are discussed in second place. The *Fast Switchover Unit* (FSU) is discussed in the last place since it works at a lower layer.

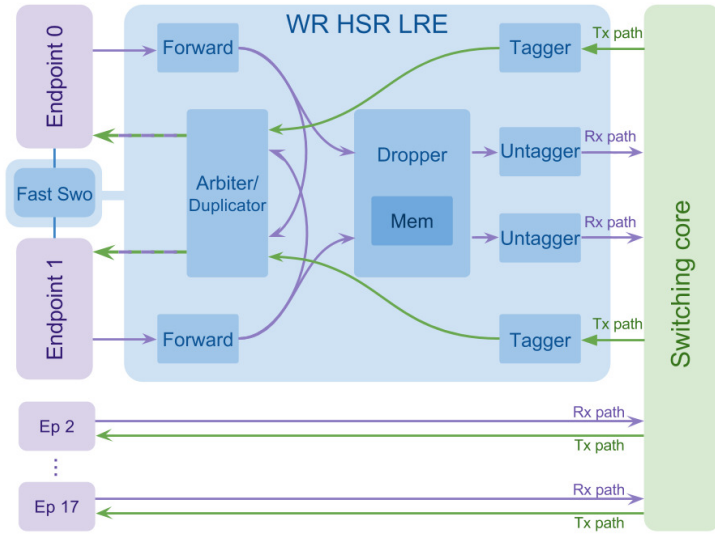

Figure 5.12
Link Redundancy Entity IP Core. Frames coming from the endpoints are forwarded by the Fast Forwarding Unit. At the same time they reach the Dropper where they might be discarded as duplicates. Otherwise, the Untagger removes the HSR tag. Frames coming from the switching core are tagged and duplicated so that it can be sent through both endpoints. The Fast Switchover Unit monitors the link and switchover mechanism status.

5.4.2.1 *Fast Forwarding Unit*

The FFU is the first unit through which an incoming packet goes. Its purpose is determining whether a frame has to be forwarded between HSR ports. If so, it generates a copy of the frame that will be sent to the other port, through the *arbiter*. This module introduces no additional delays in the incoming traffic path. The FFU acts in a cut-through manner, i.e., if the channel is available the data frame is bridged to the other port before the whole frame has been received.

5.4.2.2 *Dropper*

The *dropper* is the most complex module of the whole LRE. Its purpose is letting the first copy of each HSR frame pair reach the upper layers while discarding the duplicate one. This entails that there is the need to implement a memory with shared access for both LRE ports in which source MAC addresses and sequence number information is written for every incoming frame.

The *dropper* memory can accommodate up to 32 devices in the ring. The maximum delay after requesting a look-up operation is of 136 ns. The memory stores the full 16-bit sequence number of

the last received frame from a given device of the ring. The management of the frames is implemented in a window-like fashion: assuming that in a single ring there are no cases in which two frames coming from the same source can overtake each other, frames with a sequence number equal to the last accepted frame or slightly lower than the last accepted one will be discarded. When unexpected behavior is detected (e.g. out of order frames) the dropper notifies by issuing an alert that an upper layer application shall receive.

To compensate for newly introduced delays, the *dropper* is the component that generates RTU petitions on behalf of the HSR ports endpoints. This is necessary to prevent wrongly routed frames.

The analysis of data frames made by the *dropper* unit is also transmitted to the *untagger* to avoid replicating the same logic and improving time delays.

5.4.2.3 *Untagger*

The *untagger* removes HSR tags to provide the upper layers with standard Ethernet frames. The *dropper* commands the *untagger* whether a given frame has to be untagged. The operation is performed with zero additional delay.

5.4.2.4 *Tagger*

As its name suggests, the *tagger* appends HSR tags to outgoing frames. The tagger introduces the 6-byte tag described at the beginning of this Section.

The path identifier is used for discerning whether the frame is being forwarded clockwise or counterclockwise.

In order to avoid race conditions, taggers from both HSR ports share an auxiliary component that provide orderly monotonous sequence numbers. Currently, this module is designed to provide sequence numbers for a single device. Therefore, the switch can act as a RedBox for only one external device. This limitation does not apply to PTP frames that are handled in software.

5.4.2.5 *Arbiter*

The *arbiter* is the unit that duplicates frames and handles the demultiplexing from four incoming data paths to two outgoing ones. Two of the incoming data paths come from each *tagger*, and there are two additional sources of traffic due to the fast traffic forwarding between ports needed by the HSR protocol. The two outgoing data paths are connected to each of the HSR

ports endpoints. The *arbiter* implements four different buffers according to packet source and destination:

- traffic initially bound to the left port that must be duplicated to the right port and vice versa.
- traffic bound to the left port that must not be duplicated in hardware (for instance, PTP frames) and reciprocally, traffic bound to the right port that must not be duplicated in hardware.

This distinction is made for the sake of simplicity. Only buffers having data that does not have to be duplicated require control logic for buffer flushing.

The *arbiter* introduces variable delays that depend on frame size and are impossible to bypass, since its normal operation involves setting traffic back to avoid collisions.

It must be considered that the LRE module adds additional latency to the transmission/reception of frames compared with the standard version of the WRS. This additional latency is negligible, as summarized in the following scenarios:

- **Forwarding HSR traffic outside the ring**. This process requires untagging an incoming HSR-frame and forwarding it to a non-HSR port. It takes 20 + N/4 clock cyles (320 + (N/4)*16 ns), where N is the number of the HSR nodes that compose the ring.
- **Forwarding non-HSR traffic outside the ring**. In case the received frame is not HSR-tagged, the LRE module forwards the frame directly to the destination port. This operation takes 16 clock cycles (256 ns).
- **Inserting non-HSR traffic inside the ring**. The non-HSR traffic that is going to be inserted into the ring requires to be HSR-tagged and duplicated. This process takes 7 clock cyles (112 ns).

5.4.3 *Fast Switchover Unit (FSU)*

The switchover mechanism is one of the most important timing features of the HSR implementation. It is worth recalling that the synchronization is done from the left to right side of the ring, being the left reference the primary one for all devices.

Switchover runs locally when one of the links is disconnected or when a switchover alert is received on the primary time reference port. As soon as a node detects a link down on its left port,

it must inform the rest that this time reference is not valid anymore and that they must also switch over the backup reference immediately. Link down failure detection and alert disemination is performed by the FSU. FSU functionalities can be summarized as:

- Link failure detection.
- Force switchover to start locally.
- Send/forward the control message to force switchover to the rest of the ring.

When the FSU detects that one of the links attached to the ring is down, it forces the CPU and the SoftPLL to activate the switchover mechanism through the PTP Support Unit (PSU). At the same time, a control symbol in sent through the *alive* port to the rest of the nodes. This control symbol is created following the standard encoding system 8b/10b for the Ethernet physical layer so as to disseminate the failure state as fast as possible.

When a node with its two ports attached to the ring receives this control symbol, it forwards the symbol immediately and forces the switchover mechanism locally.

Forcing the switchover from the FPGA and forwarding the control symbol takes only few clock cycles (less than 50) at the very edge of the FPGA so that the dissemination time window is restricted to hundreds of nanoseconds.

5.4.4 *PTP Support Unit (PSU)*

The PSU developed in [80] and integrated in the HSR protocol, allows fast (~1ms) and standard-compatible notification about holdover between WR switches. It is a VHDL module that is placed between the Network Interface Controller (NIC) and the Switching-Core in the FPGA. When the PSU receives a switchover alert from the FSU, it notifies the SoftPLL and the PPSi (ARM) that the WRS is in holdover mode and that it must switch over the backup time reference.

PSU and FSU modules are only used for HSR timing reliability related to the switchover, not being used for data transmission.

5.4.5 *FPGA resource consumption comparison between HSR and non-HSR implementations*

Resource consumption and its impact on the system performance is an issue that must be taken into account for the development

of FPGA gateware. To this end, next Table 5.1 shows the differences between the standard WRS v4.0 version and the HSR WRS implementation. Note that the HSR protocol has been developed using the 8-ports WRS gateware version to streamline the synthesis process. For this reason, Table 5.1 shows the difference between 8-ports versions of the WRS.

Table 5.1
Resource consumption and impact comparison between HSR and non-HSR FPGA implementations for the WRS.

Resource	Standard WRS	HSR WRS	Total
No. of Slice Registers	36,358 (12 %)	39,881 (13 %)	301,440
No. of Slice LUTs	41,757 (27 %)	47,310 (31 %)	150,720
No. of RAMB36E1/FIFO36E1s	136 (32 %)	202 (48 %)	416
No. of BUFG/BUFGCTRLs	14 (43 %)	14 (43 %)	32
No. of DSP48E1s	3 (1 %)	3 (1 %)	768
No. of GTXE1s	8 (40 %)	8 (40 %)	20
No. of MMCM_ADVs	4 (33 %)	4 (33 %)	12
Minimum period	15.892 (62.925 MHz)	15.774 ns (63.395 MHz)	-
Maximum path delay between nodes	6.74 ns	8.30 ns	-

From the table above it can be deduced that, in terms of FPGA resources, both implementations are quite similar or identical for slice registers, look-up tables (LUTs), GTX transceivers, clock managers (MMCM) and minimum periods. However, the most relevant differences are the amount of memory used (32 vs 48 %) like FIFOS used within the LRE and FFU modules to insert HSR tags, forward, duplicate or drop frames. Finally, it must be considered the increment in the maximum path delay between two any nodes from 6.74 ns to 8.30 ns, caused by the introduction of the LRE and FFU modules.

5.5 RESULTS

This Section presents the results obtained for the three methods and mechanisms proposed in Sections 5.2, 5.3 and 5.4 of this thesis Chapter.

- A complete study of the stability and scalability of the WR protocol has been performed in order to evaluate the quality of the synchronization of WR devices with three different type of clocks combined with two delay measurement mechanisms: E2E BCs, P2P BCs, P2P TCs and P2P HYs (Section 5.5.1).

- Time accuracy results for the HSR protocol implementation. Regarding time distribution (Section 5.5.2), the expected outcome is reflected in the difference between the 1-PPS output between the GM to the slave ($offset_{ms}$) within three configurations: cascade, ring and reverse cascade, which happens after simulating a fiber cut between two WRSs, thus forcing the switchover mechanism.

- Evaluation of the distribution of data in HSR networks. The results (Section 5.5.3) are focused on data transmission latency, service availability and data bandwidth.

5.5.1 *WR stability and scalability results*

A setup composed of a cascade of 20 WR-LENs provided by Seven Solutions S.L. has been deployed to test WR timing stability and scalability using different clocks and delay measurement mechanisms. They are presented in a daisy-chain configuration over 0.5 m fiber links, in which N_{00} is the GM node and the rest are the slaves regardless of the clock type. Fig. 5.13 depicts this setup.

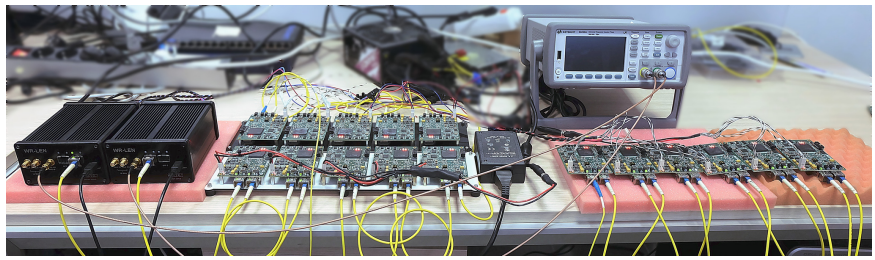

Figure 5.13
WR-LEN daisy-chain setup in lab.

Four scenarios have been deployed to evaluate the scalability of the approaches developed: E2E BCs with *Delay-Request*, P2P BCs with *peerDelay*, P2P TCs with *peerDelay* and P2P HYs with *peerDelay*. This evaluation consists in measuring the jitter of the 1-PPS output on each node (stability), and also in computing the skew of the offset (synchronization performance) between the master and the slave nodes of the daisy-chain gradually. The device used to compare 1-PPS outputs ($offset_{ms}$) is a Keysight 53230a Universal Frequency Counter/Timer[1], which presents a resolution of 20 ps. In order to provide reliable measurements, 1-PPS $offset_{ms}$ and jitter values have been measured for 180 s

1 http://www.alldatasheet.com/datasheet-pdf/pdf/863588/KEYSIGHT/53230A.html

for five times. Graphics and results are represented as the statistical measurement of the standard deviation (StDev) of the five average values measured for each hop per scenario.

5.5.1.1 *WR E2E BC using Delay-Request*

This is the WR default configuration, in which nodes are slave from their upstream node and master of the downstream. Each node generates PTP frames for the downstream node. The measurement of the delay is carried out using the WR standard *Delay-Request* approach. Nodes compute the $\mathrm{offset}_{\mathrm{ms}}$ using four timestamps.

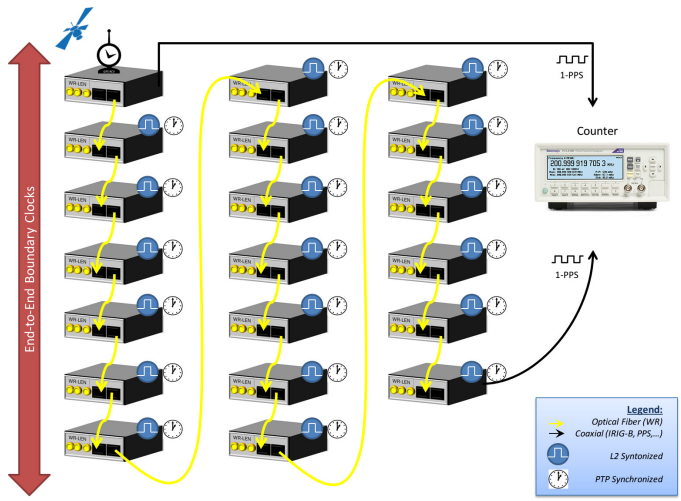

Figure 5.14
WR-LEN daisy-chain configuration composed of 20 E2E BCs using the default *Delay-Request* mechanism to compute the delay of the link. All nodes syntonize and synchronize to the master reference and generate their own PTP frames.

In this scenario (Fig. 5.14), all nodes of the cascade are syntonized and synchronized to the master reference.

The synchronization results are summarized in Fig. 5.18. Measures have been undertaken at hops N_{01}, N_{03}, N_{07}, N_{11}, N_{15} and N_{17}. There are no results for nodes N_{18} and N_{19} since they are not even able to syntonize to the retrieved frequency and thus, synchronization process can't be started. This is because of the degradation of the distributed frequency over the physical layer after 17 hops. After 17 slave devices, the quality of the received frequency does not meet the SoftPLL quality constraints to perform the syntonization process.

Regarding 1-PPS $\mathrm{offset}_{\mathrm{ms}}$, the results evidence that, in average, the sub-nanosecond accuracy is preserved till hop N_{10}, from N_{11} to N_{17}, in spite of being synchronized using WR, the aver-

age of the $offset_{ms}$ measured reaches a maximum mean value of 1.65 ± 0.075 ns (σ).

5.5.1.2 *WR P2P BC with peerDelay*

All nodes are slave from their upstream node and master of the downstream. Each node generates PTP frames for the downstream node. The measurement of the delay is carried out using the *peerDelay* approach as Fig. 5.15 depicts. All nodes compute the $offset_{ms}$ using six time-stamps.

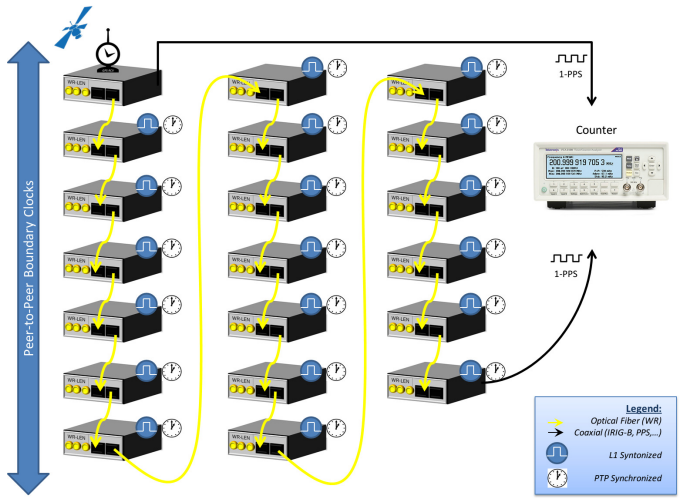

Figure 5.15
WR-LEN daisy-chain configuration composed of 20 P2P BCs using the *peerDelay* mechanism to compute the delay of the link. All nodes syntonize and synchronize to the master reference and generate their own PTP frames.

In this scenario, all nodes of the cascade are syntonized and synchronized to the master reference.

The synchronization results are summarized in Fig. 5.18. Measures have been undertaken at hops N_{01}, N_{03}, N_{07}, N_{11}, N_{15} and N_{17}. As in the previous case, there are no results for nodes N_{18} and N_{19} since they are not able to recover the L1 frequency from the previous reference either.

P2P BCs manifest similar but slightly worse accuracy results compared to E2E BCs ones. Sub-nanosecond accuracy is also preserved till hop N_{10}, from N_{11} to N_{17}, the averaged $offset_{ms}$ measured reaches a maximum value of 1.818 ± 0.038 ns (σ).

5.5.1.3 *WR P2P TC with peerDelay*

The first node of the cascade is a P2P Master clock and the last one is a P2P slave. Intermediate nodes are set as TCs so that

they only forward PTP frames from their upstream to the downstream port. The last node computes the $offset_{ms}$ using six time-stamps.

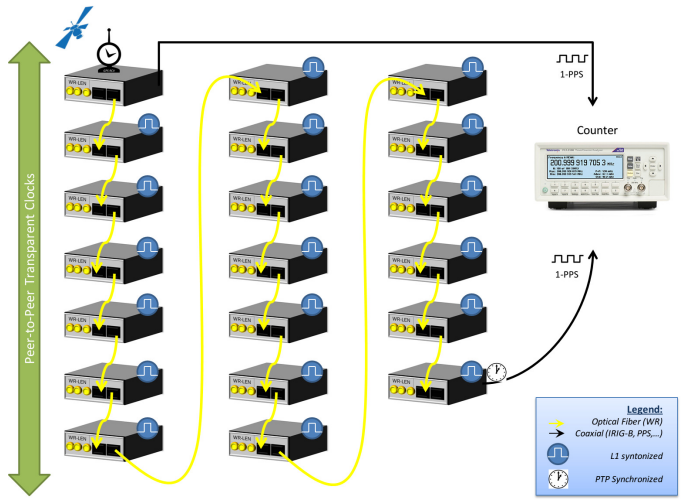

Figure 5.16
WR-LEN daisy-chain configuration composed of 18 P2P TCs, one P2P BC master device and a slave P2P using the *peerDelay* mechanism to compute the delay of the link. Only the last node synchronizes to the master reference while the rest ony forward PTP frames.

In this scenario, only the last node of the cascade is syntonized and synchronized to the master reference. Intermediate nodes are only syntonized as Fig. 5.16 shows.

The synchronization results are summarized in Fig. 5.18. Measures have been undertaken at hops N_{01}, N_{03}, N_{07}, N_{11}, N_{15}, N_{17} and N_{18}, too. There are no results for N_{19} since it was not possible to syntonize this node to the retrieved reference either, as occurred for E2E and P2P BCs scenarios.

P2P TCs present very satisfactory results, achieving accuracies below one nanosecond for the entire cascade, reaching a maximum averaged $offset_{ms}$ of 0.728 ± 0.179 ns (σ) for N_{18}.

5.5.1.4 *WR P2P HY with peerDelay*

The first node of the cascade is a P2P Master clock and the last one is a P2P slave. Intermediate nodes are set as HYs so that they forward PTP frames from their upstream to the downstream port and, in addition, compute the $offset_{ms}$ using the timing information of these forwarded PTP frames. The last node computes the $offset_{ms}$ using six time-stamps.

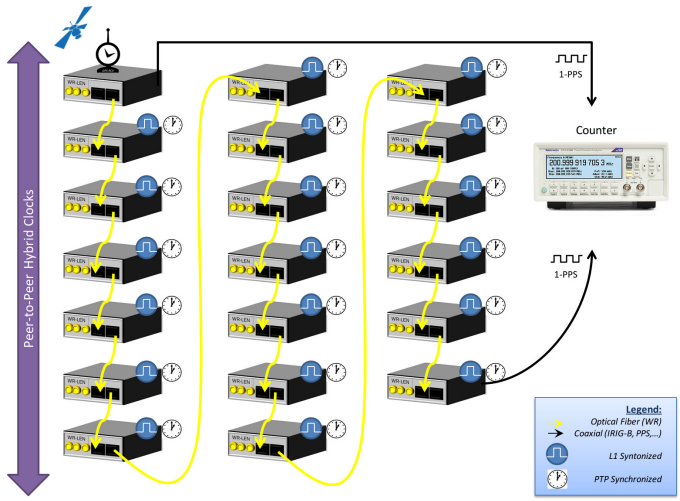

Figure 5.17
WR-LEN daisy-chain configuration composed of 18 P2P HYs, one P2P BC master device and a slave P2P HY using the *peerDelay* mechanism to compute the delay of the link. In this experiment, all nodes synchronize to the master reference and forward PTP frames.

In this scenario, only the last node of the cascade is syntonized and synchronized to the master reference. Intermediate nodes are only syntonized as Fig. 5.17 shows.

The synchronization results are summarized in Fig. 5.18. Measures have been undertaken at hopsN_{01}, N_{03}, N_{07}, N_{11}, N_{15} and N_{17}. There are no results for N_{18} and N_{19} since it was not possible to syntonize this node to the retrieved reference either, as occurred for E2E and P2P BCs scenarios.

P2P HYs $offset_{ms}$ measurements present even better results than P2P TCs, achieving a maximum averaged $offset_{ms}$ value of 0.559 ± 0.114 ns (σ) for N_{17}. Conversely, P2P HYs scalability is only capable to synchronize 17 nodes, instead of the 18 ones achieved by P2P TCs.

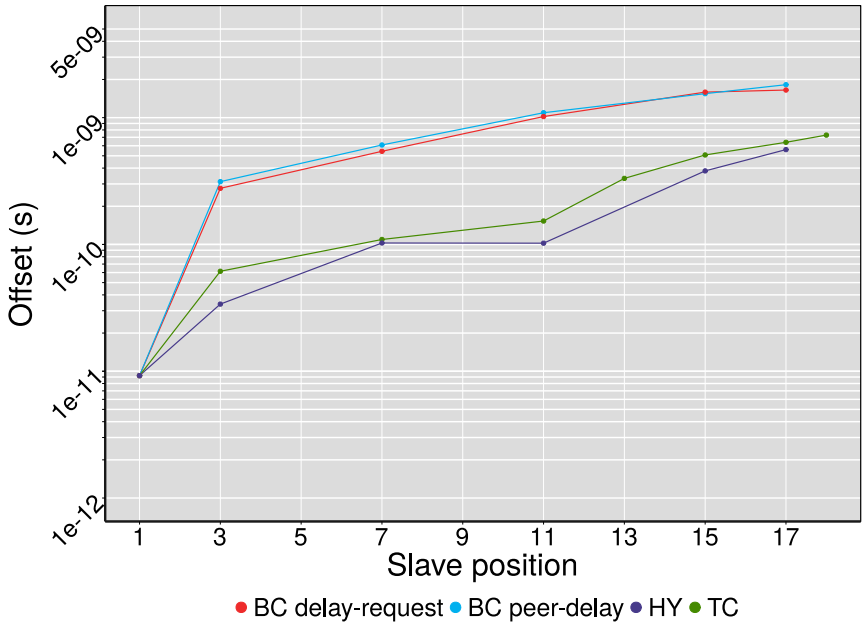

Figure 5.18
PPS offset mean measures for E2E BCs (red line), P2P BCs (blue line), P2P TCs (green line) and P2P HYs (purple line).

Table 5.2
$Offset_{ms}$ and STDEV (σ) comparison between E2E BCs, P2P BCs, P2P HYs and P2P TCs. These measurements have been performed by comparing the 1-PPS output of the master node to the 1-PPS of the last slave node.

Hop	E2E BC	P2P BC	P2P HY	P2P TC
N_{01}	0.002 ns ± 0.008 ns (σ)	0.009 ns ± 0.008 ns (σ)	0.009 ns ± 0.008 ns (σ)	0.009 ns ± 0.008 ns (σ)
N_{03}	0.277 ns ± 0.044 ns (σ)	0.031 ns ± 0.043 ns (σ)	0.033 ns ± 0.031 ns (σ)	0.061 ns ± 0.053 ns (σ)
N_{07}	0.542 ns ± 0.024 ns (σ)	0.608 ns ± 0.025 ns (σ)	0.102 ns ± 0.068 ns (σ)	0.109 ns ± 0.042 ns (σ)
N_{11}	1.019 ns ± 0.016 ns (σ)	1.093 ns ± 0.021 ns (σ)	0.102 ns ± 0.028 ns (σ)	0.152 ns ± 0.042 ns (σ)
N_{15}	1.589 ns ± 0.056 ns (σ)	1.545 ns ± 0.080 ns (σ)	0.380 ns ± 0.061 ns (σ)	0.507 ns ± 0.094 ns (σ)
N_{17}	1.65 ns ± 0.075 ns (σ)	1.818 ns ± 0.038 ns (σ)	0.559 ns ± 0.114 ns (σ)	0.638 ns ± 0.087 ns (σ)
N_{18}	not synchronized	not synchronized	not synchronized	0.728 ns ± 0.179 ns (σ)
N_{19}	not synchronized	not synchronized	not synchronized	not synchronized

Fig. 5.18, Fig. 5.19 and Table 5.2 summarizes all results for the four configurations.

It is clearly evidenced that the scalability of WR is not constrained by the degradation of PTP, it only depends on the degradation of the transmitted frequency over L1. This degradation is caused by the device circuitry, the frequency syntonization, and the synchronization process (clock phase correction). All these scenarios develop the same circuitry and performs the L1 syntonization process, however, P2P TCs do not synchronize, thus skipping the phase correction adjustment of the local oscillator.

For this reason, the degradation of the frequency in P2P TCs is a bit less than for the other clocks, so P2P TCs are a bit more scalable (one more hop).

In terms of synchronization accuracy, results are very promising. We can assume that all clock implementations are capable of synchronizing to the master reference with an accuracy below 1 ns till hop N_{10}. However, only P2P TCs and P2P HYs are capable of guaranteeing sub-nanosecond accuracies up to 17 and 18 slave nodes, respectively. By this, we have demonstrated that WR offers better accuracies using a P2P approach instead of the originally developed E2E one.

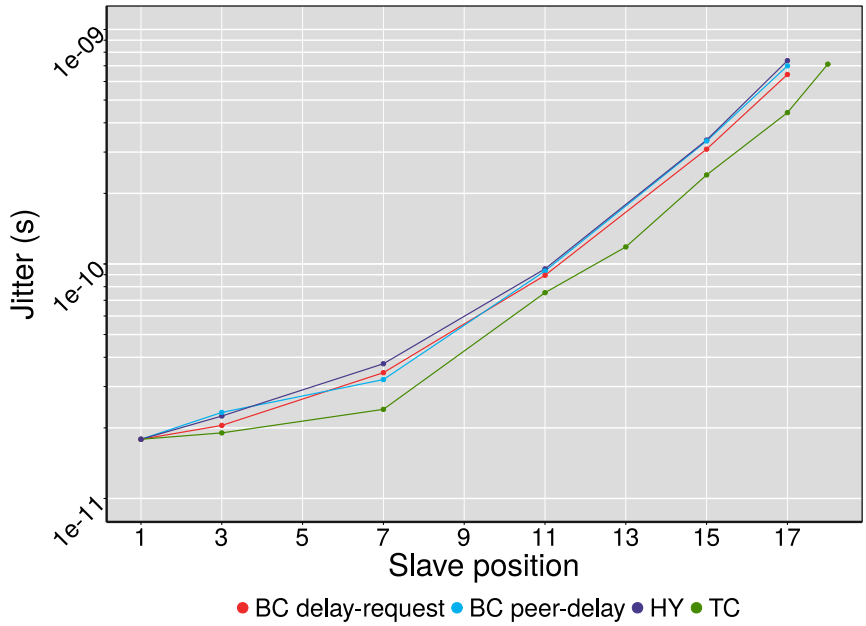

Figure 5.19
PPS jitter mean measures for E2E BCs (red line), P2P BCs (blue line), P2P TCs (green line) and P2P HYs (purple line).

Finally, Fig. 5.19 presents the WR stability results by the measurement of the 1-PPS jitter for each type of clock and delay model. In this case, the best results are provided by P2P TCs, achieving an improvement in the stability of the signal of approximately 100 ps. By contrast, E2E BCs, P2P BCs and P2P HY lines draw very similar values, which affirm the hypothesis that the adjustment of the phase difference between two clocks increases jitter.

5.5.2 *Timing redundancy results*

The setup used for the timing experiments is composed of 6 WRSs forming a ring with two ports per node attached to each other using optical fibers (Fig. 5.20). The SFP transceiver modules used are the Axcen Photonics AXGE-1254-0531 1.25 Gbps single fiber Bi-directional, which calibration parameters are 180667 ps for Δ_{tx} and 148735 ps for Δ_{rx}, and AXGE-3454-0531 1.25 Gbps single fiber Bi-directional, which calibration parameters are 180667 ps Δ_{tx} and 148735 ps for Δ_{rx}. These parameters are the same for all HSR nodes and they are used to compute the asymmetry of the channel as already said in 5.3.1.1.

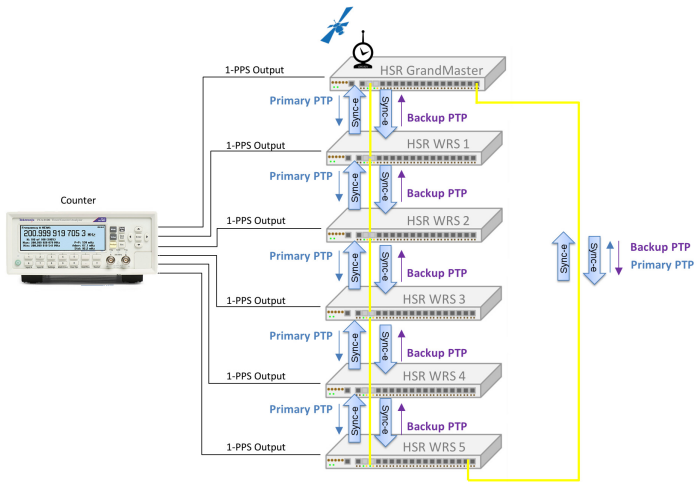

Figure 5.20
WR-HSR timing setup. It is composed of 6 WRSs in a ring topology where one of them is the GM and the rest are doubled synchronized (left and right sources) to it. The 1-PPS output of each WRS is connected to a Time Counter to compute the offset between 1-PPS outputs in cascade, HSR-ring and reverse cascade (after switchover) configurations.

The quantifiable measurement used to compare the accuracy of the synchronization has been done by computing the offset between the 1-PPS output signals of all slave nodes to the GM for 300 s (300 offset samples). The measurement instrument for the 1-PPS offset is the same counter used for the scalability results.

These measurements have been done in three different scenarios:

- A cascade of 6 WRSs. First step where all the nodes are synchronized to the primary time reference.
- A closed ring. All nodes receive two time references. The objective of this scenario is to verify whether dealing with two

L1 frequency and PTP references affect the synchronization stability.

- A switchover simulation on real-time. This scenario verifies the functioning of system before, during and after forcing the switchover mechanism.

5.5.2.1 *Cascade*

All nodes are synchronized one by one from the GM to the last WRS. This scenario shows (Fig. 5.21) the initial WR synchronization accuracy for all nodes. For this configuration, the results (Table 5.3) for the six WRSs present an $offset_{ms}$ of hundred of picoseconds, very similar to the standard version of the WR protocol.

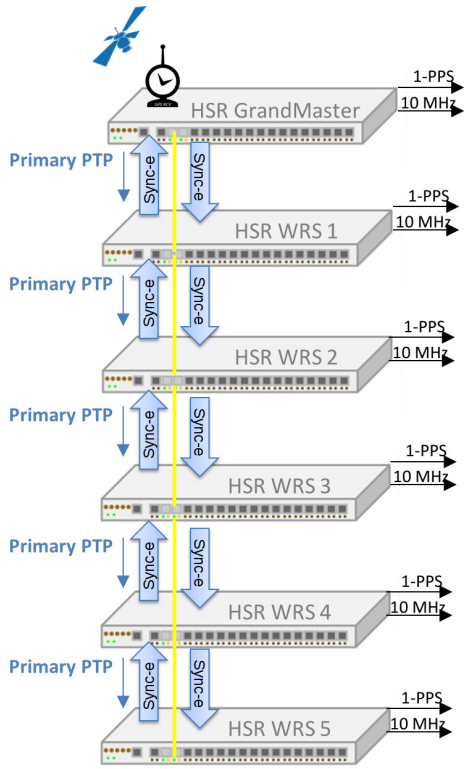

Figure 5.21
HSR timing cascade setup. This setup is formed of six WRSs in a daisy-chain configuration. This setup represents the first step to form a HSR ring, where only the primary time reference is sent from the GM to all the slaves.

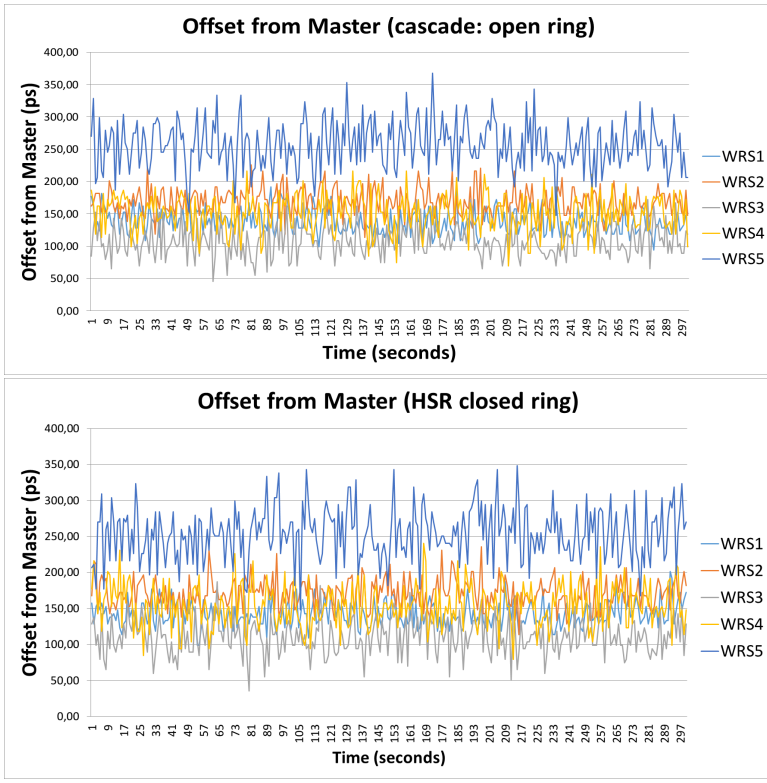

Figure 5.22
Offset skew per setup (cascade and ring topologies). The first image shows a max. $\mathrm{offset}_{\mathrm{ms}}$ for all switches forming a cascade of 256 ps. The second one presents very similar $\mathrm{offset}_{\mathrm{ms}}$ values after closing the ring.

5.5.2.2 *Ring*

Once all nodes are synchronized to its left port reference, a last optical fiber is connected from the last WRS to the first one closing the ring. Then, all slave nodes are computing both $\mathrm{offset}_{\mathrm{ms}}$ and are also locked to two frequency references. This scenario shows whether following two time references affect or not and, in which proportion, the local oscillator.

Table 5.3
$Offset_{ms}$ Comparison between cascade and ring configurations

	Cascade	Ring	ABS(Cascade - Ring)
WRS1	138 ps	141 ps	3 ps
WRS2	169 ps	171 ps	2 ps
WRS3	109 ps	111 ps	2 ps
WRS4	150 ps	154 ps	4 ps
WRS5	256 ps	254 ps	2 ps

For this scenario, in spite of making the WRS to deal with two frequencies and PTP instances at the same time, the local oscillator seems not to be affected by this fact because $offset_{ms}$ varies in few picoseconds in regard to the cascade configuration (Figure 5.22). Table 5.3 summarizes the differences between these two setups.

5.5.2.3 *Reverse cascade*

Since the implementation developed by the authors synchronizes counterclockwise, the simulation of a link down will be performed as the worst case scenario (WCS) possible, which is removing the first fiber connected from the GM to the first WRS1. This case represents the WCS (Fig. 5.23), where all WRSs should switchover from the primary to the backup time reference. For this scenario, a fiber cut has been simulated after 150 s of data acquisition.

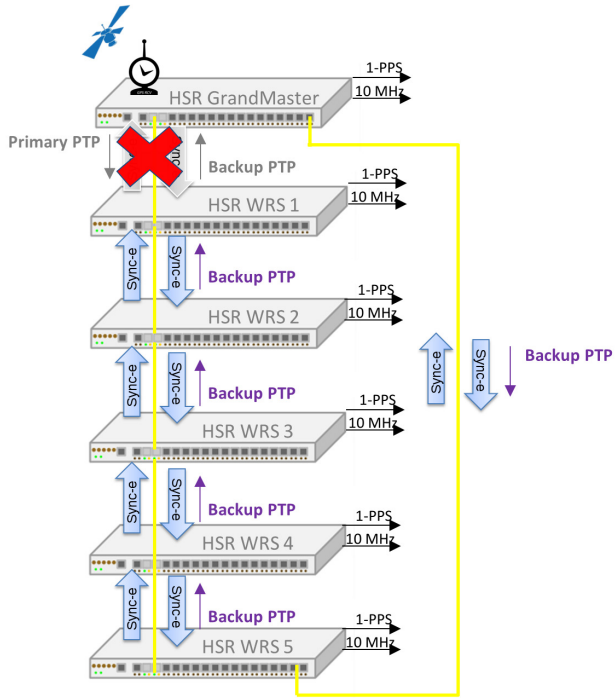

Figure 5.23
HSR timing reverse setup. This setup is formed of six WRSs in a reverse daisy-chain configuration. This setup represents the final HSR scenario, the resulting time synchronization using only the backup reference after forcing the switch-over mechanism in all nodes.

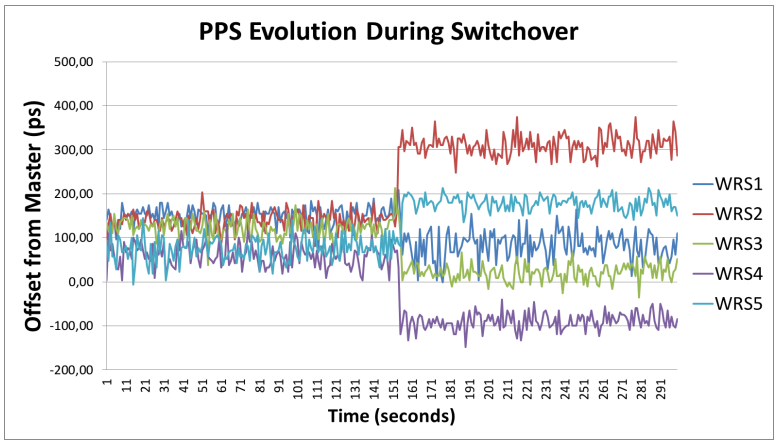

Figure 5.24
Offset skew during a switchover scenario. This figure represents the evolution of the synchronization performance before and after switching over the backup reference (300 s). The fiber cut is simulated after 150 s. The results ensure a synchronization accuracy below 1 ns during the entire process with a maximum phase shift of 170 ps.

Table 5.4
$Offset_{ms}$ Evolution before and after switchover

	Before Switchover	After Switchover	PPS Shift
WRS1	150 ps	85 ps	65 ps
WRS2	140 ps	310 ps	170 ps
WRS3	122 ps	20 ps	102 ps
WRS4	61 ps	−87 ps	149 ps
WRS5	72 ps	178 ps	105 ps

The results for $offset_{ms}$ after switching over are very encouraging. All devices remain synchronized below one nanosecond and the synchronization of their oscillators do not suffer any significantly phase shift. In order to evaluate correctly the switchover procedure, a fiber cut has been simulated after 150 s of data acquisition. First 150 samples of Figure 5.24 depict the $offset_{ms}$ for all nodes before the switchover mechanism is forced, the other 150 samples show the evolution of these offsets after switching over the backup time reference. Table 5.4 shows these results, being the maximum phase shift occurred 170 ps.

With these results, authors want to demonstrate that the implemented HSR protocol for WR maintains the main WR feature, the sub-nanosecond accuracy, even after switching from a primary time reference over a backup one, being the average $offset_{ms}$ measured hundred of picoseconds.

5.5.3 *Data redundancy results*

The setup for the data experiments is based on the one used for timing results in Section 5.5.2. Two general purpose PCs are connected, one each, to two adjacent WRSs with the help of copper SFP interfaces in the side of the switches. The WRSs act as RedBoxes for the PCs, allowing them to communicate throughout the ring (Fig. 5.25). This way, there is a three-hop link between the PCs via the shortest path of the ring and a seven-hop link via the longest one.

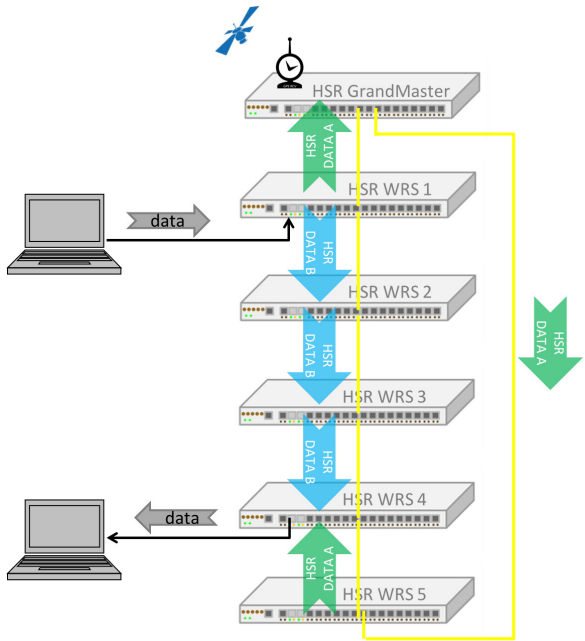

Figure 5.25
HSR unicast data transmission example. Unicast data is transmitted from the first PC to the second one through a 1000Gbps Ethernet copper cable. When PC data frames reach the first WRS, this acts as a Redbox by duplicating the frames (HSR DATA A and HSR DATA B) and sending them out through the two ports attached to the ring. Once these frames are received on the WRS to which the destination PC is connected, the WRS removes the HSR tag and forwards the first copy of frame to the PC. The second copy is discarded.

In order to evaluate the performance of the WRS HSR ring, three metrics have been chosen: latency, bandwidth and frame loss.

- Latency measurement is realized by comparing the transmission and reception timestamps of a frame that travels throughout the whole ring and comes back to the WRS that injected it (Fig. 5.26). To account for the latencies of different amounts of WRSs, several rings with varying amounts of hops from two to six are tested. In order to perform a precise comparison between the data latency for standard WRSs and HSR-WRSs, latency measures have been performed at hardware level (FPGA gateware) using Xilinx Chipscope Tool[2] to count the number of cycles between the arrival of the frame, and its departure. Different frame sizes have been used to perform these tests: 64, 128, 512 and 1024 bytes.

2 https://www.xilinx.com/itp/xilinx10/isehelp/ise_c_process_analyze_design_using_chipscope.htm

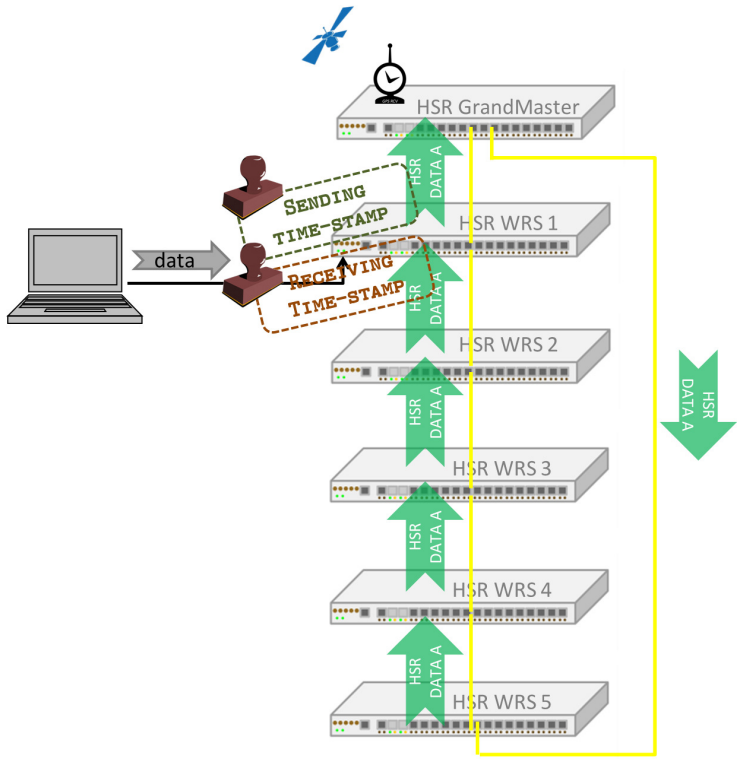

Figure 5.26
WR-HSR latency setup. This configuration is used to measure the total latency of a frame time travel through the ring. To this end, a frame is sent from a PC connected to a WRS, the WRS time-stamps the sending time of the frame and generates another time-stamp when the same frame returns to the WRS.

- Bandwidth has been determined by exchanging bursts of UDP frames between the two PCs using the network software tool found at [81] as Fig. 5.25 depicts.
- Frame loss is measured in two ways: firstly, the analysis of the bandwidth tests reveal whether frames have been lost during the whole test process. But, in addition, the ring must support *zero-delay reconfiguration*, that is, no frame must be lost after failure of one link or device. This is proved by disconnecting one of the links during bursts of ping floods. In this scenario, Fig. 5.25 has been also used as the default setup.

Latency results, as can be seen in Table 5.5 show that the simplest case (two WRSs) entails a latency of 1880 ns. From that point, every additional WRS means an addition of 1430 ns to the total time of flight. These delays are decomposed in three parts: electronic circuitry fixed delays (around 450 ns), forwarding de-

lays between HSR ports (1008 ns) and the transmission delays due to the speed of light in the fiber (1 ns every 20 cm).

Table 5.5
Data latency results for rings formed by 2, 3, 4, 5 and 6 HSR WR Switches.

	2 WRSs	3 WRSs	4 WRSs	5 WRSs	6 WRSs
Latency	1,88 µs	3,31 µs	4,74 µs	6,16 µs	7,59 µs

In contrast to the standard version of the WRS, the reduction of the latency in the HSR implementation presents a significant improvement of 50% (from 3 µs [82] to 1.4 µs). This is due to the fact that HSR forwarding capabilities deploys the FFU module previously described in Section 5.4.2.1, that resends the incoming frame immediately after it enters the FPGA, avoiding the standard internal WRS switching logic. Table 5.6 presents the results for standard WRSs' data latency.

Table 5.6
Forwarding latency in a standard WRS vs. forwarding latency in a WRS with HSR capabilities implementing a FFU

	64 bytes	128 bytes	512 bytes	1024 bytes
Switching Core	1,70 µs	2,22 µs	2,18 µs	2,17 µs
FFU Module	1,1 µs	1,1 µs	1,1 µs	1,1 µs

In terms of bandwidth, the results summarized in Table 5.7 show that the ring of WRSs under test can reliably cope with data up to 680 Mbps. From that point on, a fraction of the frames is lost. Note that the physical duplication of all data frames required by HSR leads to the insertion of twice as much frames in the ring as the amount generated by the PCs. For this reason, only 50% of the bandwidth can be guaranteed [83].

Table 5.7
Data bandwidth results for a 6 HSR WRS ring.

Frame Size	Bandwidth	Frames sent	Frames recv	% lost
288 bytes	99.5 Mbps	1113844	1113844	0.0
288 bytes	199 Mbps	2229670	2229670	0.0
288 bytes	299 Mbps	3351341	3351341	0.0
288 bytes	400 Mbps	8126971	8126971	0.0
288 bytes	498 Mbps	5572979	5572979	0.0
288 bytes	603 Mbps	6798765	6798765	0.0
350 bytes	682 Mbps	6664160	6664160	0.0
372 bytes	700 Mbps	5306220	5170491	2.56

During the data tests, link failures were thoroughly simulated by breaking the ring in both paths while these were exchanging bursts of Internet Control Message Protocol (ICMP) ping messages. The results of these experiments show that no frames are lost due to any single link failure, complying with the *zero-delay reconfiguration* requirement.

5.6 CONCLUSION

The approach presented by the authors is based on the implementation of the HSR protocol for WR devices. For the first time, a redundancy protocol has been implemented for the WR technology. This ensures the sub-nanosecond synchronization accuracy for the devices attached to the ring (avg. $\mathrm{offset}_{\mathrm{ms}}$ of 200 ps) together with the possibility of having two time references: a primary time source and a backup reference, used in case of failure. These timing results improve significantly previous works of other standard HSR PTP implementations, such as [73–76], that present a maximum accuracy between 30 ns and 100 ns.

The switchover mechanism adapted from parallel to ring topologies guarantees the seamless impact of changing from the primary time reference to the backup one. This process takes only few µs, remaining below 100 ms, the *holdover* time of a WRS. In addition, the re-syntonization and re-synchronization problem has been solved by forwarding the control symbol as soon as possible at the physical layer, forcing the switchover of the rest of the ring in hundreds nanoseconds. By this, it has been possible to maintain the entire ring synchronized below one nanosecond, even for the WCS (avg. maximum drift of 170 ps).

Regarding data, the duplication of frames of any of the services including Smart Grid control events like GOOSE (Generic Object Oriented Substation Events), GSSE (Generic Substation State Events) or SMV (Sampled Measured Values), guarantees their reception on the destination node in spite of existing a node/path failure for a maximum bandwidth of 680 Mbps, thus increasing the availability of the ring services. In addition to this, the FFU developed enhances the transmission of data frames through the ring, reducing each hop's latency from 3 µs [82] (WRS standard latency) to 1.4 µs. In addition, the insertion of frames in the ring (redbox behavior) takes around 3µs.

WR scalability have been improved by the development of P2P hybrid and transparent clocks, ensuring sub-nanosecond accuracies for daisy-chain configuration up to 17 abd 18 nodes respectably, in contrast to E2E BCs, only scalable to 10 nodes. In terms of stability and signal quality, the development of P2P TCs has been able to reduce 1-PPS jitter measures in approximately 100 ps. At the same time, the utilization of P2P also guarantees compatibility with Smart Grid timing networks as previously described in Section 5.2.

All these new integrated features in WR devices allow now the fulfillment of the requirements demanded by scientific, telecom and industrial networks. In addition, they make possible for WR to be ready for industrial infrastructures and, in particular, for Smart Grid and WAMS where time accuracy and the availability of the services of the network become prominent, as [84] highlights for the future of synchronized PMUs.

Finally, the approaches described in this Chapter, together with the functionalities previously developed in Chapter 4 for leaf-nodes, we have been able to cover reliability features of a entire Smart Grid infrastructure, fulfilling both inter-core and distributed inter-process communication requirements for dependable DCS. Next Chapter 6 presents a real Smart Grid scenario where all these concepts have been integrated in order to form a fully DCS, taking into account communication reliability requirements, safety, timing interoperability and scalability. This work was presented as one of the main use cases developed for the EMC^2 project.

6

SYNCHRONIZED LOW-LATENCY DETERMINISTIC NETWORKS: A SMART GRID USE CASE

We must use time as a tool, not as a crutch.

— John F. Kennedy

INDEX

Previous Chapters described several methods to improve the reliability of the different elements that conform distributed control systems such as Smart Grid and WAMS. Chapter 4 focuses on providing these reliability features to the end nodes of the grid, such as signal analyzers and data acquisition devices. Chapter 5 refers to the methods and mechanisms developed to increase the reliability of synchronization systems at both core and boundary of the network. In the same way, the availability of the services provided within the network together with the transmitted control data are crucially important for the grid.

The state-of-the-art presented in Section 3.3 describes the current directions followed by emerging Smart Grid trends regarding time synchronization and system criticality. This Chapter integrates the concepts of dependable multi-core architectures for end nodes, and the relevance of having a redundant high-accuracy timing technology to synchronize the entire grid using

the same global time reference. In addition, the feasibility of following two time references at the same time, together with the possibility of switching from one to the other with the less impact possible to the timing system become indispensable.

The rest of this Chapter describes the solution developed that integrates all these features for a concrete WAMS use case, a Substation Automation System (SAS), being a clear example because of its demanding scalability and time synchronization needs. It is organized as follows. Next Section 6.1 introduces the new Smart Grid needs and fundamentals for the integration of timing devices and safety-critical applications to create a new era of cutting-edge control systems for the industrial domain. Section 6.2 describes the SAS use case. Section 6.3 contains the implementation whilst Section 6.4 presents the analysis and results of the system accuracy, reliability, scalability, safety and security. Finally, Section 6.5 describes the discussion and conclusions of these results.

6.1 MOTIVATION

In SAS domain, precise time synchronization is required to have accurate clocks for system control, data acquisition and forensic analysis. Time synchronization is especially important for PMUs and event's time-stamping and interlocking control loops [85, 86]. For this reason, timing information disseminated trough the network must be considered critical for DCS, since the loss of these frames or the utilization of imprecise low accuracies might lead to a malfunctioning of the entire system. Typical time synchronization technologies in SAS are GPS, IRIG-B and 1-PPS. Other timing protocols over Local Area Networks (LAN) are NTP, SNTP, PTPv1 (IEEE1588-2002) and PTPv2 (IEEE 1588-2008), which is required mostly for IEC 61850-9-2 Process Buses or IEEE C37.118-2005 Synchrophasors implemented in PMU devices [44]. In order to correctly analyze power system disturbances, a very accurate time-stamping of events is necessary. Some RTUs have the availability to time-stamp events with a precision of ms [87]. Classically, the specific requirement for time tagging of events is 1 ms accuracy [6], which is crucial for a proper event and disturbance analysis [88]. In terms of new SAS needs, the utilization of the aforementioned PMUs is challenging the state-of-the-art as previously indicated in Chapter 3, requesting tens of ns as synchronization accuracy [41], [42], [39].

In terms of safety, a relevant standard to be considered in this mission-critical task is IEC 61508 [5] in order to guarantee the safety integrity of the system and IEEE 1686-2013 *Standard for In-*

telligent Electronic Devices Cyber Security Capabilities [89] for security. This combined with timing, interoperability, reliability, availability, safety and security requirements serves to highlight the need for an optimum use case as proof of concept for SAS. For this reason, this Chapter describes the design and the implementation of a complete use case that includes a redundant network with multiple sensor/actuators nodes and switches for a control substation scenario.

This use case, titled as *Synchronized low-latency deterministic networks*, has been has been developed jointly with research members of the University of Granada timing group[1]and Schneider Electric[2] in the framework of the EMC^2 European project.

6.2 SYNCHRONIZED LOW-LATENCY DETERMINISTIC NETWORKS USE CASE

The main element of a Electrical Distribution Network (EDN) is SAS [90], which controls and monitors the electrical infrastructure. Typically, it is composed of the three levels illustrated in the following figure:

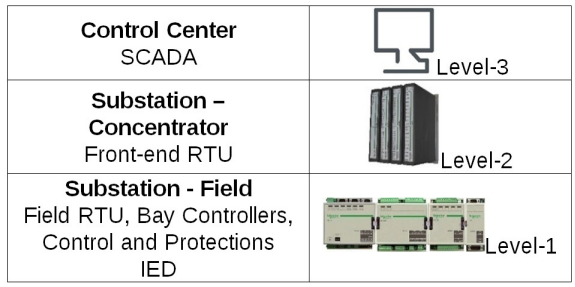

Figure 6.1
Substation Automation Systems in 3 Levels: Control Center, Substation Concentrator level and Field level.

The highest level regards to the control center that includes the SCADA system and receives all the acquired data at electrical substations. It provides control capabilities over the managed infrastructure. The SCADA system also provides functions for graphical displays, alarming, trending and historical storage of data.

The second level, the concentrator, includes the required elements for the communication between the field site and the control center. Front-end RTUs have communication capabilities,

1 http://www.timingkeepers.com/
2 https://www.schneider-electric.es/

being able to be connected to routers, fiber optics, etc., to interact with the control center.

The lower level, also named field level, focuses on acquisition and control activities to gather data and send it to the control center. Common devices present in the field site are RTUs, Bay Controllers and IEDs. These devices are critical assets equipped with input and output signals that provide control, monitoring and data gathering functions to the substations. The selected use case developed during the EMC2 project, was based on the mentioned classic hierarchical architecture for SAS, in order to provide a realistic validation scenario. The use case includes RTU devices (Saitel DR and Saitel DP RTU families of Schneider Electric[3]) set in a master-slave configuration, with a RTU acting as front-end at the higher level, and other RTUs acting as acquisition RTUs at field level.

The higher level send commands to be executed at field level, sending data from the field to the front-end. The acquisition RTUs are composed by control devices (SM_CPU866e, HU_A) and I/O acquisition devices (SM_DO32T, SM_DI32, AB_DI, AB_DO). On the other hand, the front-end RTU is composed by control devices (SM_CPU866e, HU_A). The front-end includes a program which continuously sends commands to the acquisition RTUs. From field, acquisition signals are sent upstream by industrial control protocols such as IEC 60870-5-104 [91], DNP3 [92] and Modbus [93] that have been configured for the communication. Additionally, PTPv2 and IRIG-B have been selected as synchronization sources for the control system.

Regarding time distribution, the use case includes WR devices which provide deterministic sub-nanosecond synchronization features to this SAS scenario. A WRS configured as GM of the timing network. Six WRSs with HSR capabilities forming a ring network core. These WRSs can provide time using both WR and PTPv2. Apart from WRSs, a cascade of several WR-LENs [4] propagates time from the core of the network to the boundary. WR-LENs are able to propagate time using WR, PTP and IRIG-B. Furthermore, they are used as interface for the two acquisition modules and RTUs, changing from WR to PTP and IRIG-B to synchronize the acquisition modules with the GM of the network. Finally, a specific safety and security evaluation tool, based on the standards IEC 61508 and IEEE 1686-2013 has been integrated to evaluate the safety and security level of the system.

3 https://www.schneider-electric.com.sg/en/work/products/mv-distribution-and-energy-automation.jsp

4 http://sevensols.com/index.php/products/wr-len

The proposed use case validates the following requirements for next industrial network applications and internet of things captured during EMC2 project execution. The requirements document can be found at [94], and it is summarized as follows:

- Deterministic high-accuracy time and frequency transfer and time distribution compatibility: The utilization of novel Ethernet protocols providing the best synchronization accuracy leads to an enhancement of control systems in SAS regarding event triggering, capture and data acquisition [41], [42], [39]. In addition, industrial timing compatibility support is also desirable to build a unique timing network supplying different protocols at the same time [11].
- Reliable time and data transfer: The implementation of redundancy protocols such as HSR/PRP increases fault tolerance and robustness because it avoids single point of failure network scenarios and, at the same time, the duplication of frames ensures the reception of critical packets on the destination node. Moreover, this increases the availability of any of the services in a WR network including time, which is considered critical in Smart Grid.
- Timing Scalability: SAS represents, by nature, multiple interconnected nodes in cascade and parallel configurations. These nodes must also be synchronized to the same master clock reference of the timing network and thus, the synchronization accuracy must be evaluated accordingly to determine the maximum number possible of hops.
- Safety and security: The evaluation of the Safety Integrity Level and security of the system must be carried out using specific tools based on the standards [5], [89].

Next Section presents the implementation details of the protocols, methods and mechanisms that have been developed and integrated into the use case to meet each of the requirements stated above.

6.3 SYNCHRONIZED LOW-LATENCY DETERMINISTIC NETWORKS IMPLEMENTATION

The following subsections describe the implementation of the required features for the SAS use case, focusing on time distribution, network reliability, scalability, safety and security. Fig. 6.2 represents the design of this use case.

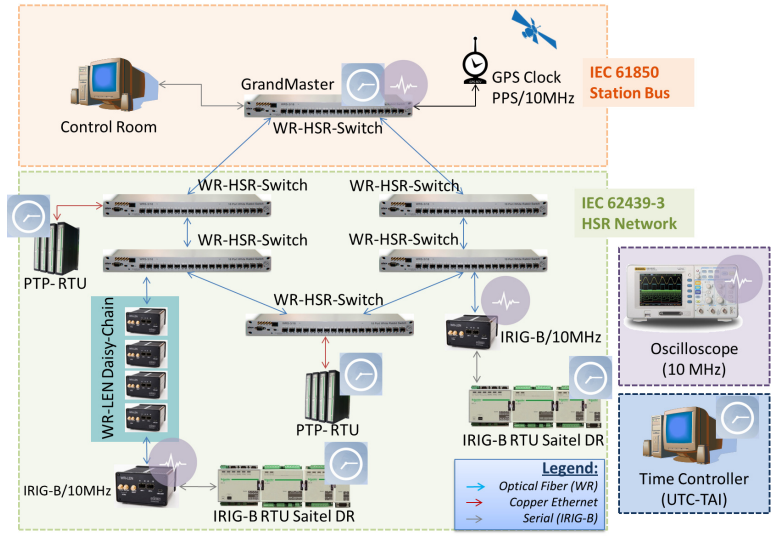

Figure 6.2
Use case implementation design. It is formed by a GM node (WRS) connected to an Ethernet HSR ring, which duplicates both timing and data frames. The acquisition system is attached as leaf nodes to the ring, assuring the reception of the time reference from the master, and also the communication between nodes (control data frames), up to daisy-chain configurations of 12 nodes. Timing technologies involved are WR, PTPv2 and IRIG-B.

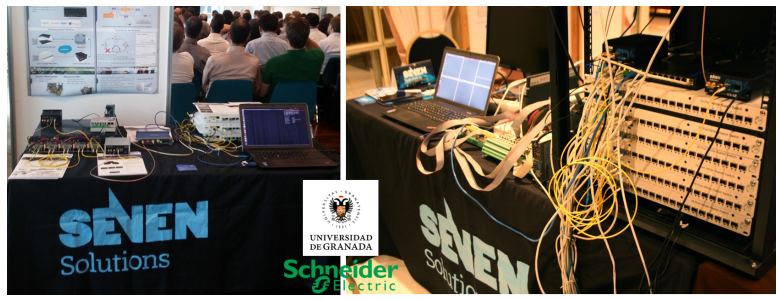

Figure 6.3
Left picture depicts the first prototype of the use case *Synchronized low-latency deterministic networks* presented in the EMC^2 General Meeting, in September 2015, hold in TTTech offices, Vienna (Austria), in which timing scalability and control elements are on the left side, and redundancy features are on the right side. Right picture shows the final version of the demonstrator, presented in the third and last EMC^2 Review Meeting in June 2017, hold in Hotel Carmen, Granada (Spain). This final demonstrator integrates all features described within this thesis: security, communication reliability, timing compatibility, sub-nanosecond synchronization accuracy together redundancy features for both timing and data dissemination.

The results of this work were firstly presented as a prototype (Fig. 6.3, left) at the 2nd EMC^2 General Assembly Meeting in September 2015, hold in TTTech offices, Vienna (Austria). The final version of this demonstrator (Fig. 6.3, right) was successfully exhibited to the EU Project Officer Reviewers at the 3rd and last EMC^2 review meeting in June 2017, organized by Seven Solu-

tions S.L. and hold in Hotel Carmen, Granada (Spain). Fig. 6.3 shows pictures of these two events.

6.3.1 *Time distribution and industrial compatibility*

As previously mentioned, the main timing technology used at the core of this network is WR. The WR solution of this use case is composed of WRSs and WR-LENs, provided by Seven Solutions S.L. By this, we ensure that the most stringent timing requirements for SAS are met, as described in Chapter 3.

In addition to this, WR devices allow using low-performance protocols such as PTP and IRIG-B at the boundary of the network (last mile). Since WR devices are also able to disseminate PTP and IRIG-B to leaf nodes such as RTUs and IEDs, it guarantees that all network devices are synchronized to the same primary time reference at the core of the network.

Consequently, all events of all RTUs are marked with a time-stamp with the same notion of time. The synchronization is implemented not only at Control level, but also at acquisition level, where digital inputs boards (SM_DI32 and AB_DI) register events with a time-stamp based on the synchronization reference.

6.3.2 *Reliable time and data transfer*

In order to meet the dependability requirements suggested by IEC 61850 for Smart Grid, the WR-compatible implementation of the HSR protocol [7] described in Chapter 5, Section 5.3 has been integrated into this use case. By this, we guarantee fault tolerance, high-availability and single point of failure avoidance for both timing and data frames in ring topologies.

6.3.3 *Timing scalability*

The scalability of a communication infrastructure becomes crucial in heterogeneous networks like the one referred in this case study [95]. In order to increase the scalability of the timing network, the WR implementation for this use case adopts the P2P approach described in Chapter 5, Section 5.2, to measure the delay between nodes, and the nodes forming the ring behave as TCs and HYs. This improves the stability and thus, the scalability of the entire timing system as [51] and [77] results present. Moreover, industrial timing interoperability is also increased since this type of applications tends to use TCs and HYs together with the P2P approach [96].

6.3.4 *Safety and security*

In order to address safety and security aspects of the system, Schneider Electric together with University of Cantabria[5] developed a new Eclipse-based tool capable of estimating the SIL and the security assessment of the whole system.

This tool is composed of three elements integrated in the same development framework. The first two elements address the IEC 61508 safety analysis, respect to HW design and HW/SW integration; and the third one addresses the security analysis (IEEE 1686), as shown in the figure below:

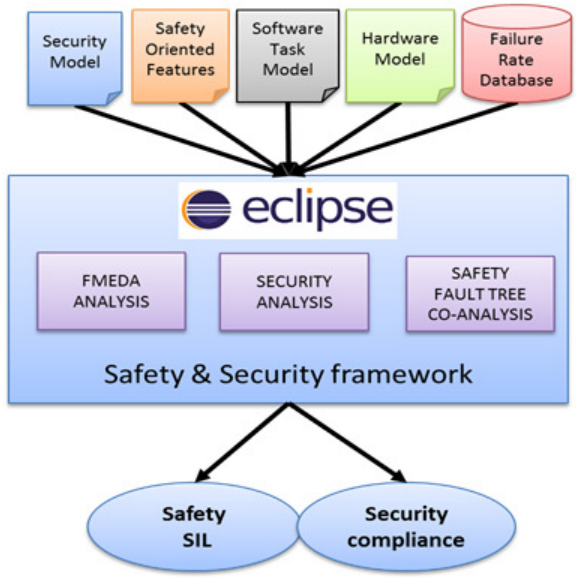

Figure 6.4
Safety and Security component integrated in the framework.

The first component is the FMEDA analysis. It is applied to the hardware design and enables the automation of the hardware model capture, using Electronic Design Interchange Format (EDIF) and Bill of Materials (BOM) files. FMEDA and test reports with the necessary information for the safety certification is automatically generated. These reports include the failure modes of each component and additional parameters, that are required to calculate the SIL of the hardware sub-system.

The second component is the *safety fault tree co-analysis* that covers the hardware/software integration and provides a SIL estimation of complete hardware-software system. The description of the hardware components is completed with models of the

5 http://web.unican.es/

software components (tasks and/or functions) and hardware/-software mapping (allocation of task/functions to hardware resources). The tool generates a FTA analysis to estimate the tolerable hazard rate. With this information, it is possible to estimate the safety level. This generates a scheme of the tree and a table with the information that helps to detect the weakest branches of the tree, in order to improve the safety parameters.

The third component is the *security analysis* and is based on IEEE 1686, which establishes clauses that the system has to comply with. These clauses are grouped in a *table of compliance*, facilitating the information capture. With this information, the framework generates a complete report with statistics and graphs, showing the level of compliance of the system.

6.4 RESULTS

This section presents the evaluation of the performance and the benefits of developments related to the use case for each of the requirements stated at the end of Section 6.2. Fig. 6.5 shows the setup built to cover all the features of the use case.

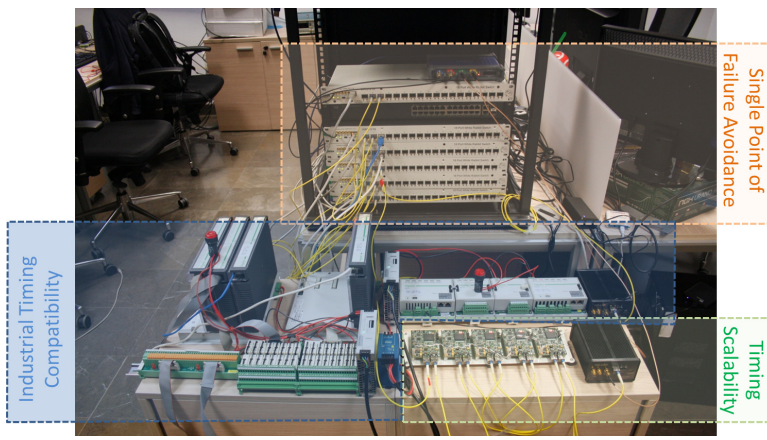

Figure 6.5
Use case setup. This setup represents the minimum grid scenario to proof industrial compatibility, timing and data redundancy and scalability. The left side of the image (blue shape) represents the acquisition modules of the system at the boundary of the network. They are composed of SM_CPU 866e and HU_A RTUs and the synchronization protocols used are PTPv2 and IRIG-B respectively. The core of the timing network is composed of six WRSs forming a HSR ring (orange shape), where the one on the top is the WR GM, guaranteeing an accuracy below 1 ns. The bottom-right side of the image represents the scalability of the timing system using WR-LENs (green shape). WR-LENs are also used to translate WR to IRIG-B.

6.4.1 *Time distribution and industrial compatibility*

Time and frequency accuracy has been measured taking into account the different protocols that coexist within the network and how they are interconnected. Apart from WR and for the sake of interoperability, PTPv2 and IRIG-B have been evaluated to synchronize the RTUs of the control system (leaf nodes). SM_CPU 866e implements PTPv2 and the HU_A model uses IRIG-B. Table 6.1 summarizes the timing system interconnection and the different accuracies of the use case depending on the technology used. Measurements have been taken using a signal counter[6], comparing the 1-PPS signal outputs between WRS, WR-LEN, SM_CPU 866e and RTU HU_A. This device adds 20 ps of uncertainty to the final results.

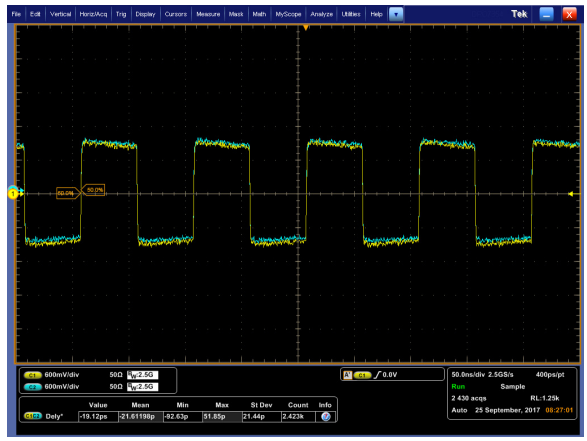

Figure 6.6
10MHz output from HSR White Rabbit devices synchronized with an accuracy below 1 ns.

Two calibrated WRS switches using WR synchronize with an average accuracy of 21 ps (Fig. 6.6); a WRLEN connected to a WRS with WR presents an accuracy of 111 ps. Using standard PTPv2 with hardware time-stamps, a SM_CPU866e connected to a WRS shows an average of 112.326 ns. Finally, the synchronization accuracy between a WR-LEN and a RTU HU_A using IRIG-B is 10 ms on average. It is important to remark that the WR time transfer technology is practically immune to the data traffic, thus leading to a more deterministic network deployment. This is not the case of PTPv2, which might suffer performance degradation due to big data traffic loads or packet delay variation.

6 http://www.tek.com/datasheet/fca3000-and-fca3100-series

Table 6.1
Synchronization accuracy per protocol used in use case. WR for the core of the network, PTPv2 for the acquisition system for medium distances, and IRIG-B for the acquisition and control system at network boundary

Master Device	WR (offset avg)	PTPv2 (offset avg)	IRIG-B (offset avg)	Slave Device
WRS	0.021 ns (σ 0.021 ns)	N/A	N/A	WRS
WRS	0.111 ns (σ 0.052 ns)	N/A	N/A	WR-LEN
WRS	N/A	N/A	112.326 ns (σ 14.6 ns)	SN_CPU866e
WR-LEN	N/A	N/A	10 ms	RTU HU_A

Finally, thanks to the adaptation of WR devices to support standard industrial timing protocols, such a PTP and IRIG-B, we are able to reduce the synchronization of all the devices to a unique GM instead of using one GM per protocol, thus increasing the level of determinism on the network.

6.4.2 *Reliable time and data transfer*

The implementation of the HSR protocol provides the acquisition and control system with redundancy features for timing and data frames, thus increasing network services availability due to the duplication of frames, and thereby allowing the single point of failure avoidance. Regarding redundant time distribution, the developed HSR protocol guarantees having two time references at the same time. In case the primary time reference is lost, HSR nodes are able to switch over the backup reference in µs [80] maintaining the sub-nanosecond accuracy. This is thanks to the adaptation of the switchover mechanism to ring topologies and to the switchover alert system using raw control symbols, which is able to disseminate the alert in one µs per hop.

These results have been previously discussed in previous Chapter 5, Sections 5.5.2 and 5.5.3.

6.4.3 *Timing scalability*

The scalability of the timing system and its impact on the synchronized 1-PPS signal in terms of jitter and skew have been addressed in Chapter 5, Section 5.5.1.

6.4.4 *Safety and security*

The new safety and security tool has been validated and tested with the relevant devices of the use case (SM_CPU866e, SM_DO 32T, WR-Switch and WR-LEN), obtaining the SIL estimation and

the IEEE 1686 level of compliance. Chapter 3, Section 3.2 includes the definitions of the measurements that have been used perform this SIL estimation.

The following table shows the SIL results for the mentioned devices.

Table 6.2
Result estimation of the Safety Integrity Level (SIL)

Device		WR-LEN	WRS	SM_CPU866e	SM_DO32T
Safe Failure Fraction (SFF)	%	83.64	81.33	72.91	76.44
Internal Hardware Fault Tolerance (HFT)		0.00	0.00	0.00	0.00
Dangerous Detected (DD) Failure Rate (λDD)	FITS	0.00	0.00	0.00	0.00
Dangerous Undetected (DU) Failure Rate (λDU)	FITS	61.22	113.02	283.24	264.70
Safe (S) Failure Rate (λS)	FITS	312.96	492.18	762.41	858.76
Fail High (H) Failure Rate (λH)	FITS	0.00	0.00	0.00	0.00
Fail Low (L) Failure Rate (λL)	FITS	0.00	0.00	0.00	0.00
No Effect (NE) Failure Rate (λNE)	FITS				
No Part (-) Failure Rate (λ-)	FITS	0.00	0.00	0.00	0.00
System Type		B	B	B	B
Safety Integrity Level (SIL)		SIL 1	SIL 1	SIL 1	SIL 1

This first iteration, offers a low SIL level as a result for all the devices, but this is the starting point for improving the SIL level. First, it is necessary to identify the critical components that are basically those with the higher dangerous undetected failure rate (λDU), then there are two possible actions to increase safety level:

- Change the component by other with an equivalent one with better λDU.
- Include diagnostic in order to increase the diagnostic coverage.

It is worth to mention that, in spite of obtaining SIL 1 for safety, SIL 2 was almost achievable thanks to the integration of the HSR protocol.

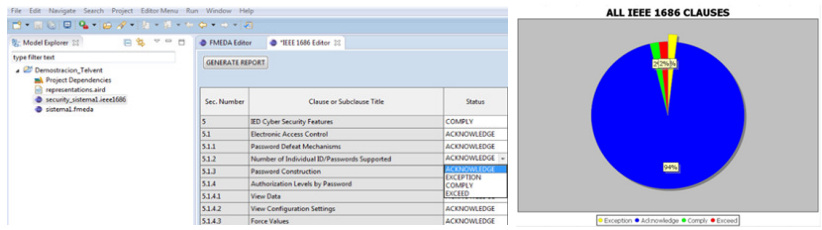

Figure 6.7
Security compliance tool developed by Schneider Electric and the security results for SM_CPU866e (blue: acknowledge, yellow: exception, green: comply, red: exceed).

Regarding the security analysis, the security information of the device SM_CPU866e has been evaluated. The IEEE 1686 clauses are introduced in the *Table of Compliance*. After completing the clauses, a report is generated. This report shows information about the level of compliance of the device respect to the standard as present in Fig. 6.7, with a 94% of acknowledge status (blue slice) for the evaluated security features.

6.5 CONCLUSION

The implemented use case can be taken as proof of concept for the new Smart Grid applications where dependability, scalability, heterogeneous synchronization protocols and low-latency are key or mandatory features for control systems. This use case represents a new horizon in Smart Grid in terms of time synchronization and dependability.

Previous work described in Chapters 5 and 4 has been integrated into a real SAS scenario that includes the first WR-compatible HSR protocol following the suggestions of [7] that guarantees a mechanism to recover from a failure in approx. 3 µs. The utilization of WR devices for the distributed control system guarantees sub-nanosecond synchronization accuracy up to 18 nodes using TCs and HYs (ring and cascade topologies). Future PMU's synchronization accuracy needs, not covered by PTPv2, are now achievable with WR [39, 41]. In addition to this, these devices can provide different time protocols such as PTPv2 and IRIG-B, thus increasing industrial devices compatibility. In terms of data, the developed WR-HSR hardware is able to reduce frame forwarding latency to half of the standard one (from 2.2 µs WCE to 1 µs).

In this contribution, we have also addressed the integration of a safety and security framework that is based on FMEDA, FTA and security analysis. The results obtained from the evaluation of the RTUs validate the technology and demonstrate that the HW/SW development and integration process can be reduced by 15% and the re-estimation of the analysis can be improved more than a factor of 2.

Finally, this Chapter encloses all the concepts, designs, developments and prototypes carried out during the course of this thesis. The integration of them into a real industrial SAS use case demonstrates the fulfillment of the strict timing, safety and security requirements for industrial applications.

Part III

CONCLUSIONS

7

CONCLUSIONS

There is no real ending.
It's just the place where you stop the story.

— Frank Herbert

INDEX

This doctoral thesis presents our contributions to the areas of DCS, placing particular emphasis on Smart Grid and WAMS applications. This Chapter is structured as follows. Firstly, we present a general discussion of the problem that motivated the developments described in previous Chapters and the proposed solutions. Secondly, future work plans and suggestions to improve our developments. Finally, we enumerate the publications derived from our work and highlight the main contributions achieved.

7.1 CONCLUSIONS

DCS realize three main activities or functionalities: data acquisition, data evaluation and control. These activities are accomplished at different devices among the network. Because of the critical nature of these systems, reliability and security must be considered as one of the most important features that must covered. This doctoral thesis stresses on the reliability requirements of critical distributed applications, starting from the leaf-nodes (acquisition modules), focusing on the development of reliable multi-core architectures for mixed-criticality applications, passing through the reliability of data transferred over the network, and ending with the dissemination of high-accurate dependable timing technologies to provide the network with the same notion of time with the best accuracy possible.

Chapter 4 works on the needs of inter-core communications of leaf-nodes distributed along a DCS to guarantee the reliability of the applications running on these devices. This entails the development of a mixed-critical architecture capable of executing critical and non-critical applications. This design has been carried out in the framework of the EU FP7 RECOMP project, resulting in a dual-core system with sensing capabilities able to stop and monitor safely an assembly line in a industrial environment by the development and integration of reliability mechanisms for hardware, software, firmware, process isolation and also for the inter-core communication itself.

In terms of hardware dependability, we have evaluated an Open Source reliable platform (ACP) for avionics that meets the hardware requirements described in DO-254. This platform runs the mixed-critical application. The benefits of this platform lie in the isolation mechanism developed for the ARM and the FPGA and also in the duplication of all peripherals to allow resources division and isolation.

This system implements an AMP dual-core architecture that runs the critical application that develops the emergency stop button logic. This architecture is composed of two MicroBlaze soft-processors inside the ACP's Virtex-6 FPGA. These cores execute two instances of a reliable (with certification possibilities) RTOS, sharing a redundant communication channel architecture 1oo2 that enables performing a cross-diagnose comparison stage as IEC 61508 suggests. A C2C communication library developed by Wittenstein is used to exchange data between these two MicroBlaze processors in order to guarantee the reliability of the communication up to SIL 3.

At the same time, this system integrates a sensing application with run-time upgrade capabilities developed by Åbo Akademi University. This application is able to retrieve critical data from the FPGA with no disturbance to the critical application because of the isolation mechanisms developed. By this, we are able to upgrade the sensing application depending on the operator needs with no impact for the critical application.

Besides the reliability of critical leaf-nodes, Chapter 5 presents the studies conducted to increase inter-process communications reliability in DCS. This includes techniques and methods to increase dependability features in data and timing transmission by the implementation of redundancy mechanisms. Apart from that, improving the stability and scalability of the timing system towards industrial compatibility have also been an important achievement.

In this connection, the implementation of P2P TCs and HYs has made possible to increase the scalability and performance of WR in daisy-chain configurations. The utilization of HYs guarantees a sub-nanosecond synchronization accuracy up to the 17^{th} node, TCs are capable of propagating this accuracy up to the 18^{th} node, whilst the standard E2E BC WR implementation is only capable up to the 11^{th} node. In addition to this, the utilization of P2P TCs allows the reduction of 1-PPS signal jitter in approximately 100 ps, thus improving the stability of the transferred signal. These results improve significantly the WR scalability, being now suitable for wide area networks which are composed of many devices forming cascade configurations like Smart Grid and more precisely, SAS [1, 11], where time accuracy becomes prominent, as [41, 84] highlight for the future of synchronized PMUs, requesting accuracies below ten nanoseconds.

For the first time, a redundancy protocol has been implemented for the WR technology. This ensures the sub-nanosecond synchronization accuracy for the devices attached to the ring with an average $offset_{ms}$ of 200 ps capable of having two time references simultaneously: a primary time source and a backup reference, used in case of failure. The approach developed guarantees no disruption on the synchronization in spite of following two time references at the same time. These timing results improve significantly previous works of other standard HSR PTP implementations from the state-of-the-art, such as [73–76], that present a maximum accuracy between 30 ns and 100 ns.

The WR-HSR implementation meets IEC 61580 and IEC 62439 suggestions related to timing fault tolerance, providing a switchover mechanism capable of changing from the primary to the backup time reference with no impact for the synchronization

of the entire network. This mechanism guarantees the WR sub-nanosecond accuracy with an observed maximum phase shift of 170 ps in ringed networks composed of six WRS. This is possible thanks to the utilization of an alert system transmitted in hundred nanoseconds over the physical layer using 8b/10b symbols.

In terms of data, HSR guarantees the reception of critical data between different processes running on different devices distributed among the grid. This fault-tolerance implementation offers a maximum bandwidth of 680Mbps and increases the availability of the Smart Grid services, such as GOOSE, GSSE and SMV.

One of the benefits of this implementation regards in the reduction of the data transfer latency in the ring, thus increasing the response time to hazardous situations. The developed FFU FPGA module enhances the transmission of data frames through the ring, reducing each hop's latency from the standard WRS 3 µs [82] (WRS standard latency) to 1.4 µs.

Finally, Chapter 6 encompasses all the concepts and functionalities described in Chapters 4 and 5 in order to deploy a real Smart Grid scenario, focusing on a specific SAS use case developed in the framework of the EMC^2 EU project. This scenario has been used to evaluate and validate the requirements previously stated in the state-of-the-art for distributed safety-critical applications: dependable data and timing dissemination, network device's integrity, synchronization accuracy and industrial timing compatibility using different timing protocols.

In terms of dependability, the development of the HSR guarantees the sub-nanosecond accuracy for the core of the network even after the failure of one of the nodes with no impact for the synchronization. It is scalable up to 17-18 nodes using TCs and HYs, thus ensuring reaching leaf-nodes at network boundary that normally use additional timing protocols. For this reason, WR devices integrate also PTPv2 and IRIG-B. Thanks to this, we are capable of providing the highest synchronization accuracy (below 1 ns) for the core of the network, intermediate nodes present an average of 100 ns (PTPv2), and leaf multi-core nodes for data acquisition and event control are synchronized also using long-term supported protocols like IRIG-B (10 ms).

The availably of the services provided within the network are now ensured thanks to the implementation of the HSR into WR devices. HSR guarantees control messages reception at the end nodes, such as RTUs, even after a node/path failure in the ring.

Safety and security issues have also been addressed based on HW FMEDA, HW/SW FTA and security analysis. The results obtained from the evaluation of the RTUs validate the technology

and demonstrate that the HW/SW development and integration process can be reduced by 15% and the re-estimation of the analysis can be improved more than a factor of 2. These analyses have also resulted in guaranteeing SIL 1, close to achieving SIL 2.

Because of all this, during the course of this doctoral thesis we have developed a cutting-edge heterogeneous high-accuracy synchronized DCS that integrates reliable inter-core and inter-process communication mechanisms using multi-core leaf-nodes for Smart Grid, where dependability features have been evaluated at the core and the boundary of the network with respect to time and data distribution. This has been made possible by the development of the first WR-compatible HSR implementation capable of guaranteeing a redundant sub-nanosecond synchronization accuracy, reducing frame forwarding latency to 50%, incrementing scalability and industrial compatibility nearly twice thanks to the utilization of transparent and hybrid clocks, and finally, improving industrial timing compatibility thanks to the integration of the P2P approach together the integration of other timing protocols such as PTP and IRIG-B.

7.2 MAIN CONTRIBUTIONS

In this section, we state the main contributions achieved in this Ph.D work:

- We have studied the evolution of SC systems from the single-core to the multi-core perspective, focusing on the isolation of hardware and software components, the utilization of reliable operating systems and the integration of secure inter-core communication libraries.
- We have developed a SC application based on FreeRTOS in a multi-core platform using a 1002 architecture to monitor and evaluate the status of an emergency stop system for the industrial domain. This development integrates a reliable inter-core communication library that guarantees SIL 3.
- We have described a case study that brings together these features and also a NSC sensing application. It defines an industrial safe motor controller in a mixed-critical environment according to IEC 61508, DO-254 and DO-178C. The resulting implementation includes the following features: a SC application to evaluate an emergency stop, a NSC application to control the status of the system with run-time upgrade capabilities, the integration of a reliable inter-core

communication library and the implementation of isolation mechanisms on FPGA hardware to isolate completely critical and non-critical elements. The SC application is SIL3-compliant according to IEC 61508.

- We have studied and evaluated the synchronization needs for DCS and Smart Grid networks, paying special attention to wide area distributed systems like SAS.

- We have developed transparent and hybrid clocks using a two-step P2P approach to improve the accuracy, stability and compatibility of the WR technology. Whilst the standard WR implementation using E2E BCs is only capable of maintaining sub-nanosecond results up to the 11^{th} node, HYs preserves this accuracy up to the 17^{th} node and TCs up to the 18^{th} one. In addition, this P2P implementation enables WR compatibility with other PTPv2-like industrial timing networks, since they normally use P2P instead of E2E. Finally, the utilization of WR P2P TCs improves the stability of the distributed 1-PPS signal in approximately 100 ps (jitter).

- We have studied different solutions to improve time and data distribution reliability on Smart Grid networks. After evaluating these solutions, the HSR protocol was proposed as the candidate to implement WR redundancy features.

- We developed a redundant version of the WR-PTP protocol for the WR-HSR, capable of handling two time references at the same time on any device of the ring. This has been achieved by the adaptation of the WR L1 syntonization process, which makes possible to syntonize two WR slave clocks to each other, enabling a simultaneous master-slave behavior.

- We have adapted a switchover mechanism from parallel to ring topologies to enable switching from the primary to the backup time reference in hundred µs. Thanks to this approach, losing the primary time reference is seamless for the entire synchronization network, observing a maximum phase shift of 170 ps in ringed networks composed of six WRS.

- We have developed a FPGA module, the FSU, able to disseminate the switchover alert to all the nodes of the ring in µs using L1 8b/10b comma symbols, thus guaranteeing that the holdover time of the WRS is not exceeded (100 ms).

- We have developed a HSR IP core for the WRS, called LRE, that enables redundancy features for data transmission. This guarantees the reception of critical frames on destination nodes, thus increasing fault tolerance and avoiding single point of failure issues. HSR-WRS guarantees a maximum bandwidth of 680Mbps.
- We have developed a FFU module that improves the original WRS forwarding latency up to a fifty percent (from 3 to 1 µs).
- We have studied the new needs for future Smart Grid applications with special focus on timing, dependability and compatibility. Thanks to this study we have been able to design an accurate and reliable timing network that satisfies the most demanding timing requirements for next generation of Smart Grid.
- According to this study, we have developed an industrial DCS to simulate a SAS environment integrating all the features and mechanisms stated above. Devices are able to distribute the timing reference using WR and other industrial protocols such as PTPv2 and IRIG-B. This guarantees an ultra-accurate synchronization performance for the core of the network with an accuracy below 1 ns, approximately 100 ns for intermediate industrial nodes using PTPv2, and IRIG-B for the network boundary with accuracies of 10 ms. In addition, this system includes HSR capabilities for both timing and data frames, thus increasing the availability and reliability of the services provided. Finally, safety analysis has resulted in guaranteeing SIL 1 for the developed system, close to achieving SIL 2.

7.3 FUTURE WORK

As future work, we plan to continue working on dependable distributed control systems.

In the first place, we want to improve the WRS reliability by the development of mechanisms similar to the ones described in Chapter 4 in order to guarantee a safety communication between the ARM and the FPGA. In addition, we plan to increment the number of LM32 processors in the FPGA to develop a redundant version of the softPLL capable of performing a cross-comparison evaluation stage between the data computed by each processor.

On the one hand, we intend to improve and adapt completely the WR-HSR implementation to IEC 62439-3 Clause 5 standard.

This involves also modifying the current version of WR TC and HY clock implementations to make them fully compatible with other industrial devices. Furthermore, a fully FPGA hardware version of TCs together with an enhanced PI controller have been proposed for the near future to improve the scalability of WR devices.

Additionally, we plan to extend the switchover concept from having the possibility to switch between two time references over fiber links, to switch between different technologies such as WR, GNSS, standard PTPv2 and standard NTP.

Finally, a natural evolution of the approaches described on this doctoral thesis probably aims to the adaptation of the WR-HSR protocol to Time Sensitive Networking (TSN). For this reason we would like to continue exploring reliable time and data transfer for mission critical applications over TSN frameworks.

7.4 PUBLICATIONS

The published or submitted works related to this doctoral thesis are the following:

7.4.1 *International journals with scientific impact*

- José Luis Gutiérrez-Rivas, Simon Holmbacka, Miguel Méndez, Wictor Lund, Sebastién Lafond, Johan Lilius and Javier Díaz. *Safe Motor Controller in a Mixed-Critical Environment with Runtime Updating Capabilities.* In Journal of Universal Computer Science. DOI: 10.3217/jucs-021-12-0177 Vol. 21, No. 2, pp. 177-205. (Q_4)

- José Luis Gutiérrez-Rivas, José López-Jiménez, Eduardo Ros and Javier Díaz. *White Rabbit HSR: a seamless sub-nanosecond redundant timing system with low-latency data capabilities for Smart Grid.* In IEEE Transactions on Industrial Informatics. DOI: 10.1109/TII.2017.2779240, Electronic ISSN: 1941-0050. (Q_1, Rank 1^{st})

- Francisco Ramos, José Luis Gutiérrez-Rivas, José López-Jiménez, Benito Caracuel and Javier Díaz. *Accurate Timing Networks for Dependable Smart Grid Applications.* In IEEE Transactions on Industrial Informatics. 14(5): 2076-2084 (2018), DOI: 10.1109/TII.2017.2787145. (Q_1, Rank 1^{st})

- Miguel Jiménez-López, Felipe Torres-González, José Luis Gutiérrez-Rivas, Manuel Rodriguez-Álvarez and Javier Díaz.

A fully programmable White-Rabbit node for the SKA Telescope PPS distribution system IEEE Transactions on Instrumentation and Measurement. DOI: 10.1109/TIM.2018.2851658. (Q_1)

- José Luis Gutiérrez-Rivas, Felipe Torres-González, Eduardo Ros and Javier Díaz. *White Rabbit Transparent and Hybrid Clocks: improving synchronization performance towards industrial domains.* In IEEE Transactions on Industrial Electronics (Q_1, Rank 1^{st}) (submitted on 26-Jun-2018).

7.4.2 *National conferences*

- José Luis Gutiérrez-Rivas, Jesper Berthing, David Fernández and Javier Díaz. *Safety-Critical Platform Model Based on Certification Standards.* In III Jornadas de Computación Empotradas, JCE2012.

- David Fernández, José Luis Gutiérrez-Rivas and Javier Díaz. *Monitorización en Tiempo de Ejecución para Sistemas Críticos Empotrados.* In XII Jornadas sobre Computación Reconfigurable y Aplicaciones. JCRA2012.

- Javier Díaz, José Luis Gutiérrez-Rivas, David Fernández and Miguel Mendez-Macías. *Investigación en Sistemas Empotrados en la Universidad de Granada: Desarrollo de aplicaciones críticas en sistemas multicore.* in II Jornadas de Computaciń Empotradas, JCE2011.

7.4.3 *International conferences*

- José López-Jiménez, José Luis Gutiérrez-Rivas, Miguel Jimé nez-López and Javier Díaz. *White-Rabbit-enabled Data Acquisition System.* In 31st European Frequency and Time Forum, 10th - July 13th 2017, Besançon, France.

- José Luis Gutiérrez-Rivas, César Prados and Javier Díaz. *Sub-Nanosecond Synchronization Accuracy for Time-Sensitive Applications on Industrial Networks.* In 30th European Frequency and Time Forum, 4th - 7th April 2016, York, United Kingdom.

- Miguel Jiménez-López, José Luis Gutiérrez-Rivas and Javier Díaz. *White-Rabbit Network Interface Card for Synchronized Sensor Networks.* In IEEE Sensors 2014 in Valencia, Spain. November 2014.

- Miguel Mendez-Macías, José Luis Gutiérrez-Rivas, David Fernández and Javier Díaz. *Open Platform for Mixed-Criticality Applications.* In Workshop on Industry-Driven Approaches for Cost-effective Certification of Safety-Critical, Mixed-Criticality Systems, WICERT 2013 (DATE '13)

8 CONCLUSIONES

No hay un final real.
Es simplemente el lugar en el que decides parar la historia.

— Frank Herbert

INDEX

Esta tesis doctoral presenta nuestra contribución al área de los DCS, poniendo especial énfasis en su aplicabilidad a Smart Grid y WAMS. En primer lugar realizaremos una discusión general sobre la problemática que motivó los desarrollos previamente descritos en Capítulos anteriores y las soluciones propuestas. En segundo lugar, hablaremos del plan de trabajo futuro y de algunas sugerencias para mejorar nuestros propios desarrollos. Finalmente, enumeraremos las publicaciones realizadas a raíz del trabajo aquí expuesto y destacaremos las principales contribuciones.

8.1 CONCLUSIONES

Los DCS realizan principalmente tres tipos de actividades o funcionalidades: adquisición de datos, evaluación de datos y control. Estas actividades se llevan a cabo en diferentes dispositivos distribuidos a lo largo de la red. Debido a la naturaleza crítica de estos sistemas, la fiabilidad y la seguridad son características que deben ser consideradas y cubiertas en todo momento. Esta tesis doctoral pone especial hincapié en los requisitos de fiabilidad de las aplicaciones distribuidas críticas y de todos los elementos que las forman, empezando por los nodos finales (módulos de adquisición), en los que tendremos que concentrarnos en el desarrollo de arquitecturas multinúcleo fiables para aplicaciones de criticidad mixta, pasando por la implementación de mecanismos de control de datos críticos a través de la red, y finalizando con la distribución de tecnologías de sincronización altamente precisas y fiables, capaces de proporcionar a lo largo de toda la red una noción temporal común con la mejor precisión posible.

El Capítulo 4 recoge las necesidades de las comunicaciones entre los diferentes núcleos de los nodos hoja distribuidos a lo largo de un DCS para garantizar la fiabilidad de las aplicaciones que están en ejecución en estos dispositivos. Esto conlleva el desarrollo de un arquitectura de criticidad mixta capaz de ejecutar aplicaciones críticas y no críticas de forma simultánea. Este diseño ha sido llevado a cabo en el marco del proyecto europeo FP7 RECOMP, dando lugar a la creación de un sistema de doble núcleo que incluye un sistema de detección capaz de parar y monitorizar de forma segura una cadena de montaje de un entorno industrial. Todo esto gracias al desarrollo e integración de mecanismos que proporcionan fiabilidad en términos de hardware, software, firmware, aislamiento de procesos y además, en la propia comunicación entre procesos.

En cuanto a la fiabilidad del hardware, hemos evaluado una plataforma segura de código abierto (ACP) para aviónica que cumple con los requisitos hardware descritos en DO-254. Esta plataforma ha sido utilizada en el desarrollo e integración de la aplicación de criticidad mixta anteriormente descrita. Los beneficios de dicha plataforma residen en el aislamiento entre el procesador ARM y la FPGA, además de duplicar todos los recursos y periféricos para facilitar una correcta utilización de los mismos.

Este sistema implementa un arquitectura de doble núcleo AMP para la aplicación crítica, la cual integra la funcionalidad del botón de emergencia. Esta arquitectura está compuesta de dos procesadores MicroBlaze sobre la FPGA Virtex-6 de la ACP. Estos procesadores ejecutan la aplicación crítica en dos instancias

diferentes de un RTOS confiable (y certificable), utilizando una arquitectura de comunicación segura y redundante 1oo2 para el desarrollo de la etapa de diagnóstico cruzado, tal y como sugiere el estándar IEC 61508. Para asegurar la comunicación entre ambos núcleos, se utilizó una librería desarrollada por Wittenstein que garantiza SIL 3 para el intercambio de datos entre los procesos.

Al mismo tiempo, este sistema integra una aplicación de detección de fallos desarrollada por Åbo Akademi University, capaz de reconfigurarse en tiempo real. Dicha aplicación es capaz de extraer la información sensible de la FPGA sin provocar ninguna alteración en la aplicación crítica, gracias a los mecanismos de aislamiento desarrollados en la FPGA. Además, esto facilita la actualización de las características de la aplicación relacionadas con la detección de errores en función de las necesidades de los operarios de la cadena de montaje.

Al margen de las características de fiabilidad necesarias en los nodos finales, el Capítulo 5 presenta los estudios realizados para incrementar la fiabilidad del proceso de comunicación entre procesos distribuidos en un DCS. Para ello, se desarrollaron diferentes mecanismos para proporcionar redundancia en la transmisión de datos y en la sincronización de los dispositivos de la red. Además, se mejoró la estabilidad y la escalabilidad del sistema de distribución de tiempo WR con el fin de adecuar la tecnología a entornos industriales.

A este respecto, la implementación de TCs y HYs junto al mecanismo P2P para medir retardos, ha hecho posible incrementar la escalabilidad, estabilidad y rendimiento de WR en configuraciones en cadena. La utilización de HYs garantiza una precisión por debajo del nanosegundo en cadenas formadas por hasta 17 nodos. Los TCs son capaces de propagar esta precisión hasta el nodo número 18, mientras que la versión estándar de WR (E2E BCs) solo es capaz de mantener dicho resultado hasta el nodo 11. Asimismo, la utilización de TCs P2P permiten reducir en torno a los 100 ps la inestabilidad de la señal de 1-PPS, mejorando así la estabilidad de las señales propagadas por WR. Estos resultados mejoran significativamente la escalabilidad de WR, permitiendo ahora utilizar esta tecnología en entornos industriales dotados de muchos nodos distribuidos a lo largo de la red, como puede ser Smart Grid y SAS [1, 11], donde la sincronización se está convirtiendo en una característica altamente demandada, como subraya [41, 84] con requisitos cercanos a los 10 nanosegundos para PMUs.

Por primera vez, se ha desarrollado un protocolo de redundancia para la tecnología WR. Esto garantiza precisiones por

debajo del nanosegundo para dispositivos conectados en anillo, con una media de 200 ps de $offset_{ms}$ junto a la posibilidad de tener dos referencias de tiempo simultáneas: una fuente principal, y otra de respaldo, utilizada en caso de fallo. La solución desarrollada garantiza poder tener dos referencias de tiempo simultáneas. Estos resultados mejoran significativamente otros desarrollos HSR/PRP anteriormente realizados, los cuales presentan una precisión máxima entre 30 y 100 ns [73–76].

La implementación WR-HSR cumple las recomendaciones de los estándares IEC 61580 y IEC 62439 en cuanto a tolerancia a fallos, proporcionando un mecanismo de conmutación capaz de cambiar de una fuente primaria a una de respaldo sin ocasionar ningún impacto en la sincronización de la red. Además, dicho mecanismo garantiza la sincronización sub-nanosegundo que proporciona WR con un salto de fase máximo de 170 ps para anillos formados por seis WRS. Esto es posible gracias a la utilización de un mecanismo de alerta que se transmite en cientos de nanosegundos a través de la capa física, utilizando caracteres de coma 8b/10b.

En cuanto a la transmisión de datos WR-HSR, podemos garantizar la recepción de datos críticos entre diferentes procesos ejecutados en diferentes dispositivos distribuidos por toda la red. Este mecanismo tolerante a fallos alcanza un ancho de banda de 680Mbps y garantiza la disponibilidad de los servicios en Smart Grid, como pueden ser GOOSE, GSSE y SMV.

Uno de los beneficios de esta implementación se encuentra en la reducción de la latencia a la hora de transmitir datos, por lo que se incrementa la respuesta ante situaciones de riesgo para el funcionamiento de la red. La solución desarrollada mejora la transmisión de datos a lo largo del anillo, reduciendo la latencia de transmisión de cada nodo de los 3 µs de la versión estándar del WRS [82] a unos 1.4 µs, gracias a la utilización del módulo FFU en la FPGA.

Finalmente, el Capítulo 6 engloba todos los conceptos y funcionalidades descritas en los Capítulos 4 y 5 para crear un escenario real de Smart Grid, poniendo especial interés en un caso de uso específico de SAS, el cual fue desarrollado en el marco del proyecto EMC^2. Este escenario ha sido utilizado para evaluar y validar lo anteriormente expuesto en el estado del arte en relación con los DCS para gestionar aplicaciones críticas. Dichos conceptos son: distribución de datos de firma segura, integridad de los dispositivos de la red, distribución de una referencia de tiempo fiable y altamente precisa, además de la compatibilidad entre diferentes protocolos industriales de sincronización.

En cuanto a la fiabilidad, el desarrollo del protocolo HSR para la distribución de datos y tiempo garantiza una precisión por debajo del nanosegundo para el núcleo de la red incluso después de sufrir fallos en un nodo de la red sin generar ningún impacto en la sincronización. Por otro lado, la solución es escalable hasta 17-18 cascadas de nodos utilizando TCs y HYs, garantizando el alcance de la referencia de tiempo a los nodos finales de la red, los cuales generalmente utilizan otros protocolos de tiempo. Por esta razón, los dispositivos WR integran también PTPv2 estándar e IRIG-B, incrementando así la compatibilidad con los nodos cuando sea necesario. Gracias a esto, somos capaces de proporcionar una precisión por debajo del nanosegundo para el núcleo de la red, alrededor de 100 ns (PTPv2) para los nodos intermedios, y 10 ms para aquellos dispositivos multinúcleo de adquisición de datos que aún utilicen IRIG-B.

Gracias al desarrollo e implementación del protocolo HSR en los dispositivos WR somos capaces de garantizar la disponibilidad de los servicios que se proporcionan a través de la red. HSR garantiza que los mensajes de control lleguen a los nodos finales, como pueden ser RTUs, incluso tras haber ocurrido una pérdida de un enlace de la red.

Cuestiones relacionadas con la seguridad y fiabilidad también han sido abordadas mediante el desarrollo de análisis basados en FMEDA (hardware) y FTA (hardware/software). Los resultados obtenidos para las RTUs validan la tecnología y demuestran que el desarrollo e integración HW/SW puede reducirse en torno al 15%, y que la re-estimación del análisis puede mejorar en un fator de 2. Estos análisis han dado como resultado un nivel de integridad SIL 1 para el sistema final, aunque se han obtenido valores muy cercanos a SIL 2.

Debido a todo esto, durante el transcurso de esta tesis doctoral hemos conseguido desarrollar un DCS sincronizado heterogéneo altamente preciso puntero, el cual integra mecanismos de comunicación seguros inter-núcleo e inter-proceso utilizando nodos finales multinúcleo para Smart Grid, donde se han evaluado las características de fiabilidad tanto en el núcleo de la red como en los límites de la misma con respecto a la distribución de tiempo y datos. Esto ha sido posible gracias al desarrollo de la primera versión del protocolo HSR compatible con WR, capaz de garantizar una sincronización con una precisión por debajo del nanosegundo redundante, reduciendo la latencia de transmisión en torno al 50%, e incrementando la escalabilidad de WR al doble de nodos, gracias a la utilización de relojes transparentes e híbridos. Finalmente, se ha mejorado también la compat-

ibilidad industrial en cuanto a la distribución de tiempo, gracias a la integración de otros protocolos como PTP e IRIG-B.

8.2 CONTRIBUCIONES PRINCIPALES

A continuación expondremos las principales contribuciones realizadas en el transcurso de esta tesis doctoral:

- Hemos estudiado la evolución de los SC partiendo de una perspectiva mononúcleo a multinúcleo, con especial énfasis en el aislamiento tanto del hardware como del software, la utilización de sistemas operativos confiables y la integración de librerías de comunicación seguras entre procesos.

- Hemos desarrollado una aplicación SC basada en FreeRTOS en una plataforma multinúcleo utilizando una arquitectura 1oo2 para monitorizar y evaluar el estado de un sistema de parada de emergencia para aplicaciones industriales. Este desarrollo incluye una librería de comunicación segura capaz de garantizar la integridad de la comunicación hasta SIL 3.

- Hemos descrito un caso de estudio capaz de unificar estas características junto a una aplicación de detección NSC. Éste define la implementación de un controlador seguro para un motor industrial en un entorno de criticidad mixta acorde a los estándares IEC 61508, DO-254 y DO-178C. La implementación resultante incluye: una aplicación SC para evaluar la parada de emergencia del motor, una aplicación NSC para el control del estado del sistema capaz de reconfigurarse en tiempo de ejecución, la integración de una librería de comunicación segura entre procesos locales, y la implementación de mecanismos de aislamiento en la FPGA para aislar la parte crítica de la no crítica. La aplicación SC es compatible con SIL3, de acuerdo a lo definido en IEC 61508.

- Hemos estudiado y evaluado las necesidades relacionadas con la sincronización en redes Smart Grid y DCS, prestando especial atención sistemas distribuidos de área extensa como puede ser SAS.

- Hemos desarrollado para WR dos nuevos tipos de relojes: transparentes e híbridos. Estos relojes implementan un nuevo mecanismo del cómputo del retardo entre dos nodos

llamados P2P (en dos pasos), el cual mejora la precisión y estabilidad de la tecnología WR. Mientras WR estándar es capaz únicamente de sincronizar 11 dispositivos en cascada manteniendo una precisión por debajo del nanosegundo, los HYs son capaces de preservar dicha precisión hasta el nodo 17, y los TCs hasta el nodo 18. Además, se ha demostrado que la utilización de P2P como mecanismo de cómputo del retardo mejora la compatibilidad con otras implementaciones industriales que utilizan PTPv2. Finalmente, la utilización de WR P2P TCs mejora la estabilidad de la señal 1-PPS en torno a 100 ps.

- Hemos estudiado las diferentes soluciones para mejorar la fiabilidad en la distribución de tiempo y datos en redes Smart Grid. Tras evaluar estas soluciones se decidió desarrollar el protocolo HSR para dotar a WR de características de redundancia.

- Hemos desarrollado una implementación redundante del protocolo WR-PTP, la cual es utilizada por la solución WR-HSR para hacer posible la gestión de dos referencias de tiempo simultáneas en topologías de anillo. Esto ha sido posible mediante la adaptación del proceso de sintonización sobre la capa física de WR, la cual hace posible sintonizar dos esclavos WR entre ellos, siendo a la vez maestro y esclavo del nodo adyacente.

- Hemos adaptado un mecanismo de conmutación originalmente diseñado para redes paralelas a redes anilladas, para hacer posible cambiar de una fuente primaria de tiempo a una de respaldo en cientos de μs. Gracias a esta solución, el hecho de perder la referencia primaria no causa ningún impacto en la sincronización de toda la red, observando un salto de fase máximo de 170 ps para anillos formados por hasta seis WRS.

- Hemos desarrollado un módulo en la FPGA llamado FSU, capaz de transmitir una alerta para provocar la conmutación de la referencia de tiempo del anillo en μs, usando para ello caracteres de coma 8b/10b sobre la capa física. Gracias a esto, podemos garantizar que la conmutación se realizada antes perder la sincronización por completo (100 ms para el WRS).

- Hemos desarrollado un módulo VHDL para el WRS denominado LRE. Este modulo habilita las capacidades de redundancia para la transmisión de datos HSR. Con esto,

garantizamos la recepción de tramas críticas en los nodos destino, incrementando así la tolerancia a fallos. Un WRS con funcionalidades HSR ofrece un ancho de banda máximo de 680Mbps.

- Hemos desarrollado el módulo FPGA FFU, el cual mejora la latencia de reenvío del WRS original en torno al 50% (de 3 a 1 µs).
- Hemos estudiado las nuevas necesidades de las futuras aplicaciones de Smart Grid, centrándonos en la distribución de tiempo, fiabilidad y compatibilidad. Gracias a este estudio, hemos diseñado una red precisa y confiable capaz de cubrir las necesidades más estrictas para la futura generación de Smart Grid (< 10 ns).
- Según este estudio, hemos desarrollado un DCS industrial para simular un entorno SAS integrando todas las características y mecanismos expuestos anteriormente. Los dispositivos son ahora capaces de distribuir una referencia de sincronización utilizando WR además de otros protocolos industriales como PTPv2 e IRIG-B. Con esto podemos garantizar un rendimiento en cuanto a la sincronización ultrapreciso para el núcleo de la red, con valores por debajo del nanosegundo, 100 ns para los nodos intermedios utilizando PTPv2, y en torno a los 10 ms para los límites de la red utilizando IRIG-B. Además, este sistema integra capacidades HSR tanto para el envío de referencias de tiempo como para datos, incrementando así la disponibilidad y fiabilidad de los servicios proporcionados en la red. Finalmente, los análisis de seguridad garantizan SIL 1 para el sistema en su totalidad, presentando valores cercanos a SIL 2.

8.3 TRABAJO FUTURO

Como trabajo futuro queremos seguir trabajando en la línea de sistemas de control distribuidos confiables.

En primer lugar, queremos mejorar las propiedades de fiabilidad del WRS, desarrollando mecanismos similares a los descritos en el Capítulo 4 para garantizar una comunicación segura entre ARM y FPGA, además de la posibilidad de incrementar la redundancia del softPLL utilizando dos procesadores LM32 junto con una etapa cruzada de comprobación entre ambos.

Por otro lado, tenemos planeado mejorar y adaptar completamente la implementación WR-HSR a lo descrito por el estándar

IEC 62439-3. Esto supondría modificar la versión actual de los WR TCs y HYs, para hacerlos completamente compatibles con otras versiones industriales de los mismos. Además, pretendemos diseñar una nueva versión de los TCs en VHDL dentro de la FPGA, junto a una versión mejorada del controlador PID, para así mejorar aún más la escalabilidad de los dispositivos WR.

Además, queremos extender el concepto de conmutación de múltiples referencias de tiempo a más protocolos y tecnologías más allá de WR, como pueden ser GNSS, PTPv2, NTP, etc.

Finalmente, una evolución natural de todos los planteamientos y desarrollos descritos en esta tesis posiblemente apunten a la adaptación del protocolo WR-HSR hacia redes sensibles al tiempo (Time Sensitive Networking, TSN). Por este motivo, nos gustaría seguir explorando la transmisión de los datos y el tiempo en aplicaciones de misión crítica en entornos TSN.

8.4 PUBLICACIONES

A continuación se listan los trabajos publicados o pendientes de publicación relacionados con esta tesis doctoral:

8.4.1 *Revistas internacionales con impacto científico*

- José Luis Gutiérrez-Rivas, Simon Holmbacka, Miguel Méndez, Wictor Lund, Sebastién Lafond, Johan Lilius and Javier Díaz. *Safe Motor Controller in a Mixed-Critical Environment with Runtime Updating Capabilities.* In Journal of Universal Computer Science. DOI: 10.3217/jucs-021-12-0177 Vol. 21, No. 2, pp. 177-205. (Q_4)

- José Luis Gutiérrez-Rivas, José López-Jiménez, Eduardo Ros and Javier Díaz. *White Rabbit HSR: a seamless sub-nanosecond redundant timing system with low-latency data capabilities for Smart Grid.* In IEEE Transactions on Industrial Informatics. DOI: 10.1109/TII.2017.2779240, Electronic ISSN: 1941-0050. (Q_1, Rank 1^{st})

- Francisco Ramos, José Luis Gutiérrez-Rivas, José López-Jiménez, Benito Caracuel and Javier Díaz. *Accurate Timing Networks for Dependable Smart Grid Applications.* In IEEE Transactions on Industrial Informatics. 14(5): 2076-2084 (2018), DOI: 10.1109/TII.2017.2787145. (Q_1, Rank 1^{st})

- Miguel Jiménez-López, Felipe Torres-González, José Luis Gutiérrez-Rivas, Manuel Rodriguez-Álvarez and Javier Díaz.

A fully programmable White-Rabbit node for the SKA Telescope PPS distribution system IEEE Transactions on Instrumentation and Measurement. DOI: 10.1109/TIM.2018.2851658. (Q_1)

- José Luis Gutiérrez-Rivas, Felipe Torres-González, Eduardo Ros and Javier Díaz. *White Rabbit Transparent and Hybrid Clocks: improving synchronization performance towards industrial domains.* In IEEE Transactions on Industrial Electronics (Q_1, Rank 1^{st}) (submitted on 26-Jun-2018).

8.4.2 *Conferencias nacionales*

- José Luis Gutiérrez-Rivas, Jesper Berthing, David Fernández and Javier Díaz. *Safety-Critical Platform Model Based on Certification Standards.* En III Jornadas de Computación Empotradas, JCE2012.
- David Fernández, José Luis Gutiérrez-Rivas and Javier Díaz. *Monitorización en Tiempo de Ejecución para Sistemas Críticos Empotrados.* En XII Jornadas sobre Computación Reconfigurable y Aplicaciones. JCRA2012.
- Javier Díaz, José Luis Gutiérrez-Rivas, David Fernández and Miguel Mendez-Macías. *Investigación en Sistemas Empotrados en la Universidad de Granada: Desarrollo de aplicaciones críticas en sistemas multicore.* En II Jornadas de Computacińn Empotradas, JCE2011.

8.4.3 *Conferencias internacionales*

- José López-Jiménez, José Luis Gutiérrez-Rivas, Miguel Jiménez-López and Javier Díaz. *White-Rabbit-enabled Data Acquisition System.* En 31st European Frequency and Time Forum, 10 - 13t julio 2017, Besançon, Francia.
- José Luis Gutiérrez-Rivas, César Prados and Javier Díaz. *Sub-Nanosecond Synchronization Accuracy for Time-Sensitive Applications on Industrial Networks.* En 30th European Frequency and Time Forum, 4 - 7 Abril 2016, York, Reino Unido.
- Miguel Jiménez-López, José Luis Gutiérrez-Rivas and Javier Díaz. *White-Rabbit Network Interface Card for Synchronized Sensor Networks.* En IEEE Sensors 2014 en Valencia, España. Noviembre 2014.
- Miguel Mendez-Macías, José Luis Gutiérrez-Rivas, David Fernández and Javier Díaz. *Open Platform for Mixed-Criti*

cality Applications. En Workshop on Industry-Driven Approaches for Cost-effective Certification of Safety-Critical, Mixed-Criticality Systems, WICERT 2013 (DATE '13).

BIBLIOGRAPHY

[1] Hassan Farhangi. The path of the smart grid. *IEEE power and energy magazine*, 8(1), 2010.

[2] Vehbi C Gungor, Dilan Sahin, Taskin Kocak, Salih Ergut, Concettina Buccella, Carlo Cecati, and Gerhard P Hancke. Smart grid technologies: Communication technologies and standards. *IEEE transactions on Industrial informatics*, 7(4):529–539, 2011.

[3] V Cagri Gungor, Dilan Sahin, Taskin Kocak, Salih Ergut, Concettina Buccella, Carlo Cecati, and Gerhard P Hancke. A survey on smart grid potential applications and communication requirements. *IEEE Transactions on Industrial Informatics*, 9(1):28–42, 2013.

[4] Piotr Gaj, Jürgen Jasperneite, and Max Felser. Computer communication within industrial distributed environment-a survey. *IEEE Transactions on Industrial Informatics*, 9(1):182–189, 2013.

[5] International Electrotechnical Commission et al. Functional safety of electrical/electronic/programmable electronic safety related systems. *IEC 61508*, 2000.

[6] IEC TC57. Iec 61850: Communication networks and systems for power utility automation. *International Electrotechnical Commission Std*, 53:54, 2010.

[7] International Electrotechnical Commission et al. *IEC 62439-3: Industrial Communication Networks: High Availability Automation Networks. Parallel Redundancy Protocol (PRP) and High Availability Seamless Redundancy (HSR). Protocole de Redondance Parallèle (PRP) Et Redondance Transparente de Haute Disponibilité (HSR).* IEC, 2012.

[8] Ketan Maheshwari, Marcus Lim, Lydia Wang, Ken Birman, and Robbert van Renesse. Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud. In *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*, pages 1–6. IEEE, 2013.

[9] Carl H Hauser, David E Bakken, and Anjan Bose. A failure to communicate: next generation communication require-

ments, technologies, and architecture for the electric power grid. *IEEE Power and Energy Magazine*, 3(2):47–55, 2005.

[10] Jaime De La Ree, Virgilio Centeno, James S Thorp, and Arun G Phadke. Synchronized phasor measurement applications in power systems. *IEEE Transactions on Smart Grid*, 1(1):20–27, 2010.

[11] MI Ridwan, NS Miswan, MSM Shokri, MN Noran, RM Lajim, and HN Awang. Interoperability in smart grid using iec 61850 standard: A power utility prospect. In *Innovative Smart Grid Technologies-Asia (ISGT Asia), 2014 IEEE*, pages 261–266. IEEE, 2014.

[12] John Eidson and Kang Lee. Ieee 1588 standard for a precision clock synchronization protocol for networked measurement and control systems. In *Sensors for Industry Conference, 2002. 2nd ISA/IEEE*, pages 98–105. IEEE, 2002.

[13] Ecsel joint undertaking. https://www.ecsel.eu/. Accessed: 2018-05-22.

[14] Artemis industry association. https://artemis-ia.eu/. Accessed: 2018-05-22.

[15] Horizon 2020 programme. https://ec.europa.eu/programmes/horizon2020/. Accessed: 2018-05-22.

[16] Jeff Parkhurst, John Darringer, and Bill Grundmann. From single core to multi-core: preparing for a new exponential. In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pages 67–72. ACM, 2006.

[17] Sergey Zhuravlev, Sergey Blagodurov, and Alexandra Fedorova. Addressing shared resource contention in multicore processors via scheduling. In *ACM Sigplan Notices*, volume 45, pages 129–142. ACM, 2010.

[18] Dionisio De Niz, Karthik Lakshmanan, and Ragunathan Rajkumar. On the scheduling of mixed-criticality real-time task sets. In *Real-Time Systems Symposium, 2009, RTSS 2009. 30th IEEE*, pages 291–300. IEEE, 2009.

[19] Mohamed B Abdelhalim and SE-D Habib. An integrated high-level hardware/software partitioning methodology. *Design Automation for Embedded Systems*, 15(1):19–50, 2011.

[20] Rodolfo Pellizzoni, Patrick Meredith, Min-Young Nam, Mu Sun, Marco Caccamo, and Lui Sha. Handling mixed-criticality in soc-based real-time embedded systems. In *Proceedings of the seventh ACM international conference on Embedded software*, pages 235–244. ACM, 2009.

[21] Sergio Cuenca-Asensi, Antonio Martinez-Alvarez, Felipe Restrepo-Calle, Francisco R Palomo, Hipolito Guzman-Miranda, and Miguel A Aguirre. Soft core based embedded systems in critical aerospace applications. *Journal of Systems Architecture*, 57(10):886–895, 2011.

[22] Sanjoy Baruah, Haohan Li, and Leen Stougie. Towards the design of certifiable mixed-criticality systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE*, pages 13–22. IEEE, 2010.

[23] Tim P Kelly. Managing complex safety cases. In *Proceedings of 11th Safety Critical System Symposium (SSS'03)*, 2003.

[24] YC Yeh. Safety critical avionics for the 777 primary flight controls system. In *Digital Avionics Systems, 2001. DASC. 20th Conference*, volume 1, pages 1C2–1. IEEE, 2001.

[25] David Powell, Jean Arlat, Yves Deswarte, and Karama Kanoun. Tolerance of design faults. *Dependable and Historic Computing*, 6875:428–452, 2011.

[26] John C Knight and Nancy G Leveson. A reply to the criticisms of the knight & leveson experiment. *ACM SIGSOFT Software Engineering Notes*, 15(1):24–35, 1990.

[27] J-C Laprie, Jean Arlat, Christian Beounes, and Karama Kanoun. Definition and analysis of hardware-and software-fault-tolerant architectures. *Computer*, 23(7):39–51, 1990.

[28] BRITISH Standard and BS IEC. Functional safety-safety instrumented systems for the process industry sector. *ANSI/ISA S*, 84, 2003.

[29] Leslie A Johnson et al. Do-178b, software considerations in airborne systems and equipment certification. *Crosstalk, October*, 199, 1998.

[30] RTCA DO. Do-254, design assurance guidance for airborne electronic hardware. *Research and development for alternate antenna designs is needed to solve the antenna crash tolerance and*

installation problems. e. A New Zealand designed, privately developed Secondary Antenna Switching Device may offer a solution but will need development to certification and deployment, 2000.

[31] ISO26262 ISO. 26262: Road vehicles-functional safety. *International Standard ISO/FDIS*, 26262, 2011.

[32] TÜV SÜ. Training course interpreting functional safety acc. iec 61508 / iso 26262. 2011.

[33] Keith Stouffer, Joe Falco, and Karen Scarfone. Guide to industrial control systems (ics) security. *NIST special publication*, 800(82):16–16, 2011.

[34] Ali Keyhani and Muhammad Marwali. *Smart power grids 2011*. Springer, 2012.

[35] Ganesh Kumar Venayagamoorthy and Scott F Belcher. Smart grid and electric transportation. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, pages 1–2. IEEE, 2009.

[36] AJ Dinusha Rathnayaka, Vidyasagar M Potdar, and Samitha J Kuruppu. An innovative approach to manage prosumers in smart grid. In *Sustainable Technologies (WCST), 2011 World Congress on*, pages 141–146. IEEE, 2011.

[37] Vehbi Cagri Gungor, Dilan Sahin, Taskin Koçak, Salih Ergüt, Concettina Buccella, Carlo Cecati, and Gerhard P. Hancke. A survey on smart grid potential applications and communication requirements. *IEEE Trans. Industrial Informatics*, 9(1):28–42, 2013.

[38] Hossein Zeynal, Mostafa Eidiani, and Dariush Yazdanpanah. Intelligent substation automation systems for robust operation of smart grids. In *Innovative Smart Grid Technologies-Asia (ISGT Asia), 2014 IEEE*, pages 786–790. IEEE, 2014.

[39] U Annakage, A Rajapakse, B Bhargava, N Chaudhuri, A Mehrizi-sani, C Hauser, D Wadduwage, S Ribeiro Campos Andrade, V Pathirana, K Katsaros, et al. Application of phasor measurement units for monitoring power system dynamic performance. Technical report, Cigré, 2017.

[40] YaShian Li-Baboud, Cuong T Nguyen, Marc A Weiss, Dhananjay Anand, Allen R Goldstein, Jason Allnutt, Bob Noseworthy, and Ravi Subramaniam. Timing challenges in the smart grid. Technical report, 2017.

[41] Reza Razzaghi, A Derviskadic, and Mario Paolone. A white rabbit synchronized pmu. In *Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2017 IEEE PES*, pages 1–6. IEEE, 2017.

[42] Guglielmo Frigo, Daniele Colangelo, Asja Derviškadić, Marco Pignati, Claudio Narduzzi, and Mario Paolone. Definition of accurate reference synchrophasors for static and dynamic characterization of pmus. *IEEE Transactions on Instrumentation and Measurement*, 66(9):2233–2246, 2017.

[43] IRIG Standard. 200-04–irig serial time code formats–september 2004, timing committee, telecommunications and timing group, range commanders council. *US Army White Sands Missile Range, NM*.

[44] Kenneth E Martin. Synchrophasor standards development-ieee c37. 118 & iec 61850. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–8. IEEE, 2011.

[45] Xingxing Li, Maorong Ge, Xiaolei Dai, Xiaodong Ren, Mathias Fritsche, Jens Wickert, and Harald Schuh. Accuracy and reliability of multi-gnss real-time precise positioning: Gps, glonass, beidou, and galileo. *Journal of Geodesy*, 89(6):607–635, 2015.

[46] David Hambling. Gps chaos: How a $30 box can jam your life. *New Scientist*, 2803:44–47, 2011.

[47] Kris Maine, Carrie Devieux, and Pete Swan. Overview of iridium satellite network. In *WESCON/'95. Conference record.'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies'*, page 483. IEEE, 1995.

[48] Maciej Lipinski. Ieee standard for a precision clock synchronization protocol for networked measurement and control systems high accuracy changelog. https://www.ohwr.org/projects/wr-std/wiki, 2017.

[49] Pedro Moreira, Javier Serrano, Tomasz Wlostowski, Patrick Loschmidt, and Georg Gaderer. White rabbit: Sub-nanosecond timing distribution over ethernet. In *Precision Clock Synchronization for Measurement, Control and Communication, 2009. ISPCS 2009. International Symposium on*, pages 1–5. IEEE, 2009.

[50] Tomasz Wlostowski. Precise time and frequency transfer in a white rabbit network. *master of science thesis*, 2011.

[51] José Luis Gutiérrez-Rivas, César Prados, and Javier Díaz. Sub-nanosecond synchronization accuracy for time-sensitive applications on industrial networks. In *European Frequency and Time Forum (EFTF), 2016*, pages 1–4. IEEE, 2016.

[52] IBM DO. 178b compliance: turn an overhead expense into a competitive advantage. *White paper, IBM Rational*, 2010.

[53] RECOMP Workpackage 4 Deliverable. D.4.2b.1 recommendations for use of multi-core in certifiable applications for avionics. *Artemis FP7 JU RECOMP Project*.

[54] M Mendez, JLG Rivas, DF Garca-Valdecasas, and J Diaz. Open platform for mixed-criticality applications. In *Proceedings of the Conference on Design, Automation and Test in Europe, WICERT, DATE*, 2013.

[55] Hannu T Toivonen and Jari Tamminen. Minimax robust lq control of a thermomechanical pulping plant. *Automatica*, 26(2):347–351, 1990.

[56] Shariful Islam, Neeraj Suri, András Balogh, György Csertán, and András Pataricza. An optimization based design for integrated dependable real-time embedded systems. *Design Automation for Embedded Systems*, 13(4):245, 2009.

[57] Armin Wasicek, Christian El-Salloum, and Hermann Kopetz. A system-on-a-chip platform for mixed-criticality applications. In *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2010 13th IEEE International Symposium on*, pages 210–216. IEEE, 2010.

[58] Iain Bate and Tim Kelly. Architectural considerations in the certification of modular systems. *Reliability Engineering & System Safety*, 81(3):303–324, 2003.

[59] Malcolm S Mollison, Jeremy P Erickson, James H Anderson, Sanjoy K Baruah, and John A Scoredos. Mixed-criticality real-time scheduling for multicore systems. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 1864–1871. IEEE, 2010.

[60] Jaynarayan H Lala and Richard E Harper. Architectural principles for safety-critical real-time applications. *Proceedings of the IEEE*, 82(1):25–40, 1994.

[61] J. Berthing and Danfoss A.S. *D5.6 Drive Controller*. Artemis JU RECOMP Project.

[62] Wittenstein High Integrity Systems. *34-190-MAN-01 C2C Communication Library User Manual*. Artemis JU RECOMP Project.

[63] Michael Wahler, Stefan Richter, and Manuel Oriol. Dynamic software updates for real-time systems. In *Proceedings of the 2nd International Workshop on Hot Topics in Software Upgrades*, page 2. ACM, 2009.

[64] OpenRTOS. Wittenstein high integrity system. http://www.highintegritysystems.com/openrtos/, 2006.

[65] Seven Solutions Inc. *D5.2 Reference design dual core Avionic Compute Platform for avionics applications*. Artemis JU RECOMP Project.

[66] R Barry. Freertos reference manual: Api functions and configuration options, real time engineers ltd. *URL: http://www.freertos. org*, 2009.

[67] SafeRTOS. Wittenstein high integrity system. http://www.highintegritysystems.com/safertos/, 2006.

[68] Wictor Lund. *A unified run-time updating and task migration mechanism*. PhD thesis, Master Thesis, 2012.

[69] Waldemar Wojdak. Rapid spanning tree protocol: A new solution from an old technology. *Reprinted from CompactPCI Systems*, 2003.

[70] Hubert Kirrmann, Claudio Honegger, Ioannis Sotiropoulos, and Diana Ilie. Industrial ethernet seamless redundancy and sub-microsecond clock synchronization with iec 62439-3 and iec 61588.

[71] Hubert D. Kirrmann, Karl Weber, Oliver Kleineberg, and Hans Weibel. HSR: zero recovery time and low-cost redundancy for industrial ethernet (high availability seamless redundancy, IEC 62439-3). In *Proceedings of 12th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2009, September 22-25, 2008, Palma de Mallorca, Spain*, pages 1–4, 2009.

[72] Hubert Kirrmann, Karl Weber, Oliver Kleineberg, and Hans Weibel. Seamless and low-cost redundancy for substation automation systems (high availability seamless redundancy,

hsr). In *Power and Energy Society General Meeting, 2011 IEEE*, pages 1–7. IEEE, 2011.

[73] SoCe HPS. Hsr-prp switch ip core. https://goo.gl/JwgiJn.

[74] CISCO. Cisco parallel redundancy protocol (prp) for ie 4000, ie 4010, and ie 5000 switches. https://goo.gl/QBW8U8.

[75] Flexibilities TTTech. Flexibilities tttech hsr/prp switches and redboxes. https://www.flexibilis.com/products/hsrprp-switches/.

[76] Texas Instruments. Texas instruments high-availability seamless redundancy (hsr) ethernet for substation automation. http://www.ti.com/lit/ug/tidub08/tidub08.pdf, 2006.

[77] José Luis Gutiérrez-Rivas, Felipe Torres-González, and Javier Díaz. White rabbit transparent and hybrid clocks: improving synchronization performance towards industrial domains. *IEEE Transactions on Industrial Informatics*.

[78] José Luis Gutiérrez-Rivas, César Prados, and Javier Díaz. Sub-nanosecond synchronization accuracy for time-sensitive applications on industrial networks. In *European Frequency and Time Forum (EFTF), 2016*, pages 1–4. IEEE, 2016.

[79] Pedro Moreira, Javier Serrano, Tomasz Wlostowski, Patrick Loschmidt, and Georg Gaderer. White rabbit: Sub-nanosecond timing distribution over ethernet. In *Precision Clock Synchronization for Measurement, Control and Communication, 2009. ISPCS 2009. International Symposium on*, pages 1–5. IEEE, 2009.

[80] Maciej Marek Lipinski. *Vol. 40-Methods to Increase Reliability and Ensure Determinism in a White Rabbit Network*. PhD thesis, Warsaw U. of Tech.

[81] ESnet Energy Sciences Network. iperf. https://iperf.fr/, 2006.

[82] M Lipinski. White rabbit-ethernet-based solution for sub-ns synchronization and deterministic, reliable data delivery. In *IEEE Plenary Meeting Geneva*, volume 15, 2013.

[83] Saad Allawi Nsaif and Jong Myung Rhee. RMT: A novel algorithm for reducing multicast traffic in HSR protocol networks. *Journal of Communications and Networks*, 18(1):123–131, 2016.

[84] Mario Paolone. Time synchronisation needs in phasor measurement units for the real-time monitoring of power grids. https://indico.cern.ch/event/399810/contributions/950558/attachments/800656/1097273/2015-06-22_Paolone.pdf, 2015.

[85] NASPI North American SyncroPhasor Initiative. Time synchronization in the electric power system. In *NASPI Time Synchronization Task Force, March 2017*, 2017.

[86] Waheed Ur Rahman, Muhammad Ali, Chaudhry A Mehmood, and Asadullah Khan. Design and implementation for wide area power system monitoring and protection using phasor measuring units. *WSEAS TRANSACTIONS on POWER SYSTEMS*, 8:57–64, 2013.

[87] James Northcote-Green and Robert G Wilson. *Control and automation of electrical power distribution systems*, volume 28. CRC Press, 2006.

[88] Paolo Romano and Mario Paolone. Enhanced interpolated-dft for synchrophasor estimation in fpgas: Theory, implementation, and validation of a pmu prototype. *IEEE Transactions on Instrumentation and Measurement*, 63(12):2824–2836, 2014.

[89] IEEE Power and Energy Society. *1686-2013 - IEEE Standard for Intelligent Electronic Devices Cyber Security Capabilities*. IEEE.

[90] Mark Adamiak and Ashish Kulshrestha. Design and implementation of a uca-based substation control system. In *Proceedings of the 2001 Georgia Tech Protective Relay Conference, Atlanta, Georgia*, 2002.

[91] International Electrotechnical Commission et al. *IEC 60870-5-104. Telecontrol equipment and systems ransmission protocols: Network access for IEC 60870-5-101 using standard transport profiles*. IEC.

[92] Adrian Segall. Distributed network protocols. *IEEE transactions on Information Theory*, 29(1):23–35, 1983.

[93] Zhu Xiaoxiang. Modbus protocol and programing [j]. *Electronic Engineer*, 7:016, 2005.

[94] EMC^2 Workpackage 11 Deliverable. emc^2 d11.1 system level requirements. *Artemis IA*, 2014.

[95] Sebastian Meiling, Till Steinbach, Thomas C Schmidt, and Matthias Wählisch. A scalable communication infrastructure for smart grid applications using multicast over public networks. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 690–694. ACM, 2013.

[96] Jiho Han and Deog-Kyoon Jeong. A practical implementation of ieee 1588-2008 transparent clock for distributed measurement and control systems. *IEEE transactions on instrumentation and measurement*, 59(2):433–439, 2010.