

GRADO EN INGENIERIA EN TECNOLOGIA DE
TELECOMUNICACION

TRABAJO FIN DE GRADO

***BANCO DE PRUEBAS PARA LAS
FUNCIONALIDADES PREEMPTION E
INTERCALADO DE TRAFICO CON PRIORIDAD
DEL ESTANDAR ETHERNET 802.3BR***

Alumno/Alumna: Santiago Santo-Tomás, Adrián

Director/Directora: Lázaro Arrotegui, Jesús

Curso: 2018-2019

Fecha: Bilbao, 24, Junio, 2019

Resumen

Hoy en día, las comunicaciones han alcanzado un nivel globalizado, en el que una gran cantidad de dispositivos muy diferentes interaccionan para intercambiar información. Surge la necesidad de encontrar una solución que permita la interoperabilidad de dispositivos de diferente naturaleza. Esta globalización ha llegado al mundo industrial y a las comunicaciones entre los equipos que la componen, dando lugar a la denominada Industria 4.0. Este proyecto se basa en el desarrollo de un banco de pruebas que permite comprobar el funcionamiento de nuevos sistemas que soporten el estándar Ethernet 802.3 en redes sensibles al tiempo. El estándar concreto sobre el que se trabaja es el 802.3br que aporta una mejora a la interoperabilidad, mediante la convivencia en un mismo enlace de tráfico de alta prioridad junto a tráfico no prioritario. Este banco de pruebas supone el inicio hacia el desarrollo de esta nueva tecnología y permite verificar el cumplimiento de la normativa del estándar por parte de equipos que se desarrollen en un futuro. Además, permite conocer y analizar desde un grado más detallado el funcionamiento y las necesidades de las redes sensibles al tiempo, en las que los enlaces deben cumplir las condiciones necesarias para ofrecer un servicio adecuado, de calidad y fiable.

Palabras clave: Preemption, Intercalado de Tráfico con Prioridad, Comunicaciones Industriales, Industria 4.0.

Abstract

Nowadays, communications have reached a globalized level, in which a large number of very different devices interact to exchange information. The need arises to find a solution that allows the interoperability of devices of different nature. This globalization has reached the industrial world and communications between the devices that make it up, giving rise to the so-called Industry 4.0. This project is based on the development of a testbench that allows checking the operation of new systems that support the Ethernet 802.3 standard in time sensitive networks. The concrete standard on which we work is the 802.3br that provides an improvement to interoperability, through coexistence in the same link of high priority traffic and non-priority traffic. This testbench marks the start towards the development of this new technology and allows verifying compliance with the standards of the norm that develop in the future. In addition, it allows knowing and analysing from a more detailed level the operation and the needs of time-sensitive networks, in which the links must meet the necessary conditions to offer an adequate, quality and reliable service.

Key words: Preemption, Interspersing Express Traffic, Industrial Communications, Industry 4.0.

Laburpena

Gaur egun, hedabideak maila batera heldu dira non, neurri handi batean, oso ezberdinak diren gailu askok elkarrekin izaten dituzten informazioa elkarrekin trukatzeko duten. Mota ezberdineko gailuen elkarrenganako operakortasuna ahalbidetzeko konponbidea aurkitzearen beharra sortzen da. Globalizazio hori industriako mundura eta mundu hori osatzen duen tresneriaren komunikazioetara heldu da, '4.0' deitutako industria sortuz. Proiektua frogatu banku baten garapenean oinarritzen da, zeinean sistema berri batzuen funtzionamendua frogatu ahal dugun, Ethernet 802.3 estandarra denborarekiko sentikorrak diren sareetan onartzen dutenek. Erabiltzen den estandarra, zehazki, '802.3br'-a da, interoperabilitatearekiko hobekuntza dakarrena, lotura berean lehentasun altuko eta lehentasun baxuko trafikoaren elkarbizitzaren bidez. Froga banku horrek aipatutako teknologia berriaren garatzearen hasiera dakar, eta etorkizunean garatutako gailuak araudia betetzen duten a la ez egiaztatzea ahalbidetzen du. Bestalde, denborarekiko sentikorrak diren sareen beharrak eta funtzionamendua ezagutzea eta aztertzea baimentzen du, non loturak beharrezko baldintzak bete behar dituzten egokia, kalitatezkoa eta fidagarria den zerbitzua eskaintzeko.

Hitz gakoak: Preemption, Trafikoaren Arteko Tartekatze Lehentasuna, Industriako Komunikazioak, Industria 4.0

Acrónimos

- VHSIC: Very High Speed Integrated Circuit
- HDL: Hardware Description Language
- VHDL: combinación de VHSIC y HDL
- QoS: Quality of Service
- IEEE: Institute of Electrical and Electronics Engineers
- RAMI 4.0: Reference Architectural Model Industrie 4.0
- TSN: Time-Sensitive Networking
- CIM: Computer Integrated Manufacturing
- OSI: Open System Interconnection
- LAN: Local Area Network
- MAN: Metropolitan Area Network
- MAC: Media Access Control
- SFD: Start Frame Delimiter
- IPV4: Internet Protocol Version 4
- FCS: Frame Check Sequence
- YANG: Yet Another Next Generation
- TSN TG: Time-Sensitive Networking Task Group
- AVB: Audio Video Bridging
- OLE: Object Linking and Embedding
- OPC UA TSN: OLE for Process Control Unified Architecture for Time-Sensitive Networking
- eMAC: express Media Access Control
- pMAC: preemptable Media Access Control
- MAC DA: Media Access Control Destination Address
- MAC SA: Media Access Control Source Address
- SMD: Start mPacket Delimiter

- CRC32: Cyclic Redundancy Check 32-bits
- mCRC: mPacket Cyclic Redundancy Check
- FPGA: Field Programmable Gate Array
- AXI: Advanced eXtensible Interface
- EDT: Estructura de Descomposición de Trabajo

Índice

Resumen.....	2
Abstract.....	3
Laburpena.....	4
Acrónimos.....	5
Índice.....	7
Índice de figuras.....	9
Índice de tablas.....	13
1. Introducción.....	15
2. Contexto.....	17
3. Objetivos y alcance del trabajo.....	21
4. Beneficios que aporta el trabajo.....	23
4.1 Beneficios técnicos.....	23
4.2 Beneficios económicos.....	23
4.3 Beneficios sociales.....	24
5. Análisis del estado del arte.....	25
5.1 Industria 4.0.....	25
5.2 Comunicaciones Industriales.....	30
5.3 Ethernet.....	34
5.4 Redes Sensibles en el Tiempo.....	37
5.5 Intercalado de tráfico prioritario y no prioritario.....	41
6. Análisis de alternativas.....	50
6.1 Alternativas de diseño.....	50
6.1.1 Módulo de transmisión.....	50
6.1.2 Módulo de recepción.....	51
6.1.3 Selección de alternativa de diseño.....	51
6.2 Alternativas de funcionalidades de bloques.....	51
6.2.1 Módulos básicos controlados.....	51
6.2.2 Módulos complejos pseudoaleatorios/aleatorios.....	52

6.2.3 Selección de alternativa de funcionalidades de bloques.....	52
6.3 Alternativas de implementación	52
6.3.1 Implementación en sistema real.....	52
6.3.2 Implementación en entorno de simulación	53
6.3.3 Selección de alternativa de implementación	53
7. Selección, descripción y diseño de la solución propuesta.....	54
7.1 Perspectiva global del sistema.....	55
7.2 Banco de pruebas.....	56
7.3 Bloque de control de generación	57
7.4 Bloque de generación	60
7.5 Bloque ensamblador de tramas	62
7.6. Bloque exportador de datos generados	67
8. Descripción de tareas, fases, y procedimientos	68
9. Diagrama de Gantt/Cronograma	71
10. Análisis de costes.....	72
13. Descripción de los resultados	75
14. Conclusiones.....	86
Referencias.....	88
Anexos	91
A. Descripción detallada de la solución	91
B. Diagramas de bloques	103
C. Diagramas de estados	106
D. Código.....	109

Índice de figuras

Figura 1: Tecnologías del ecosistema de la Industria 4.0 [3]	18
Figura 2: Historia de la evolución industrial [8]	26
Figura 3: Arquitectura RAMI 4.0 [11]	28
Figura 4: Grupos de trabajo TSN más importantes [13]	29
Figura 5: Tendencia de los sistemas de comunicación actuales [1]	30
Figura 6: Pirámide CIM [15]	31
Figura 7: Tipo de tráfico y velocidad de operación de los niveles de la pirámide CIM [14]	32
Figura 8: Evolución de los protocolos de comunicación industrial [4]	33
Figura 9: Comunicación entre diferentes protocolos mediante mecanismos TSN [18]	33
Figura 10: Nivel de enlace del modelo OSI, para el caso tradicional del estándar Ethernet 802.3 [19]	35
Figura 11: Estructura de tramas Ethernet según el estándar tradicional 802.3 [19]	35
Figura 12: Componentes de los estándares desarrollados por los grupos de trabajo TSN [2]	38

Figura 13: Tendencia de estandarización mediante la unificación de la red de comunicaciones industriales [6].....	41
Figura 14: Tiempos y retardos propios de las comunicaciones [20].....	42
Figura 15: Nivel de enlace del modelo OSI, implementando el estándar 802.3br de intercalado de tráfico con prioridad [21]	44
Figura 16: Estructura de tramas Ethernet bajo el estándar 802.3br y la funcionalidad Preemption [22].....	46
Figura 17: Diagrama conceptual del funcionamiento del sistema global.....	56
Figura 18: Diagrama conceptual del funcionamiento del bloque “Banco de Pruebas”	56
Figura 19: Diagrama conceptual del funcionamiento del bloque de control de generación.....	59
Figura 20: Diagrama conceptual del funcionamiento del bloque de generación	61
Figura 21: Diagrama conceptual del funcionamiento del bloque ensamblador de tramas	64
Figura 22: Diagrama de Gantt del proyecto	71
Figura 23: Secuencia de generación.....	75

Figura 24: Secuencia generada por el Banco de Pruebas en el entorno de simulación	76
Figura 25: Inicio de primera trama generada por el Banco de Pruebas	77
Figura 26: Final de primera trama generada por el Banco de Pruebas	77
Figura 27: Inicio de segunda trama generada por el Banco de Pruebas.....	77
Figura 28: Final de segunda trama generada por el Banco de Pruebas	78
Figura 29: Inicio de tercera trama generada por el Banco de Pruebas	78
Figura 30: Final de tercera trama generada por el Banco de Pruebas.....	79
Figura 31: Inicio de cuarta trama generada por el Banco de Pruebas	79
Figura 32: Final de cuarta trama generada por el Banco de Pruebas.....	79
Figura 33: Inicio de quinta trama generada por el Banco de Pruebas.....	80
Figura 34: Final de quinta trama generada por el Banco de Pruebas	80
Figura 35: Datos de secuencia generada exportados a fichero de texto plano	81
Figura 36: Datos de secuencia generada exportados a fichero de capturas PCAP	82
Figura 37: Detalles de primera trama obtenidos mediante Wireshark.....	82

Figura 38: Detalles de segunda trama obtenidos mediante Wireshark	83
Figura 39: Detalles de tercera trama obtenidos mediante Wireshark.....	83
Figura 40: Detalles de cuarta trama obtenidos mediante Wireshark.....	84
Figura 41: Detalles de quinta trama obtenidos mediante Wireshark	84
Figura 42: Diagrama de bloques del sistema global.....	103
Figura 43: Diagrama de bloques de “Banco de Pruebas”	103
Figura 44: Diagrama de bloques de “Bloque de control de generación”	104
Figura 45: Diagrama de bloques de “Bloque de generación”	104
Figura 46: Diagrama de bloques de “Bloque ensamblador de tramas”	105
Figura 47: Diagrama de estados de bloque de control de generación	106
Figura 48: Diagrama de estados de bloque de control de ensamblado de tramas	107
Figura 49: Diagrama de estados de bloque de selección de generador de ensamblado de tramas.....	108

Índice de tablas

Tabla 1: Valores SMD de tramas Ethernet según estándar 802.3br [21]	48
Tabla 2: Valores FRAG_COUNT de tramas Ethernet según estándar 802.3br [21]	48
Tabla 3: Descripción de tareas del proyecto	68
Tabla 4: Presupuesto. Horas internas	73
Tabla 5: Presupuesto. Gastos	74
Tabla 6: Presupuesto. Amortizaciones	74
Tabla 7: Presupuesto. Coste total del proyecto	74
Tabla 8: Valores de datos de la carga útil de las tramas generados por el sistema	97
Tabla 9: Ficheros que contienen el código de los bloques generales del sistema	109
Tabla 10: Ficheros que contienen el código de los bloques internos del bloque “DATA CONTROL”	110
Tabla 11: Ficheros que contienen el código de los bloques internos del bloque “DATA GENERATOR”	110

Tabla 12: Ficheros que contienen el código de los bloques internos del bloque
“FRAME GENERATOR” 111

Tabla 13: Ficheros que contienen el código Python para procesamiento de
datos exportados 111

1. Introducción

En este documento se expone el proceso llevado a cabo para desarrollar un banco de pruebas para sistemas que implementen el estándar 802.3br basado en el concepto denominado Preemption, mediante lenguaje VHDL y en un entorno de simulación controlado.

Inicialmente, se parte de una toma de contacto teórica y una contextualización del tema en la actualidad, en la situación social, tecnológica e industrial. Una vez obtenida la contextualización, se obtienen las necesidades y requerimientos a solventar de la situación actual y se elaboran las posibles alternativas. Se identifica cada alternativa, junto a sus puntos favorables y desfavorables, para elegir finalmente la solución a desarrollar.

Partiendo de la solución seleccionada, es necesario obtener con mayor detalle los requerimientos necesarios para desarrollar el sistema. También se identifican los beneficios concretos aportados por el proyecto. Se analizan entonces los posibles riesgos del desarrollo e implementación de la idea final.

Una vez situada la propuesta en su contexto, habiendo identificado los puntos a desarrollar y aquellos aspectos necesarios para ello, se expone el estado actual de las tecnologías relacionadas con el trabajo. Se estudian con detalle los aspectos tecnológicos necesarios para comprender las funcionalidades y características que debe cumplir el proyecto, obteniéndose también, los conocimientos adecuados para la correcta elaboración del sistema, a nivel de desarrollo de código y simulación.

Con los conceptos teóricos afianzados, se describe de forma general la solución aportada, analizando cada uno de los componentes del sistema, indicando sus

características y funcionalidades, viendo qué aspectos aportan a la solución completa.

Es necesario realizar un análisis de la metodología seguida. Por ello, se obtiene una lista de las tareas que se llevan a cabo durante el proceso, junto a los intervalos de tiempo correspondientes.

Cuando el sistema está desarrollado completamente, se procede a la simulación del mismo, analizando su comportamiento y observando que los resultados son adecuados para la solución propuesta.

También se añaden los datos de personal y los datos económicos, viendo el posible impacto económico en una situación real.

Para finalizar, se añaden las conclusiones obtenidas tras el desarrollo del sistema y de su simulación, aportando los detalles necesarios para obtener una idea general de los aspectos aportados por el proceso seguido.

En la última parte de la memoria se incluye la bibliografía y las referencias utilizadas para la obtención de los conocimientos necesarios para la elaboración del sistema completo, tanto para la contextualización, como para el desarrollo a nivel de diseño.

En los anexos se incluyen los diagramas de bloques del sistema global, junto a los diagramas de bloques de los elementos internos. También se incorpora el código en lenguaje VHDL que permite la implementación en el entorno de simulación del sistema completo. Junto a esta información, se introduce una explicación detallada de las características de cada uno de los elementos individuales del sistema, a fin de completar la información incluida en la memoria.

2. Contexto

Actualmente, el entorno de las comunicaciones se encuentra en un estado globalizado, prácticamente accesible desde cualquier punto del planeta. Es por ello, que cada día es mayor el número de equipos y dispositivos conectados a la red. Estos dispositivos presentan naturalezas de lo más variadas, desde equipos de uso cotidiano, como pueden ser ordenadores personales, teléfonos móviles o televisores, hasta equipos industriales, como puede ser un robot de una fábrica o una máquina de una cadena de producción. Debido a esta variedad de posibles elementos que interaccionan entre sí, intercambiando flujos de información, surge la necesidad de establecer una serie de normas que permitan la interoperabilidad adecuada entre ellos. Partiendo de esta premisa, en los últimos años han surgido numerosas iniciativas que luchan por lograr una estandarización generalizada, a fin de crear un ecosistema adecuado para la comunicación entre todos los posibles equipos conectados [1], [2].

Una iniciativa importante es la denominada Industria 4.0. Tiene su origen en Alemania y rápidamente se expande logrando causar un gran impacto en el mundo de las telecomunicaciones. Se basa en crear el ecosistema que permita la interacción entre todos los equipos de la red de comunicación, estableciendo una relación entre los elementos físicos y su representación virtual dentro de la red. Los equipos se conocen como sistemas ciber-físicos y se establecen las bases a cumplir para el correcto desempeño de la funcionalidad, dentro del entorno industrial. Es tal el impacto de esta ideología, que se considera como la cuarta revolución industrial.



Figura 1: Tecnologías del ecosistema de la Industria 4.0 [3]

Teniendo en cuenta el nuevo entorno industrial y las diferentes normas establecidas para mejorarlo, aparecen tendencias e ideas centradas en desarrollar cada uno de los aspectos individuales necesarios para el correcto crecimiento del proyecto. Esto significa, que cada funcionalidad particular será desempeñada por un sector determinado, siempre buscando el funcionamiento correcto de la red global.

Surge entonces el interés por definir nuevas normas y estándares de dichas funcionalidades, de cara a que, en el futuro, todos los equipos puedan comunicarse sin impedimentos, logrando una red que se comunica en todo momento (en tiempo real), capaz de gestionar cantidades de información muy elevadas. A partir de estos estándares, aparecen nuevas necesidades no contempladas en la etapa inicial y concretas del sistema de comunicación sobre el que se trabaja.

En el caso concreto de las comunicaciones industriales, se ha producido también una evolución hacia la estandarización. Inicialmente, cada sistema de comunicaciones industriales, utilizaba un protocolo propietario. Esto dificulta la comunicación entre equipos de sistemas diferentes. En los últimos años, la tendencia también ha sido hacia la búsqueda de la unificación de los protocolos,

o al menos, hacia la compatibilidad entre ellos mediante sistemas intermedios [4]–[6].

Tras varios años de investigación y análisis de las diferentes alternativas, se ha elegido el protocolo de enlace Ethernet como el medio de comunicación generalizado para la industria. Esto se debe a su bajo coste y a su accesibilidad y funcionalidades. Partiendo de esta elección, se ha deducido que es necesario implementar una serie de funcionalidades adicionales, no contempladas inicialmente en este tipo de comunicación. Estas funcionalidades son añadidas con la introducción del concepto de TSN. Analizando la nueva situación, se desarrollan aquellos estándares que permitan al sistema funcionar según los conceptos expuestos, unificando Ethernet y TSN [6].

En esta nueva tesitura, aparece el concepto de QoS y junto al mismo, las características que se deben cumplir para que el sistema sea el adecuado. Dentro de esta idea, aparece el caso concreto de la necesidad de asegurar una latencia determinada, ante un tipo de tráfico de información susceptible al tiempo y al retardo. Esto significa, que existe un tipo de tráfico de información prioritario y que debe llegar sin fallos y con total urgencia al destinatario.

Teniendo en cuenta la globalidad de la red y los diferentes tipos de protocolos utilizados, aparece la necesidad de introducir un mecanismo que permita intercalar tráfico prioritario y no prioritario, de cada uno de estos protocolos, a través del mismo enlace, permitiendo que todos los dispositivos puedan comunicarse por el mismo medio, sin perjudicar a las señales o a la información urgente. Se desarrolla entonces un nuevo estándar, IEEE 802.3br que se incluye dentro del estándar de Ethernet IEEE 802.3, en el que se definen las características y cualidades que debe cumplir un sistema que soporte y trabaje con el intercalado de tráfico prioritario y no prioritario.

Es necesario, por tanto, elaborar este nuevo tipo de funcionalidades, que se encuentran en un estado conceptual y de desarrollo inicial.

Ante estas necesidades y teniendo en cuenta la tendencia actual, en la que cada día es más inminente la implementación de estos nuevos sistemas, se deben desarrollar de forma paralela a los estándares, bancos de pruebas que permitan no solo comprender y estudiar de forma más detallada las funcionalidades, sino también de probar los nuevos dispositivos que se fabriquen en un futuro y que trabajen sobre este tipo de protocolos de comunicación.

En este entorno, se basa este proyecto, desarrollando un banco de pruebas, que supone un punto inicial hacia el desarrollo de las funcionalidades del intercalado de tráfico urgente para comunicaciones Ethernet.

3. Objetivos y alcance del trabajo

El proyecto presenta un objetivo principal, el cual consiste en la obtención de un banco de pruebas para la funcionalidad sobre latencia del protocolo Ethernet, denominada Preemption, basada en el estándar IEEE 802.3br. Para lograrlo, se deben cumplir una serie de objetivos parciales y existe también un objetivo secundario. Desglosando cada elemento:

- Objetivo principal:

Elaborar un sistema que realice la función de banco de pruebas para el estándar indicado, permitiendo poner a prueba diferentes equipos y sistemas bajo entornos de simulación controlados. El banco de pruebas a desarrollar, será un núcleo para el posterior crecimiento del sistema, permitiendo la escalabilidad en la complejidad. Debe tomar una secuencia de ejecución de directrices de generación y transmisión de tramas Ethernet y generar a partir de la misma los datos y señales necesarios. Es decir, a partir de la secuencia definida, el sistema otorga tramas de prueba para el equipo colocado en la siguiente etapa.

- Objetivos parciales:

- Desarrollar un sistema que permita controlar los datos a generar en función de una secuencia preestablecida. Debe leer la secuencia e indicar a los demás bloques la forma en que se debe operar.
- Codificar un sistema que se encargue de recibir las directrices de los bloques de control y generar datos válidos para los campos de información de las tramas de prueba.
- Desarrollar un sistema que ensamble las tramas de prueba de forma correcta, permitiendo el funcionamiento en enlaces que cumplen con el estándar 802.3.

- Codificar un sistema que conecte los diferentes bloques de forma adecuada, permitiendo el control global del equipo.
- Generar el sistema necesario para la simulación de cada uno de los elementos, comprobando el correcto funcionamiento del sistema y verificando el resultado final.
- Objetivo secundario:

Establecer el punto inicial de la tecnología, para permitir su continuidad en futuros proyectos. Supone un nuevo paso hacia la estandarización del protocolo para comunicaciones Ethernet. Permite el estudio y el análisis de la funcionalidad del intercalado de tráfico con prioridad en redes sensibles al tiempo.

En cuanto al alcance del proyecto, se basa en el desarrollo del núcleo de un sistema cuya función es actuar como banco de pruebas para sistemas que implementan el intercalado de tráfico urgente, en redes sensibles al tiempo, utilizando el enlace Ethernet. Finalmente, se extraen las conclusiones que verifican la viabilidad del proyecto y su mejora de la situación actual del protocolo de comunicación utilizado.

4. Beneficios que aporta el trabajo

Los beneficios aportados por el proyecto seleccionado pueden dividirse en diferentes sectores o ámbitos.

4.1 Beneficios técnicos

- Obtención de un conocimiento más extenso y detallado de la funcionalidad del estándar expuesto. Permite avanzar en el desarrollo de la tecnología, aplicando un modelo escalable en complejidad y extensión.
- Avance hacia una estandarización global. Defiende la idea de la unificación de protocolos de variada naturaleza, dentro de una misma red. Por tanto, es un proyecto que está a favor de la interoperabilidad de equipos.

4.2 Beneficios económicos

- Ingresos económicos. Una vez desarrollado, permite obtener ingresos mediante la realización de pruebas con el sistema, aportando a los clientes los datos de interés obtenidos tras las simulaciones. Permite realizar diagnósticos de sistemas de terceros para su control, corrección y producción.
- Expansión empresarial. Es un buen medio de captación de posibles clientes, mediante la presentación de pruebas iniciales remotas, aplicando el sistema desarrollado, antes de comenzar con la actividad económica.
- Unificación y estandarización. Establece el punto inicial hacia la interoperabilidad de los equipos existentes. Permite seguir utilizando los equipos ya adquiridos, añadiendo la posibilidad de

intercomunicación entre ellos. Supone la posibilidad de amortización de los equipos en propiedad, junto a los equipos de nueva adquisición.

- Control de errores en nuevos sistemas. Permite realizar pruebas en prototipos antes de llegar a la producción para evitar posibles problemas. Se evitan gastos innecesarios de producción, al poder analizar los equipos en entornos controlados y de interés.

4.3 Beneficios sociales

- Divulgación de la tecnología. Mediante el sistema, se puede exponer la funcionalidad ante personas del mismo sector en eventos sociales y empresariales, como pueden ser convenciones o congresos, a fin de promulgar la idea de la estandarización y la necesidad del desarrollo en este campo.

5. Análisis del estado del arte

Para comprender en su totalidad los requerimientos y necesidades que solventa el proyecto, es necesario estudiar y analizar ciertos conceptos. De esta forma, se obtiene una idea del contexto global del trabajo y se puede situar la funcionalidad desarrollada dentro del contexto tecnológico en que se encuentra. Se tiene entonces una idea más completa de la situación y del objetivo del proyecto, junto a las soluciones que aporta.

5.1 Industria 4.0

Este término hace referencia a una iniciativa estratégica introducida por el gobierno alemán recientemente. Su principal objetivo consiste en la transformación de la producción industrial a través de la digitalización y explotación de los potenciales de las nuevas tecnologías. Este concepto ha sido extendido hacia otros países, generando diferentes corrientes, todas ellas en busca del mismo objetivo. Esto se debe a que, según los últimos estudios, se estima que un 46% de la economía global podría beneficiarse de estas estrategias y del Internet Industrial. Es tal la dimensión de su repercusión y el auge de esta tendencia, que ha sido considerada como la cuarta revolución industrial [7].

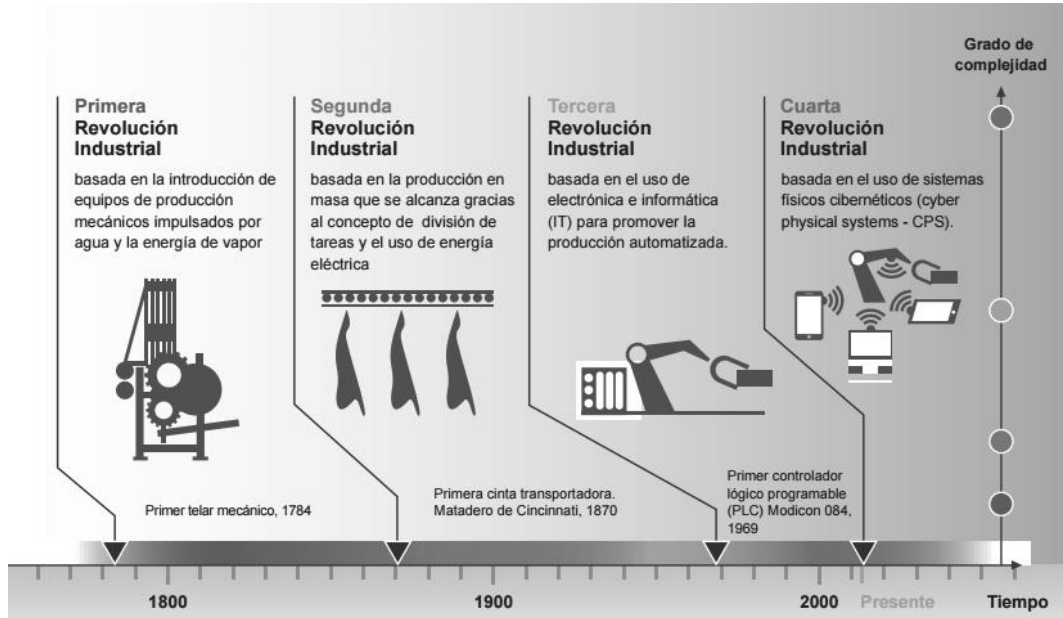


Figura 2: Historia de la evolución industrial [8]

Según el propio gobierno alemán, en particular los ministerios que fomentan la estrategia ("Federal Ministry for Economics Affairs and Energy" y "Federal Ministry of Education and Research"), el concepto de Industria 4.0 se refiere a la red inteligente entre máquinas y procesos de producción de los que consta la industria, apoyados por las tecnologías de la información y de las comunicaciones [9]. Estos mismos organismos, definen las características que ha de cumplir dicha red:

- **Producción flexible:** En el proceso de producción, varias empresas y elementos interactúan entre sí para la elaboración del producto final. Una buena coordinación en el proceso permite obtener resultados más satisfactorios. La digitalización del proceso permite una mejor planificación.
- **Fábrica cambiante:** Futuras infraestructuras de producción pueden ser construidas de forma modular, a fin de lograr una mejor adaptación

hacia las nuevas tendencias y necesidades que pueden surgir. Se pueden adaptar de esta forma más fácilmente a los requerimientos concretos de la producción para cada caso particular. Mejora la productividad y la eficiencia.

- Soluciones orientadas al consumidor: La interacción entre clientes y productores busca ser más dinámica y cercana, permitiendo una mejor adaptación del proceso a las demandas particulares del cliente. Además, los propios productos ya elaborados, pueden aportar información al sistema durante su vida útil, a fin de mejorar la producción futura.
- Logística optimizada: Las rutas logísticas pueden ser calculadas mediante los datos obtenidos del sistema y las propias máquinas pueden realizar demandas (productos, servicios...) en cuanto sean requeridos. Esto aumenta la efectividad y la optimización del flujo de bienes y materiales.
- Uso de los datos y la información: Los datos obtenidos del proceso de producción y de las condiciones del producto durante su vida útil se almacenan y analizan de forma combinada. Este análisis aporta una guía de como elaborar los productos de manera eficiente. Supone el inicio de un nuevo modelo de negocio y de servicios.
- Economía circular eficiente: El ciclo de vida completo del producto se considera como una fuente de información continua.

Para lograr el objetivo de la estrategia elaborada por el gobierno alemán, se propone a su vez una plataforma de referencia para la digitalización. Esta plataforma se basa en la búsqueda de valor añadido sobre el flujo de información existente y futuro, permitiendo su aplicación en sectores muy variados. Trabaja sobre el precepto de la unificación y la estandarización.

La plataforma propuesta consiste en un modelo de referencia y pretende ser la guía para la implementación de la Industria 4.0. Se denomina RAMI4.0 (Reference Architectural Model Industrie 4.0) y consiste en un mapa tridimensional en el que se muestran los aspectos más importantes de la industria 4.0. Este planteamiento asegura que todos los participantes involucrados comparten una perspectiva común y desarrollan un entendimiento también común. El grafo tridimensional relaciona el ciclo de vida del producto, las cuestiones básicas de las ideas de negocio y la jerarquía de la manufactura [10], [11].

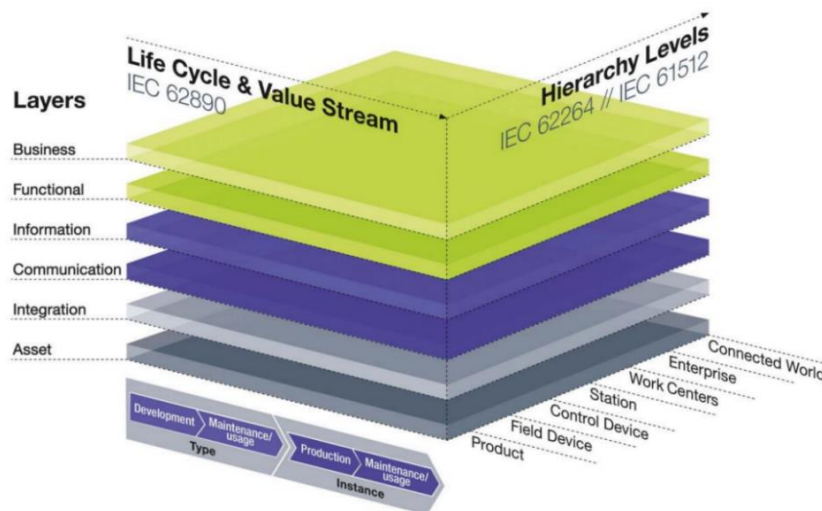


Figura 3: Arquitectura RAMI 4.0 [11]

Teniendo en cuenta estas tres dimensiones, se pretende analizar toda la información aportada por los productos durante el ciclo completo de vida, para obtener conclusiones sobre las cuestiones básicas que afectan al propio negocio sobre el que se sustenta el sector de la industria de interés, para una serie de jerarquías de manufactura definidas.

Partiendo de los conceptos definidos por la propia estrategia, se busca un camino hacia una industria en la que todos los equipos se comunican entre sí de

forma constante. La información intercambiada permite analizar el sistema de producción y los propios productos, para buscar medidas que aporten una mayor eficiencia a la manufactura. Se trata también de una industria escalable y dinámica [7].

En este ecosistema industrial, los flujos de información y los equipos conectados son muy variados, por lo que se fomenta una interoperabilidad y una correcta conectividad. Es por tanto necesario, buscar paralelamente, nuevas medidas que aporten las características necesarias para la convergencia en una misma red. Es decir, se deben desarrollar plataformas tecnológicas y protocolos de comunicación que garanticen el flujo, procesamiento y seguridad de la información. Cuanto mejores sean estas funcionalidades, mejores serán los resultados de la implementación de la estrategia indicada [12].

Las ideas y medidas aportadas por la plataforma de la Industria 4.0 han sido aceptadas por una gran parte de las potencias tecnológicas mundiales y actualmente ya juega un papel importante en el panorama industrial mundial. Estados Unidos, Alemania, Francia, China y Japón ya colaboran de forma activa en la búsqueda de la estandarización global [13].



Figura 4: Grupos de trabajo TSN más importantes [13]

Partiendo de ello, aparecen grupos de trabajo que buscan desarrollar los estándares necesarios, así como las funcionalidades de los mismos. En la conferencia "TSN/A Conference 2018" se demuestra la presencia de estos grupos y se exponen los nuevos caminos que ha de seguir la industria 4.0, demostrando que la unificación se está logrando gradualmente y que cada día las mejoras en los protocolos de comunicación son mayores [1].

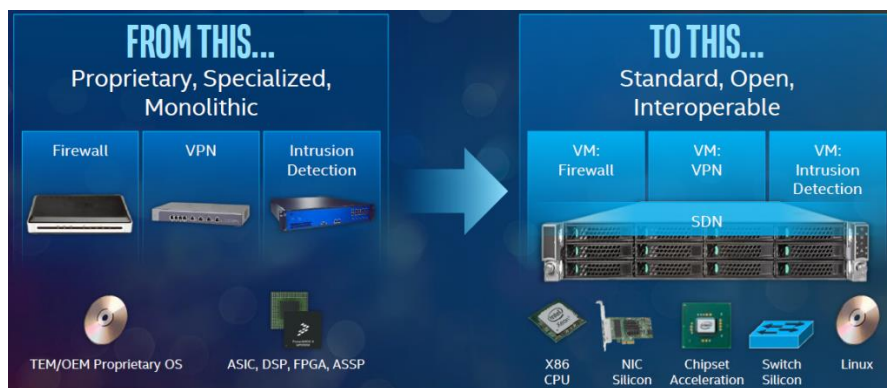


Figura 5: Tendencia de los sistemas de comunicación actuales [1]

Los avances actuales indican que la unificación adecuada debe ser a través del protocolo de comunicaciones Ethernet, gracias a sus características y a la versatilidad que otorga para la interoperabilidad. Supone un buen punto de partida ya que las características actuales permiten una buena comunicación y únicamente se debe focalizar el desarrollo en aquellos aspectos que mejoren las funcionalidades concretas de la Industria 4.0. Es decir, el estándar de Ethernet presenta la tecnología base necesaria para la correcta comunicación entre equipos [1], [2].

5.2 Comunicaciones Industriales

Se pueden definir como el área de la tecnología que estudia la transmisión de información entre circuitos y sistemas electrónicos utilizados para llevar a cabo

tareas de control y gestión del ciclo de vida de los productos industriales. Deben resolver la problemática de la transferencia de información entre los equipos de control del mismo nivel y entre los correspondientes a los niveles contiguos de la pirámide CIM (Computer Integrated Manufacturing) [14].

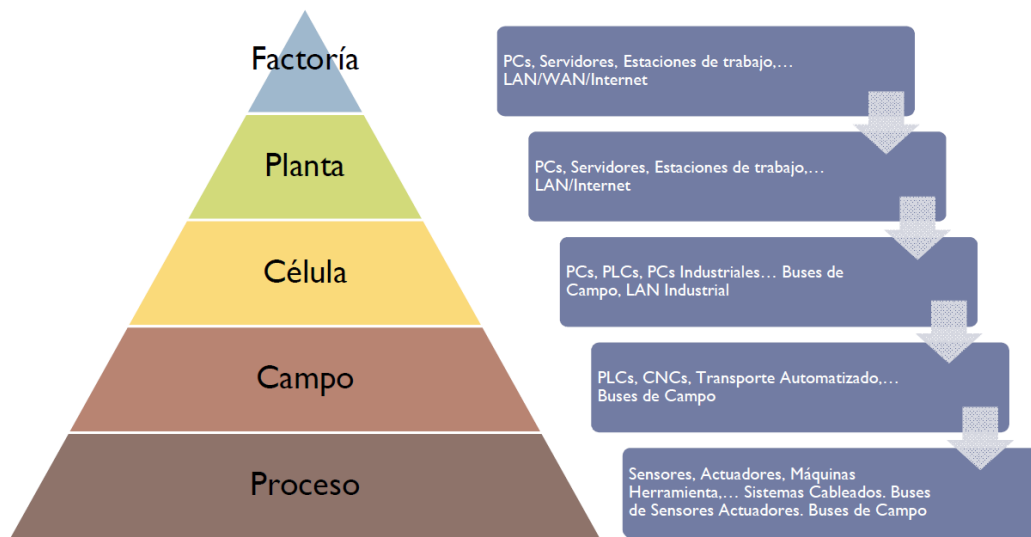


Figura 6: Pirámide CIM [15]

Teniendo en cuenta dicha pirámide, se pueden diferenciar dos tipos de tráfico, fácilmente identificables en la Figura 7:

- Tráfico de gran volumen, utilizado en los niveles superiores. Este tipo de tráfico no presenta un tiempo de respuesta crítico.
- Tráfico de bajo volumen, pero en tiempo real. Esto significa que se exigen tiempos de transmisión más rápidos y se espera un comportamiento de tipo determinista. Este tipo de tráfico se corresponde con los niveles inferiores.

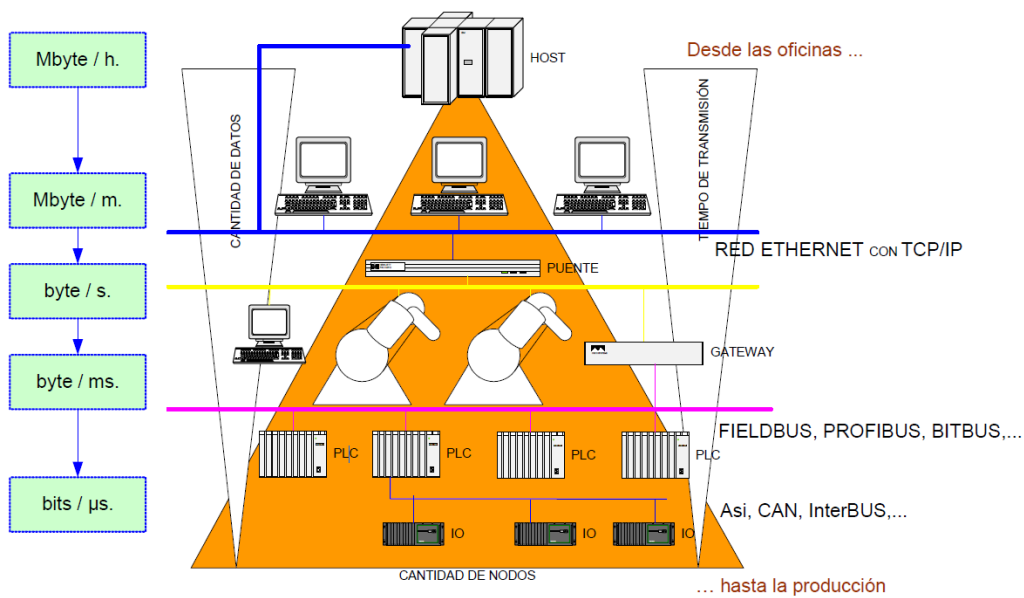


Figura 7: Tipo de tráfico y velocidad de operación de los niveles de la pirámide CIM [14]

Teniendo en cuenta el modelo OSI, y que las funcionalidades de los niveles más bajos del mismo son sensibles al tiempo, es necesario introducir modificaciones y mejoras en los mecanismos encargados de la comunicación en estos niveles inferiores. Es por ello que surge la idea de implementar las comunicaciones industriales a través del protocolo Ethernet. Unido a esto, surge el concepto de Redes Sensibles al Tiempo (Time Sensitive Networking - TSN), que trata de solventar las necesidades de este tipo de redes en que los retardos y tiempos de transmisión son críticos en cuanto a calidad y fiabilidad de la comunicación. Por tanto, se trata de combinar TSN junto al protocolo Ethernet [4], [6], [16], [17].

Actualmente, los tipos de protocolos para redes de comunicación industrial son muy variados. Para el caso de buses de campo, que son los que afectan a los niveles más bajos de comunicación, enlace y físico, se corresponden con los protocolos del apartado de Comunicaciones Industriales de la Figura 8 [16].

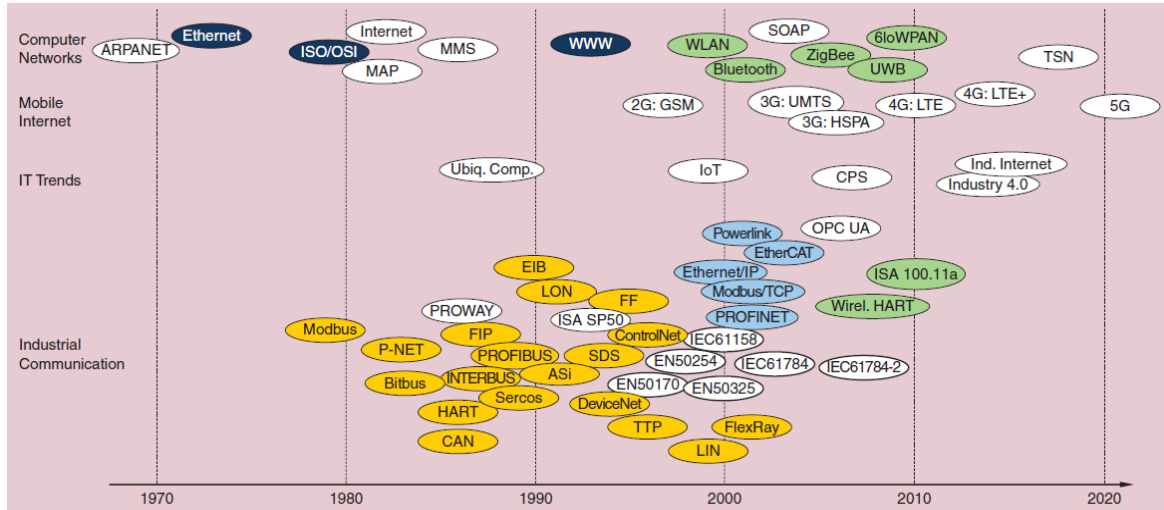


Figura 8: Evolución de los protocolos de comunicación industrial [4]

Como se puede apreciar, la gran variedad perjudica la interoperabilidad y es necesario buscar la estandarización. Esta es la naturaleza del concepto TSN, buscando la compatibilidad de los diferentes protocolos existentes y permitiendo la comunicación futura, en un ecosistema de medios de comunicación variados.

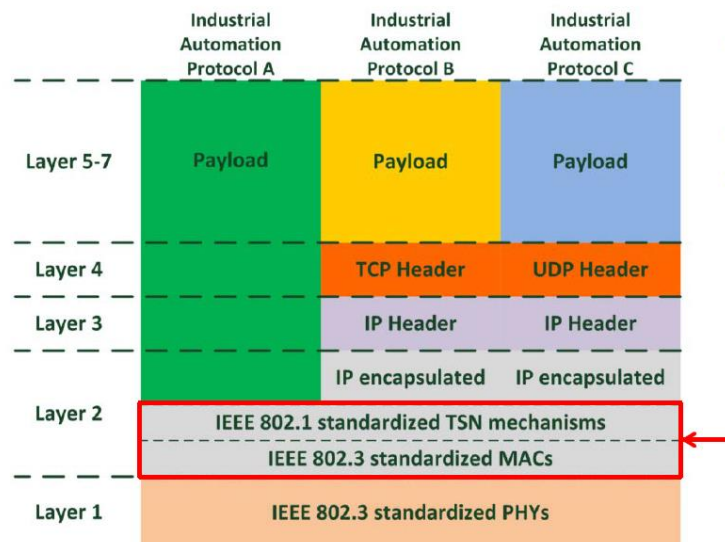


Figura 9: Comunicación entre diferentes protocolos mediante mecanismos TSN [18]

Mediante un estándar de los niveles de comunicación inferiores (TSN y Ethernet), se otorgan las funcionalidades adecuadas al protocolo de comunicación Ethernet, permitiendo que protocolos de niveles superiores muy variados, puedan comunicarse entre sí sin ningún tipo de impedimento.

5.3 Ethernet

Teniendo en cuenta la revisión del estándar sobre la que se basa este proyecto, este apartado trata de definir el estándar Ethernet previo a la introducción de la tecnología TSN.

Se trata de un estándar internacional para Redes de Área Local y Metropolitana (Local and Metropolitan Area Networks - LANs y MANs). Emplea un método de acceso al medio compartido y define el protocolo y el formato de tramas según el estándar IEEE 802.3, para el intercambio de información. Este estándar pretende abarcar una gran variedad de tipos de medios y técnicas de acceso a dichos medios. Se define para una selección determinada de velocidades de operación y usa un control de acceso al medio común (Media Access Control - MAC). Establece cómo la capa física codifica las tramas para la transmisión y decodifica las tramas recibidas mediante la modulación adecuada para la velocidad de operación, el medio de transmisión utilizado y la longitud del enlace soportado [19].

Este estándar, define las características del cableado y señalización del nivel físico del modelo OSI, así como los formatos de tramas del nivel de enlace del mismo modelo.

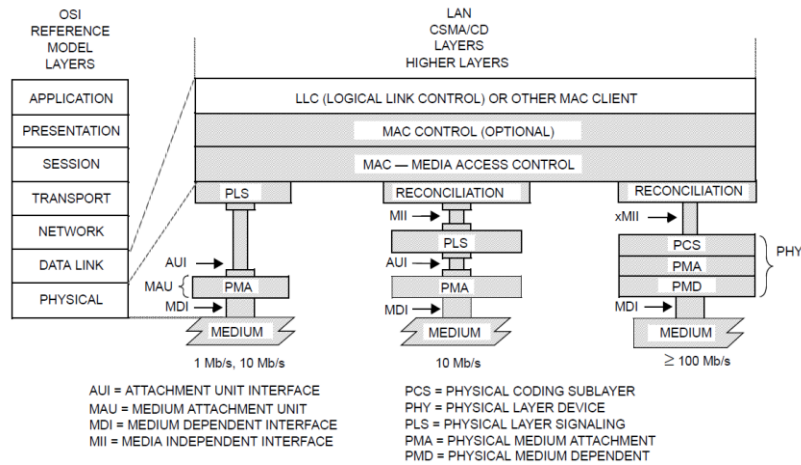


Figura 10: Nivel de enlace del modelo OSI, para el caso tradicional del estándar Ethernet 802.3 [19]

Por tanto, permite que, a través de un medio común, se puedan transmitir tramas procedentes de niveles superiores diferentes. Aporta versatilidad al sistema.

Según el propio estándar, las tramas deben cumplir con una secuencia de campos determinada y preestablecida, en función de su tipo, tamaño y secuencia dentro del flujo global de intercambio de información. Dicha secuencia se muestra en la Figura 11.

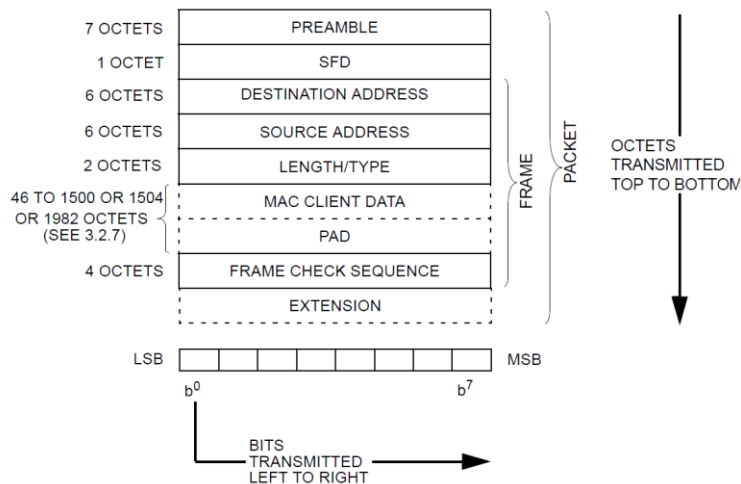


Figura 11: Estructura de tramas Ethernet según el estándar tradicional 802.3 [19]

Se debe iniciar la transmisión con un preámbulo determinado y continuar con la secuencia de inicio de trama. A continuación, se indican los campos de control referentes al protocolo utilizado (origen, destino, tipo) y se introduce la información útil a transmitir. Una vez enviados todos los datos, se envía una secuencia de verificación de la trama y se introduce un espaciado antes de comenzar con la emisión de la siguiente trama. Los octetos transmitidos, se emiten desde el bit menos significativo hasta el bit más significativo.

Cabe destacar algunos valores de los diferentes octetos para el correcto funcionamiento de intercambio de información:

- Preamble o preámbulo: contiene una serie de octetos que permiten la sincronización del sistema de recepción y su preparación para la extracción de los datos de la trama. El valor de los octetos es constante, siendo el mismo la secuencia “10101010” en valor binario o “0x55” en valor hexadecimal.
- SFD o “Start Frame Delimiter”: consiste en un octeto que, a continuación del preámbulo, indica que la información propia de la trama va a ser transmitida. Para el caso general, este valor será siempre de “10101011” en valor binario o de “0xD5” en valor hexadecimal.
- Destination Address y Source Address: direcciones de destino y de origen respectivamente. Ambas están compuestas por seis octetos, de forma que los tres primeros de cada dirección se corresponden con el fabricante del equipo y los tres últimos con la identificación única del dispositivo en cuestión. El formato de una dirección MAC será, por tanto, “hh:hh:hh:hh:hh:hh” donde “hh” se corresponde con el valor hexadecimal de cada octeto de la dirección del dispositivo. En el caso concreto de difusión de tramas a todos los equipos conectados a nivel de enlace, el valor de la dirección MAC de destino será “FF:FF:FF:FF:FF:FF”.

- Length/Type: campo correspondiente a la longitud de la trama o al tipo de comunicación en el caso de Ethernet. Para el caso de estudio se utiliza el valor hexadecimal “0x0800” que corresponde con el protocolo superior IPV4, definiendo de esta forma el tipo de comunicación de cara a simulación.
- MAC Client Data: carga útil de la trama procedente de los niveles superiores del modelo OSI. Depende de cada protocolo superior y de cada caso de comunicación e intercambio de información.
- FSC o “Frame Check Sequence”: consiste en una secuencia de verificación de los datos de la trama que se transmite. Mediante un cálculo cíclico redundante (cyclic redundancy check en inglés, CRC) se genera un valor de 32 bits (4 octetos) que permite que el receptor, realizando la misma operación sobre los datos, detecte posibles errores de transmisión. El procedimiento matemático completo aparece detallado en el apartado 3.2.9 del estándar IEEE 802.3.
- Extension: valores nulos al final de la trama para permitir la correcta sincronización del sistema.

5.4 Redes Sensibles en el Tiempo

Time Sensitive Networking (TSN) aparece con el objetivo de aportar a las capacidades determinísticas de Ethernet, nuevas capacidades de conexión y control de las comunicaciones. Estas soluciones surgen para soportar tráfico mixto de sectores muy variados, como el transporte (control de tráfico, redes automovilísticas, vehículos autónomos...), control de movimientos (turbinas eólicas, máquinas de impresión...), automatización de utilidades de potencia y comunicaciones de sistemas de control industriales.

Ofrece soluciones que proporcionan conectividad segura en internet, mientras se garantiza la correcta comunicación en redes que trabajan en tiempo real y

son críticas en el tiempo, utilizando una infraestructura de red física común. Se mantiene el determinismo, con la introducción de la capacidad de satisfacción de los requerimientos propios de redes con bajo volumen de tráfico y que utilizan un mismo medio compartido. El gran avance aportado por este estándar, es la capacidad de convergencia, permitiendo unificar en una misma red, tráfico de control crítico, tráfico de control no crítico y flujos de datos [18].

TSN define una serie de componentes y herramientas que deben estar presentes en las redes para permitir el correcto funcionamiento bajo las condiciones indicadas [2].

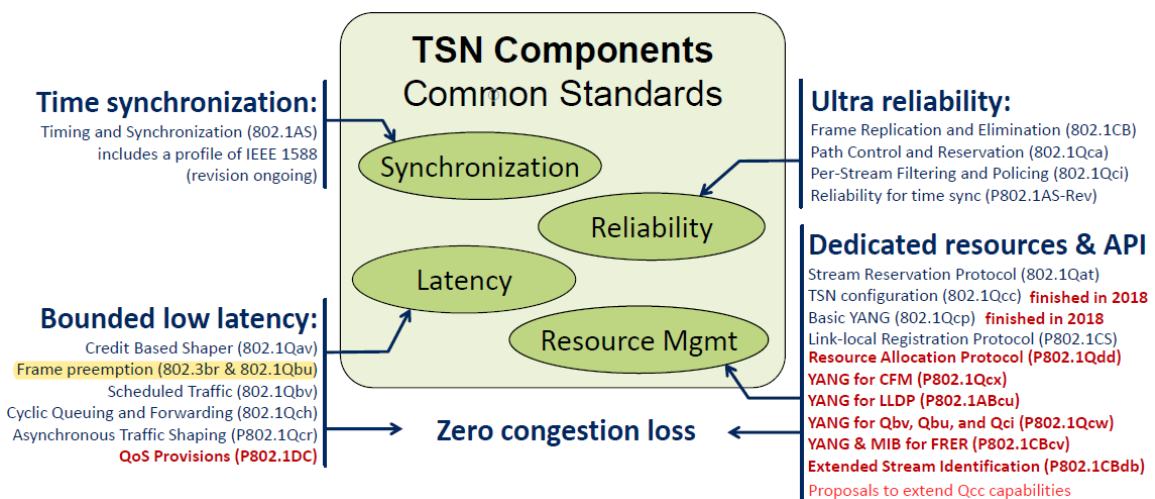


Figura 12: Componentes de los estándares desarrollados por los grupos de trabajo TSN [2]

Para asegurar el cumplimiento de estos aspectos, se elaboran una serie de estándares y normas que deben seguirse para el correcto desempeño de las funcionalidades.

Separando los componentes principales en los que se basa la estandarización de las redes TSN, se pueden diferenciar cuatro conceptos importantes:

- Sincronización: La sincronización en el tiempo de los diferentes equipos debe ser perfecta, a fin de evitar pérdidas de información y posibles funcionamientos no deseados. De esta funcionalidad en concreto, se encarga el estándar 802.1AS, basado en el protocolo de precisión temporal generalizado.
- Confiabilidad: La transmisión debe ser segura y robusta, para evitar pérdidas de datos, por fallos o interferencias. Se desarrollan para ello los estándares:
 - 802.1CB: Replicación de tramas para redundancia y Eliminación de tramas duplicadas.
 - 802.1Qca: Control de ruta y reserva del medio.
 - 802.1Qci: Filtrado y control de flujo.
 - 802.1AS-Rev: Sincronización temporal.
- Gestión de recursos: Es necesaria una correcta utilización de los recursos del medio. Para ello, se elaboran los estándares:
 - 802.1Qat: Protocolo de reserva de flujo.
 - P802.1Qcc: Mejoras del protocolo de reserva de flujo y mejoras de rendimiento.
 - P802.1Qcp: Modelo de datos YANG para configuración y reporte de estado de componentes.
 - P802.1CS: Protocolo de gestión para enlaces punto-a-punto.
- Latencia: Se debe asegurar una latencia máxima para la transmisión de información. Para ello se elaboran diferentes estándares:
 - 802.1Qav: Mejoras para el reenvío y puesta en cola en transmisiones sensibles al tiempo.
 - 802.1Qbu: Prioridades de tramas.
 - 802.3br: Intercalado de tráfico urgente. Complementa al estándar 802.1Qbu para redes basadas en el protocolo Ethernet 802.3. En

este estándar se basa el fundamento de este proyecto y se define en detalle en el siguiente apartado.

- 802.1Qbv: Mejoras para tráfico programado.
- 802.1Qch: Añade funcionalidades de reenvío y puesta en cola cíclicos.
- 802.1Qcr: Se encarga de la funcionalidad correspondiente con la organización de tráfico asíncrono.

Si se aseguran cada uno de estos conceptos, implementando las medidas pertinentes para ello (en muchos casos siendo necesario su desarrollo previo), se asegura la comunicación fiable y segura en redes sensibles al tiempo, manteniendo las características propias del protocolo de comunicación utilizado. Es un aporte extra sobre la funcionalidad básica de la comunicación. Se garantiza el transporte de la información, bajo una latencia máxima limitada y con pérdidas extremadamente bajas.

Se trata de una tecnología en auge, viable y productiva, aunque se encuentra en un estado inicial y requiere un largo desarrollo para estar completa. Por ello, surgen numerosos grupos de trabajo que se encargan de desarrollar los estándares y las funcionalidades particulares de cada aspecto sobre el que se basa la tecnología TSN [18]. También se han creado asociaciones de grandes empresas para potenciar estas metodologías de desarrollo, como es el caso de "Avnu Alliance" [1].

En el caso del IEEE, existe un grupo de trabajo denominado TSN TG (Time Sensitive Networking Task Group), que deriva del grupo conocido anteriormente como AVB (Audio Video Brinding) del estándar IEEE 802.1. Este grupo se dedica a elaborar las funcionalidades y los estándares directamente relacionados con la calidad del servicio en comunicaciones en redes sensibles al tiempo. Los últimos avances han generado nuevos estándares para el caso concreto de

ethernet, en el que se han incluido nuevos conceptos y características, como es el caso del estándar IEEE802.3br que permite el intercalado de tráfico prioritario y no prioritario, estableciendo una subcapa en el nivel de enlace del modelo OSI [5].

Para el caso concreto de las comunicaciones industriales, aparece el termino de OPC UA TSN, que es una fundación que lucha por la unificación de los estándares de comunicaciones industriales, unificando ethernet y TSN, para lograr los objetivos propios de la industria 4.0 [2].

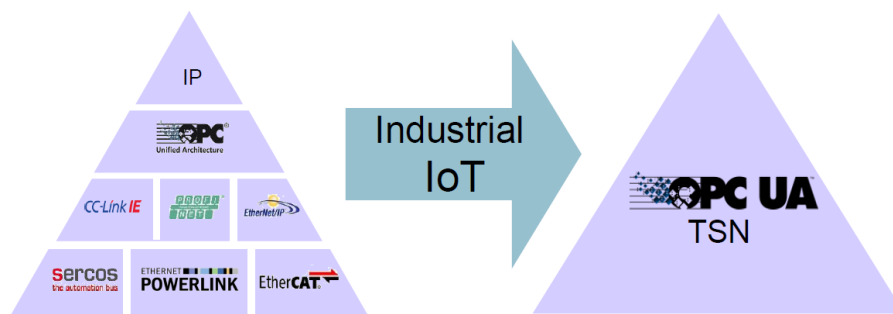


Figura 13: Tendencia de estandarización mediante la unificación de la red de comunicaciones industriales [6].

La tendencia es, por tanto, en todos los sectores afectados, la unificación y estandarización, buscando siempre un servicio adecuado y una interoperabilidad entre equipos.

5.5 Intercalado de tráfico prioritario y no prioritario

Se trata de una funcionalidad que surge ante la necesidad de garantizar una latencia mínima para un tipo de tráfico específico, en un ecosistema en el que convergen sobre el mismo medio de comunicación, tráfico procedente de naturalezas variadas, cada una de ellas presentando una prioridad diferente.

Este concepto en concreto, forma parte de lo que se conoce como Quality of Service (QoS o Calidad del Servicio) dentro de la solución TSN.

QoS trata de garantizar un servicio de comunicación adecuado y para ello engloba una serie de funcionalidades muy variadas. Para el caso concreto que nos concierne, el relativo a tiempo, retardos y latencia, pretende asegurar un tiempo máximo de comunicación entre dos equipos de la red.

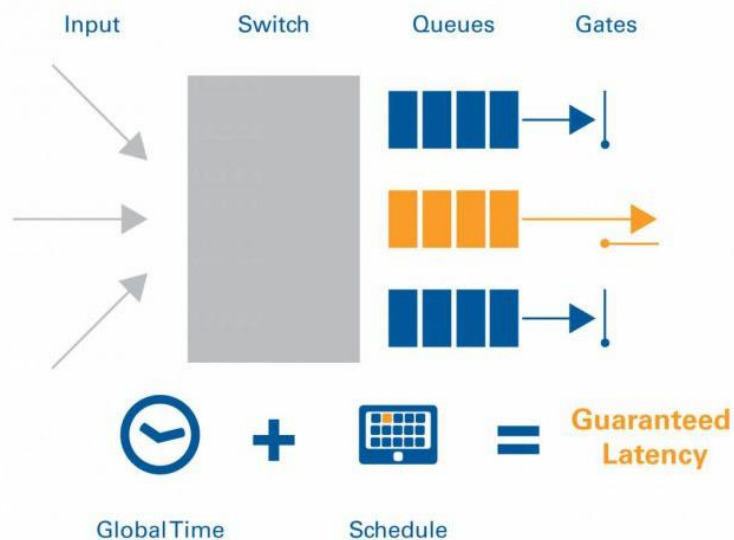


Figura 14: Tiempos y retardos propios de las comunicaciones [20]

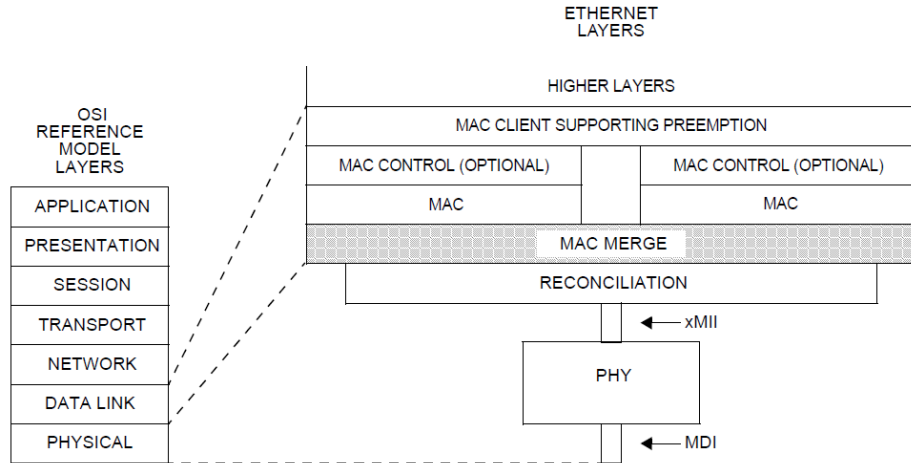
Como se puede apreciar en la Figura 14, la latencia que se debe asegurar, se encuentra condicionada por dos elementos:

- Tiempo global de procesamiento del sistema: procesamiento del emisor, conmutación de la información a través de la red, operación combinacional y secuencial de los elementos electrónicos de la red y encolado de la información en cada nodo de la red.

- Tiempo requerido para la transmisión y recepción de tráfico programado: retardos entre emisiones discretas en el tiempo que siguen una secuencia programada.

Teniendo en cuenta esta dependencia de la latencia con los tiempos definidos, el estándar IEEE 802.3br otorga los mecanismos y funcionalidades necesarios para asegurar que el impacto temporal del retardo debido a tráfico programado, se vea reducido, permitiendo una mejora en los tiempos de transmisión, así como una disminución de las bandas de guarda (tiempos de seguridad en los que no se emite información para evitar colisiones entre comunicaciones). De esta forma, se incrementa de forma simultánea la eficiencia de las comunicaciones, al reducir el tiempo en que se obliga a la red a permanecer en reposo. También permite la introducción en el medio de tráfico asíncrono, sin perjudicar al tráfico síncrono y cíclico.

Este estándar complementa al IEEE 802.1Qbu de TSN, para el caso concreto del modelo OSI, en el que se cuenta con un nivel de enlace que trabaja sobre el protocolo Ethernet. Para introducir la funcionalidad, define una capa intermedia entre el nivel de enlace MAC tradicional y el físico. Esta subcapa se encarga de procesar la información procedente de varios interfaces MAC, cada uno de ellos con una prioridad definida, estableciendo la secuencia necesaria de transmisión, de forma que, los paquetes con mayor prioridad, cumplan con las exigencias de retardos y latencias mínimas. Esta capa recibe el nombre de MAC Merge Sublayer (Subcapa de Unión MAC) [21].



NOTE—In this figure, the xMII is used as a generic term for the Media Independent Interfaces for implementations of 100 Mb/s and above. For example: for 100 Mb/s implementations this interface is called MII; for 1 Gb/s implementations it is called GMII; for 10 Gb/s implementations it is called XGMII; etc.

MAC = MEDIA ACCESS CONTROL
 xMII = MEDIA INDEPENDENT INTERFACE

MDI = MEDIUM DEPENDENT INTERFACE
 PHY = PHYSICAL LAYER DEVICE

Figura 15: Nivel de enlace del modelo OSI, implementando el estándar 802.3br de intercalado de tráfico con prioridad [21]

En la Figura 15, se representa la nueva estructura definida para el nivel de enlace del modelo OSI, teniendo en cuenta la subcapa MAC Merge, diferente a la mostrada anteriormente para el caso Ethernet tradicional, en la Figura 10. Se procesa el tipo de tráfico a generar en función de su procedencia y se elabora el paquete adecuado mediante el interfaz MAC apropiado. Mediante la subcapa de unión, se realizan las operaciones necesarias para enviar al mismo medio de transmisión, evitando colisiones e interferencias, los paquetes prioritarios y no prioritarios. Gracias a la introducción de la subcapa en el nivel de enlace, se abstrae al medio físico de la funcionalidad del intercalado de tráfico prioritario, facilitando la implementación de los protocolos propios de dicha capa del modelo OSI. Además, permite la continuidad de las tecnologías existentes en el nuevo ecosistema del nivel de enlace indicado (retrocompatibilidad).

Junto a la definición del concepto de MAC Merge Sublayer, aparecen una serie de términos importantes para la implementación de las nuevas características y su comprensión:

- Control de Acceso al Medio Express (eMAC): instancia de una subcapa MAC en la cual se procesa tráfico prioritario o urgente.
- Tráfico Express: tramas transmitidas por una eMAC, que presentan necesidad de latencia mínima o una prioridad máxima.
- Control de Acceso al Medio Preemptable (pMAC): instancia de una subcapa MAC en la cual se procesa tráfico interrumpible o no prioritario.
- Tráfico Preemptable o Normal: tramas transmitidas por una pMAC, que no presentan necesidad de latencia mínima sobre otras o prioridad máxima.

Por tanto, la subcapa MAC Merge recibe tráfico tanto de eMAC como de pMAC y realiza las operaciones necesarias para otorgar la prioridad a las tramas Express sobre las Preemptable.

Para ello, el estándar define una serie de modificaciones sobre los paquetes tradicionales con los que trabaja el protocolo Ethernet, representados anteriormente, en la Figura 11.

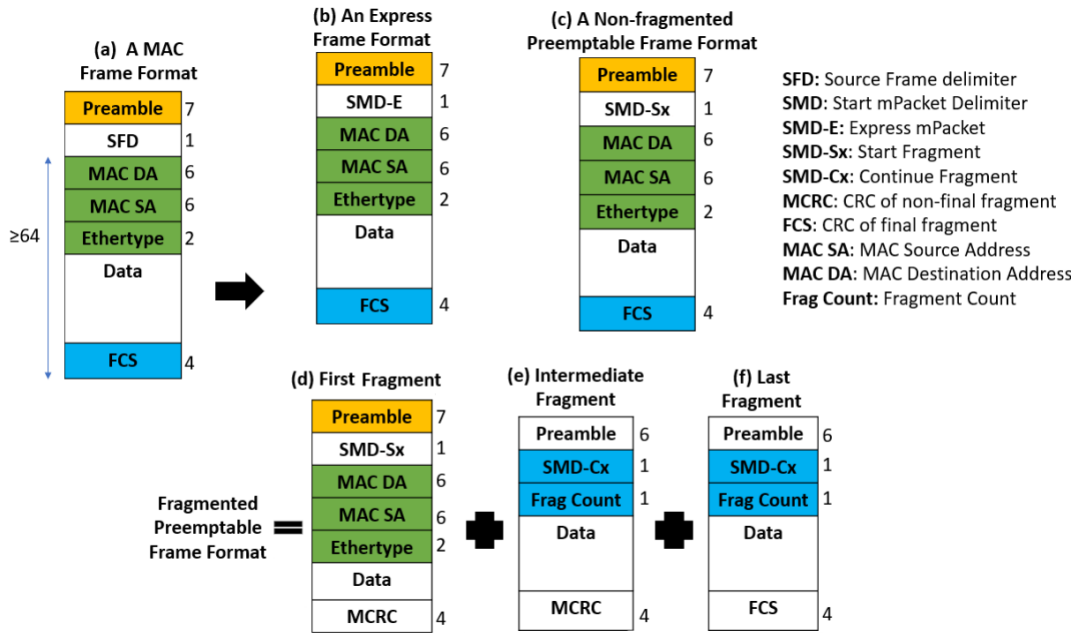


Figura 16: Estructura de tramas Ethernet bajo el estándar 802.3br y la funcionalidad Preemption [22]

Para cada uno de los casos expuestos en la Figura 16, se procede a explicar el contenido del paquete y la diferencia con el contenido tradicional:

- Campos comunes: Preamble, MAC DA, MAC SA y Ethertype presentan valores idénticos siempre que se trate de las mismas direcciones MAC destino y origen y el mismo tipo de protocolo de nivel utilizado, independientemente del tipo de paquete a generar.
- Trama MAC Tradicional (a): Mismo caso expuesto en el apartado correspondiente al protocolo Ethernet. Es el modelo de referencia y debe ser soportado para permitir la retrocompatibilidad.
- Trama eMAC (b): En este caso, el número de octetos con valor Preamble generados es de siete. El valor del tipo de paquete es SMD-E. El valor de FCS se corresponde con el CRC32 de ethernet para los datos comprendidos entre el octeto siguiente al SMD-E y el final de la carga útil de la trama (Data).

- Trama pMAC no interrumpida (c): El funcionamiento es similar al caso (b), con la diferencia de que el SMD-Sx se corresponde con el valor adecuado en función de la secuencia de tramas generadas.
- Trama pMAC interrumpida, fragmento inicial (d): El funcionamiento es similar al caso (c), con la diferencia de que el campo FCS se calcula realizando una operación XOR con el valor hexadecimal "0x0000FFFF" al CRC32 obtenido de los datos de la trama. De esta forma se identifica la interrupción de la trama, para su posterior reensamblado en destino.
- Trama pMAC interrumpida, fragmento intermedio (e): Este caso es el que más novedades introduce. El Preamble es generado durante seis octetos y se emite entonces el valor correspondiente SMD-Cx. Éste debe ser el correspondiente al valor de SMD-Sx de la trama inicial interrumpida. A continuación, para completar los ocho octetos de la cabecera del paquete, se introduce un octeto que contiene el valor adecuado del fragmento en cuestión. Este valor permite controlar errores para casos de hasta tres interrupciones para el mismo paquete. El valor mCRC se calcula de la misma forma que en el caso (d).
- Trama pMAC interrumpida, fragmento final (f): Similar al caso (e), con la diferencia de que el campo FCS se corresponde con el valor CRC32 típico de Ethernet, para el contenido de toda la trama emitida, es decir, la suma de los datos desde el fragmento inicial, hasta el fragmento final. Permite que, al reensamblar la trama en destino, se puedan detectar errores de transmisión producidos durante todo el proceso.

Los identificadores de tipo de trama indicados anteriormente, conocidos como SMD-XX (SMD-E, SMD-Sx y SMD-Cx), presentan unos valores preestablecidos por el estándar, permitiendo la asociación adecuada de fragmentos, así como la correcta decodificación de los datos internos de la trama. Estos valores se muestran en la Tabla 1. También aparecen valores adicionales, SMD-V y SMD-

R, que permiten la correcta sincronización de las capas MAC Merge de los equipos transmisor y receptor, a fin de conocer si el protocolo es soportado por ambos.

Para el caso de tramas pMAC interrumpidas correspondientes a fragmentos, tanto intermedios como finales, se definen aquellos valores posibles para el campo Frag_Count. Se muestran en la Tabla 2.

Tabla 1: Valores SMD de tramas Ethernet según estándar 802.3br [21]

mPacket type	Notation	Frame count	Value
verify packet	SMD-V	—	0x07
respond packet	SMD-R	—	0x19
express packet	SMD-E	—	0xD5
preemptable packet start	SMD-S0	0	0xE6
	SMD-S1	1	0x4C
	SMD-S2	2	0x7F
	SMD-S3	3	0xB3
continuation fragment	SMD-C0	0	0x61
	SMD-C1	1	0x52
	SMD-C2	2	0x9E
	SMD-C3	3	0x2A

Tabla 2: Valores FRAG_COUNT de tramas Ethernet según estándar 802.3br [21]

Frag_count	Value
0	0xE6
1	0x4C
2	0x7F
3	0xB3

Conociendo las características y los conceptos introducidos por el estándar, se puede concluir que las mejoras en el protocolo Ethernet son importantes, permitiendo su aplicación en redes con tráfico que no debe sufrir retardos por encima de un umbral, manteniendo una comunicación segura y robusta. Es por ello, que Ethernet es el protocolo de nivel de enlace preferido y utilizado por los diferentes grupos y equipos encargados del desarrollo de las funcionalidades TSN, permitiendo mediante la capa MAC Merge, la interoperabilidad entre protocolos de niveles superiores muy variados.

6. Análisis de alternativas

En este apartado, se analizan las diferentes alternativas del proyecto, determinando las razones y los beneficios de la selección final realizada.

Existen varios conceptos a tener en cuenta para la obtención de la selección final, alguno de ellos subordinado a los demás. Se analiza en detalle cada aspecto, identificando las diferentes posibilidades y obteniendo las conclusiones adecuadas para explicar la razón de la selección escogida.

6.1 Alternativas de diseño

Ante las diferentes posibilidades del diseño surge la primera duda sobre el desarrollo de un banco de pruebas para el módulo de transmisión de tramas ethernet bajo la funcionalidad preemption o de un banco de pruebas para el módulo de recepción del mismo protocolo. Existe la posibilidad de implementar ambos de forma simultánea, aunque la extensión del proyecto sería demasiado elevada. Es por ello que esta última opción es descartada.

Para la elección del sistema completo a desarrollar, módulo de transmisión o de recepción, se analizan diferentes aspectos.

6.1.1 Módulo de transmisión

Se trata del punto de mayor control dentro del flujo de intercambio de información y permite simular el sistema en un entorno controlado, pudiendo establecerse un flujo de información concreto. Esto permite un estudio más preciso e interesante de la propia funcionalidad y supone un elemento para probar, de forma directa, sistemas que deben soportar este protocolo.

6.1.2 Módulo de recepción

Se trata de un módulo subordinado al sistema que se desea probar y por tanto si éste falla o tiene un funcionamiento no esperado, el entorno de simulación deja de estar controlado y la prueba no da resultados tan precisos como en el caso de transmisión. Es por ello, que este sistema es descartado, ya que, no contando con un sistema de transmisión del banco de pruebas desarrollado, el entorno deja de ser controlado y los posibles errores no pueden ser identificados.

6.1.3 Selección de alternativa de diseño

Como se ha expuesto, el bloque más interesante y que supone el punto inicial para el desarrollo del banco de pruebas es el de transmisión, ya que permite controlar el entorno de simulación, definiendo los datos a transmitir. Además, es el punto inicial del flujo completo de prueba de un sistema determinado, ya que el módulo de recepción debe comparar los resultados obtenidos con un flujo de información inicial (aportado por el bloque de transmisión), generado en un entorno controlado, para identificar errores y características de interés producidas durante el intercambio de datos.

6.2 Alternativas de funcionalidades de bloques

Una vez se ha seleccionado la alternativa de diseño, es necesario analizar la complejidad y las características de los bloques que compondrán el sistema. Por tanto, es necesario definir la naturaleza de los mismos.

6.2.1 Módulos básicos controlados

Módulos creados estableciendo entornos controlados, mediante secuencias de ejecución preestablecidas. Los datos a generar serán controlados, obteniendo una secuencia esperada. Dichos datos pueden ser utilizados posteriormente

para determinar el funcionamiento correcto o incorrecto del sistema puesto a prueba. Se tiene control sobre la generación.

6.2.2 Módulos complejos pseudoaleatorios/aleatorios

Módulos creados para generar entornos de generación y simulación aleatorios o pseudoaleatorios. Permite estudiar posibles situaciones más reales que el entorno controlado, aunque el desarrollo inicial presenta mayor complejidad. Los datos aportados pueden ser más fiables a la realidad, aunque dificultan el estudio de la funcionalidad. Este tipo de funcionalidad depende de librerías de terceros que aporten la aleatoriedad, a fin de simplificar el desarrollo.

6.2.3 Selección de alternativa de funcionalidades de bloques.

Teniendo en cuenta la extensión del proyecto y las necesidades del mismo, se opta por elaborar módulos básicos para trabajar en entornos controlados. Esto se debe a que el desarrollo es más simple y eficiente en entornos en los que se pueda comprobar de forma directa el funcionamiento correcto del sistema. Además, se evita depender de librerías de terceros. De esta forma, se define un punto inicial del banco de pruebas con bloques que trabajan en entornos controlados, obteniendo pruebas con resultados previsibles y fácilmente comparables con los teóricos.

6.3 Alternativas de implementación

6.3.1 Implementación en sistema real

Una vez generado el sistema y el código correspondiente, se sintetiza y se introduce en un sistema electrónico real, analizando los resultados y comparándolos con los teóricos. A su vez, podrá diferenciarse dos casos de implementación destacados:

- Implementación en microprocesadores o microchips: permite una mejor adecuación a la aplicación particular del sistema, aunque en perjuicio de la versatilidad. El funcionamiento óptimo queda restringido al dispositivo para el que se desarrolla.
- Implementación en FPGA o matriz de puertas programables: permite una mayor versatilidad en cuanto a la aplicación del sistema. De cara a la implementación final, se debe optimizar para cada caso individual, pero su operabilidad global es mayor.

6.3.2 Implementación en entorno de simulación

Se genera el sistema completo y se simula en el entorno de desarrollo, obteniendo unos resultados teóricos para el funcionamiento del equipo.

6.3.3 Selección de alternativa de implementación

Finalmente, teniendo en cuenta la extensión del proyecto y el estado inicial del banco de pruebas, se opta por trabajar en el entorno de simulación. Esto permite estudiar mejor la funcionalidad del protocolo, permitiendo que los próximos avances se hagan sobre perspectivas y conocimientos más sólidos. Se elimina también la complejidad de adaptar el circuito a un elemento real, evitando posibles contratiempos por una configuración inadecuada.

Por otra parte, mediante el desarrollo de este tipo de implementación, se abre el camino hacia la implementación en FPGA, ya que se tiene la base sobre la que se sustenta dicha aplicación. Es decir, permite implementar en un futuro la funcionalidad en una FPGA adaptando el sistema a la electrónica concreta con la que cuenta el dispositivo.

7. Selección, descripción y diseño de la solución propuesta

Analizando el contexto industrial indicado, es necesario introducir nuevas funcionalidades que complementen a los estándares desarrollados hasta la fecha. Por ello, surge la idea de este proyecto, buscando solucionar los problemas que tienen lugar en redes sensibles al tiempo cuando se comunican, a través del mismo medio, equipos diferentes y bajo protocolos de comunicación variados.

Tras los últimos avances y tendencias, es inevitable seleccionar al protocolo de comunicación de ethernet como el medio de unificación del sistema, gracias a su papel en el panorama actual y futuro y su versatilidad en cuanto a funcionalidad, despliegue y coste.

Por tanto, la selección de la solución final consiste en la elaboración de un banco de pruebas para la funcionalidad preemption del estándar ethernet, codificado mediante lenguaje VHDL en el entorno de desarrollo/simulación Xilinx Vivado. Mediante la plataforma de desarrollo y el lenguaje utilizados, se puede realizar el proceso completo, desde la conceptualización hasta la implementación en un entorno de simulación controlado, así como en un posible circuito real, buscando la escalabilidad futura del sistema.

El banco de pruebas indicado consta de diferentes elementos y funcionalidades, los cuales se pasan a explicar de forma ordenada, desde la perspectiva más global del sistema, hasta las características internas del mismo.

La descripción detallada de cada uno de los elementos que conforman la solución, se encuentra en su respectivo apartado, en el Anexo A. Los diagramas de bloques completos de cada uno de los elementos del sistema y los diagramas de estados de las máquinas de estados que forman parte de estos bloques, se

encuentran en los Anexos B y C, respectivamente. El código propio de cada bloque, se especifica en el Anexo D.

7.1 Perspectiva global del sistema

Desde el punto de vista del sistema completo, se trata de un elemento que genera una secuencia de tramas características del protocolo Ethernet, teniendo en cuenta la estructura de los paquetes procedentes de eMAC y pMAC, para la funcionalidad del intercalado de tráfico prioritario.

Esta secuencia generada, se entrega mediante lo que se conoce como AXI Stream a un circuito determinado, externo al banco de pruebas, al que se desea someter a la prueba. Esta comunicación se basa en el intercambio de información a través de un bus de datos, en función de una señal de reloj proporcionada, teniendo en cuenta que tanto el emisor como el receptor deben estar preparados para ello. Esto se consigue mediante dos señales, una que indica que el receptor está listo para recibir datos, y otra que indica que los datos que el emisor está enviando son válidos.

De forma paralela, el banco de pruebas cuenta con un bloque interno que se encarga de exportar los datos de los paquetes emitidos, a un fichero de texto, procesable mediante un algoritmo desarrollado externamente. Dicho algoritmo, consiste en un script elaborado mediante lenguaje Python, que se encarga de leer el fichero de exportación de datos, y construir un fichero de captura de paquetes PCAP. Este fichero, es compatible con software de análisis de capturas de datos, como es el caso concreto utilizado de Wireshark. De esta forma, se puede observar y estudiar la naturaleza y las características de cada uno de los paquetes generados, conociendo el funcionamiento esperado del circuito testado y siendo capaces de encontrar posibles errores y verificar el correcto funcionamiento del mismo.

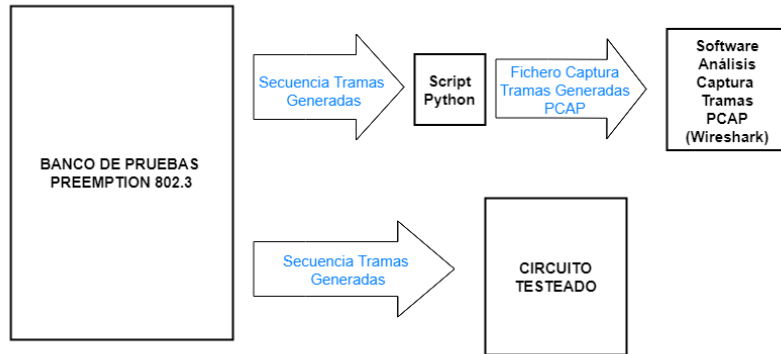


Figura 17: Diagrama conceptual del funcionamiento del sistema global

7.2 Banco de pruebas

Teniendo en cuenta las funcionalidades expuestas en el apartado del sistema global, se pueden diferenciar cuatro bloques internos, cada uno con una labor diferente:

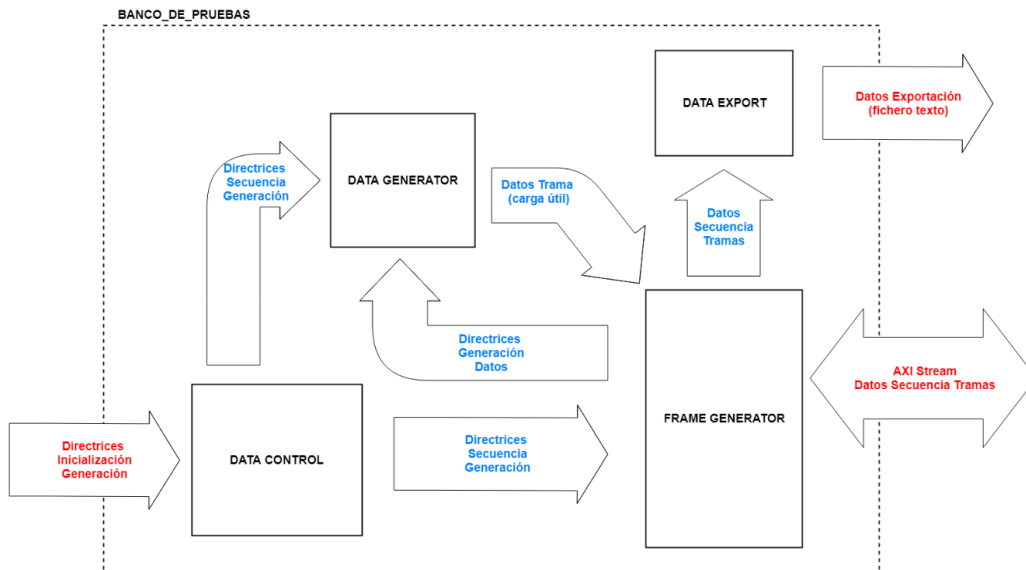


Figura 18: Diagrama conceptual del funcionamiento del bloque “Banco de Pruebas”

- Bloque de control de generación (DATA CONTROL): Bloque que se encarga de seguir la secuencia preestablecida para la prueba, emitiendo al resto de bloques las directrices necesarias para la generación de las tramas de salida, ensambladas de forma correcta.
- Bloque de generación (DATA GENERATOR): Se encarga de generar los bits que forman parte de la carga útil de las tramas. Recibe las directrices del bloque de control y del bloque ensamblador.
- Bloque ensamblador de tramas (FRAME GENERATOR): Se encarga de generar cada uno de los campos que forman las tramas que el bloque de control solicita, según la secuencia definida del entorno controlado. Genera la información que se transmite a través del AXI Stream hacia el circuito de prueba. Además, proporciona dicha información al bloque exportador.
- Bloque de exportación de tramas (DATA EXPORT): Recibe la secuencia de datos de las tramas generadas y elabora el fichero de exportación adecuado para el posterior tratamiento de la información.

7.3 Bloque de control de generación

Es necesario controlar la forma, el orden y la cadencia de generación de los datos que conformarán el contenido del campo de datos de las tramas que se enviarán mediante el sistema global. De esto se encarga este bloque, generando las señales adecuadas que serán enviadas al bloque generador, indicando cuando y como comenzar y controlando cuando y como terminar.

Es importante este control ya que, por una parte, trabajamos en un entorno en el que el sincronismo es necesario, para evitar pérdidas de datos en información. Por otra parte, se trata de un entorno sensible al tiempo, en el cual las respuestas deben ser de la forma más rápida y precisa posible, ante cualquier evento o variación.

Será inicializado y evolucionará a lo largo del tiempo siguiendo un flujo determinado por una máquina de estados, permitiendo que el control del sistema global sea completo.

Este bloque se encarga del control completo de la generación de datos (bits que irán encapsulados en el campo de información de las tramas). Internamente realizará la sucesión de estados necesarios para que así sea, permitiendo un sincronismo completo en el sistema, tanto en dar las directrices, como en realizar las operaciones propiamente dichas. De esta forma, se elabora un bloque que permitirá la flexibilidad adecuada (actuando como caja negra), ante las diferentes casuísticas que se quieran desarrollar en la generación de datos.

Requerirá por tanto un “reset” en el momento de inicio y aportar una señal de reloj síncrona con el resto del sistema. Una vez aportadas las condiciones anteriores, se deberá habilitar el bloque y seguirá el funcionamiento normal.

Internamente, por tanto, realiza la sucesión de estados necesarios para desarrollar el proceso completo de generación de paquetes de diferentes tipos y tamaños, permitiendo un sinfín de casuísticas posibles, únicamente variando ciertos aspectos de su configuración interna.

Proporciona al resto de bloques los tipos de paquete a generar “ePacket” o “pPacket” y el tamaño de los mismos. Deberá recibir mediante otro bloque las señales de finalización de generación de cada uno de ellos, para controlar de forma absoluta el flujo del proceso.

Este bloque general cuenta con dos bloques internos que realizan dos funciones separadas:

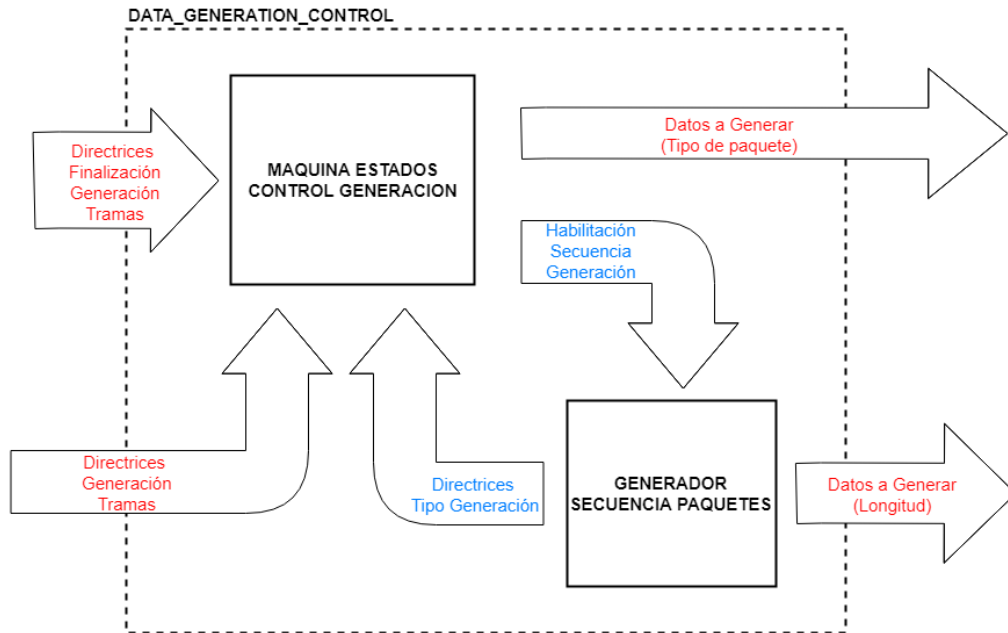


Figura 19: Diagrama conceptual del funcionamiento del bloque de control de generación

- **Máquina de Estados (MAQUINA ESTADOS CONTROL GENERACIÓN):**
 Este bloque se encarga de realizar la sucesión de los estados del sistema, en función de las entradas que recibe. Genera también las salidas necesarias para controlar los bloques que dependen de éste.

Proporciona por tanto las salidas referentes al tipo de paquete a generar, en caso de generar, y el habilitador para el otro elemento del bloque global de control.

- **Control tipo de paquete (GENERADOR SECUENCIA PAQUETES):**
 Recibe la señal de habilitación del bloque “Máquina de Estados” y se encarga de generar las señales necesarias para indicar a la máquina de estados el tipo de paquete a generar y la longitud de los mismos.

Este bloque será el que aporta el determinismo o la aleatoriedad. Partimos de un entorno determinista en el que los paquetes que se van a generar siguen una secuencia definida en el tiempo. La siguiente fase consistiría en la introducción de la pseudoaleatoriedad en la selección de la naturaleza y del tamaño de cada paquete.

7.4 Bloque de generación

Es necesaria la obtención de aquellos datos que conformarán el contenido del campo de datos de las tramas que se enviarán mediante el sistema global. De esto se encarga este bloque, siendo un simple subordinado de los bloques respectivos de control y de ensamblado de tramas, quienes solicitarán su trabajo mediante las señales adecuadas. Será inicializado y actuará como caja negra ante la obtención de los bits que formarán parte del campo de información de la trama ethernet correspondiente.

Este bloque, como se ha indicado anteriormente, se encarga de generar los datos. En función de una serie de parámetros aportados por el bloque de control de generación de datos del sistema, se realizan las operaciones adecuadas para la obtención de la totalidad de los datos que formarán parte del paquete de interés. De esta forma, se elabora un bloque que permitirá la flexibilidad adecuada (actuando como caja negra), ante las diferentes casuísticas que se quieran desarrollar mediante los diferentes tipos de control.

Solo será necesario controlar la inicialización del bloque de forma adecuada, en cada inicio del proceso de generación (reset y valor de tamaño de paquete inicial), aportar la señal de reloj síncrona con el sistema de control y habilitar el bloque cuando se tengan todos los datos en el estado adecuado.

Internamente, realiza los conteos adecuados para la generación de sub-paquetes del paquete total, y se generan las señales de validación y control respectivos a la generación de los datos.

Este bloque general cuenta con dos bloques internos que realizan dos funciones separadas:

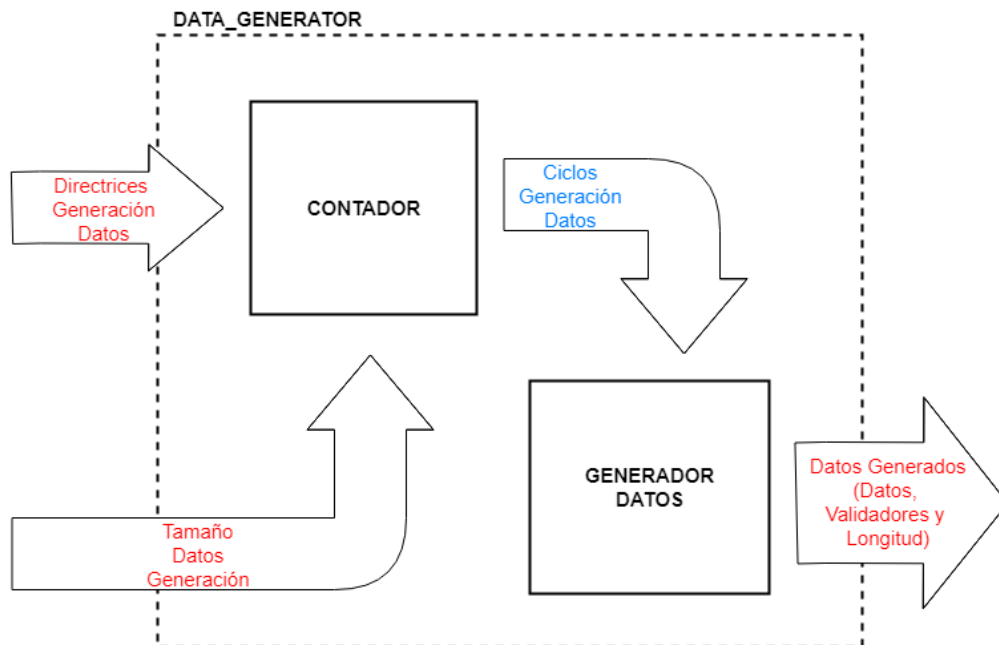


Figura 20: Diagrama conceptual del funcionamiento del bloque de generación

- Contador (CONTADOR): Este bloque se encarga de realizar los conteos descendientes adecuados para generar, en cada ciclo de reloj 1 byte de datos, es decir un octeto, hasta completar el total de bytes del paquete.

Proporciona la señal de habilitación y de finalización de paquete al bloque “Generador de datos”, hasta que llegue al fin de la cuenta. Además,

proporciona el tamaño del sub-paquete que se va a generar en el siguiente ciclo de reloj.

- Generador de datos (GENERADOR DATOS): Recibe las señales del bloque “Contador” y se encarga de generar los datos necesarios para el tamaño de paquete proporcionado. Aporta a la salida los datos del sub-paquete (1 byte), y las señales de validación para los siguientes bloques que indican que los datos están listos y en caso necesario que el paquete está finalizando.

Este bloque será el que aporta el determinismo o la aleatoriedad. Partimos de un entorno determinista en el que los sub-paquetes se rellenan con valores preestablecidos, siguiendo una secuencia cíclica. La siguiente fase consistiría en la introducción de la pseudoaleatoriedad en la generación de los bits que componen el paquete.

7.5 Bloque ensamblador de tramas

Es necesario generar la secuencia de bits que conformarán el contenido de las tramas que se enviarán mediante el sistema global, de forma adecuada. Esto permite el correcto encapsulado de la información y el establecimiento de los campos de control que permitirán el buen desempeño de la funcionalidad del protocolo de comunicación utilizado. De esto se encarga este bloque, siendo un simple subordinado del bloque de control, quien solicitará su trabajo mediante las señales adecuadas.

El bloque ensamblador proporciona las señales necesarias para generar cada uno de los campos de las tramas, así como las señales necesarias para la comunicación con el resto de bloques. Se encarga, por tanto, de la correcta coordinación del sistema, según las directrices obtenidas del bloque de control.

En función de una serie de parámetros aportados por el bloque de control de generación de datos del sistema, se realizan las operaciones adecuadas para la obtención de la totalidad de los datos que formarán parte de la trama de interés. De esta forma, se elabora un bloque que permitirá generar tramas ethernet a partir de un bloque de control y de un bloque generador de información. Así, se aíslan estos tres procesos, permitiendo desarrollar y mejorar cada uno de los bloques individualmente.

Solo será necesario controlar la inicialización del bloque de forma adecuada (cuando se desee comenzar con la generación de tramas), aportar la señal de reloj síncrona con el sistema de control y habilitar el bloque cuando se tengan todos los datos en el estado adecuado.

Internamente, realiza las sucesiones necesarias para generar las cabeceras y los datos de información de cada una de las tramas que el bloque de control solicita generar.

Este bloque general cuenta con seis bloques internos que realizan funciones separadas:

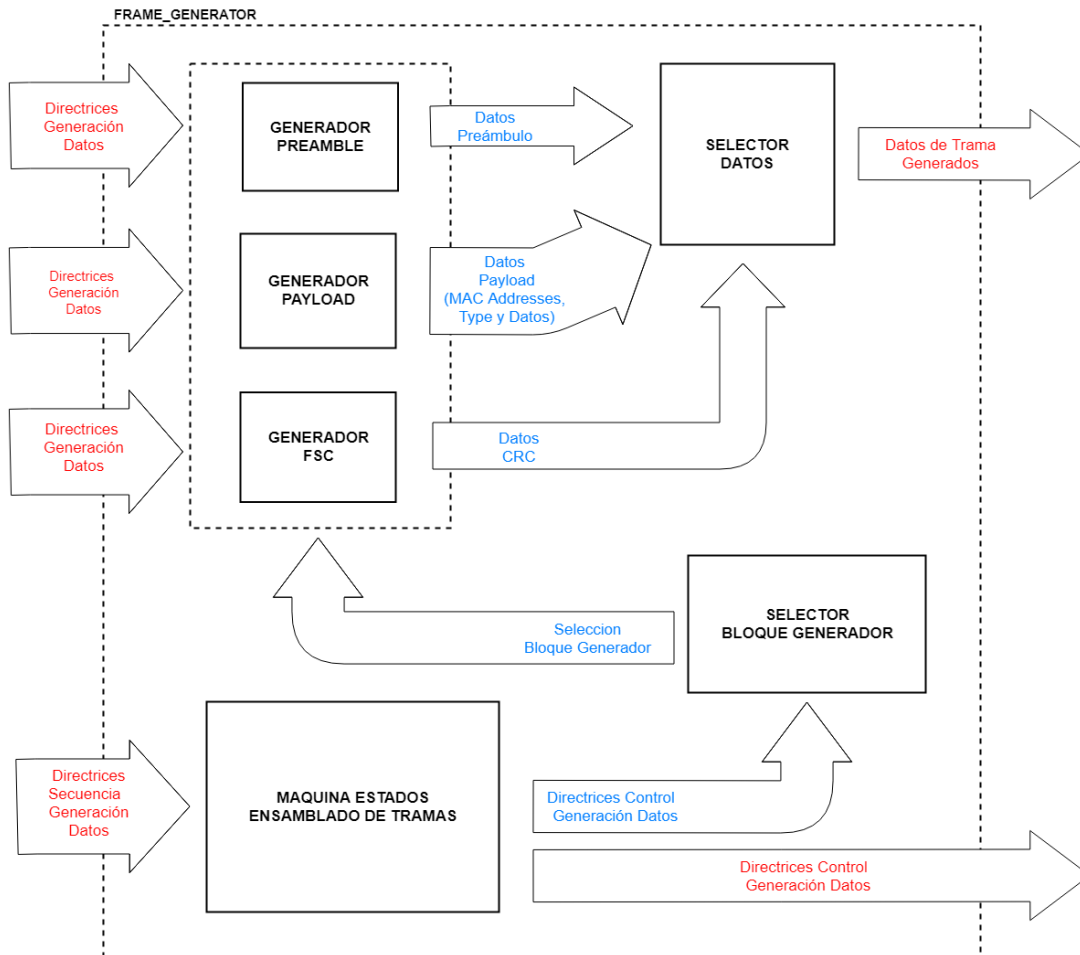


Figura 21: Diagrama conceptual del funcionamiento del bloque ensamblador de tramas

- **Preamble Generator (GENERADOR PREAMBLE):** Para que la trama sea identificada, el protocolo ethernet define una secuencia de bits que debe incluirse al comienzo de la trama como identificador. Así, se controla el inicio de la trama y se prepara al sistema para la transmisión y recepción correctas de la misma.

Este bloque se encarga de generar el preamblo adecuado para el tipo de paquete que se va a transmitir, en función de las señales de control recibidas del resto de bloques.

- Payload Generator (GENERADOR PAYLOAD): La trama debe tener un equipo origen y un equipo destino, de forma que la comunicación a través del protocolo de enlace utilizado se lleve a cabo de forma correcta.

Así mismo, la naturaleza de la trama es la de transmitir información entre estos equipos y, por tanto, esta información debe ser encapsulada en el interior de la misma.

Este bloque se encarga de gestionar estos dos procesos, realizándolos de forma adecuada y en el orden correcto.

- Frame Generator Control (MAQUINA ESTADOS ENSAMBLADO DE TRAMAS): Para que la generación de los diferentes campos de la trama se realice en el orden adecuado, obteniéndose cada uno de los datos en el momento apropiado, es necesario controlar de forma síncrona el proceso.

Para ello se genera este bloque. En su interior cuenta con una máquina de estados que, según las señales de control del bloque externo, comienza la secuencia de forma apropiada y avanza en el proceso, proporcionando las señales adecuadas para que la generación y el ensamblado de la trama sean correctos.

- FSC Generator (GENERADOR FSC): Para que la comunicación sea adecuada, es necesario que los datos recibidos sean idénticos a los enviados por el transmisor. Por ello, existe un campo en las tramas ethernet que se encarga de verificar la fiabilidad de la información

contenida en ella. Este campo es el FSC y se genera mediante este bloque.

Se encarga, por tanto, de realizar las operaciones necesarias sobre los datos generados por el bloque generador, para la obtención del código de verificación de la trama, diferenciando el valor del mismo, para cada uno de los posibles tipos de paquete.

- **Select Generating Block (SELECTOR BLOQUE GENERADOR):** Mediante las señales de control de la máquina de estados, se puede controlar el sistema de generación/ensamblado de tramas, aunque es necesario un bloque intermedio que comunique y sincronice estas señales de control, con cada uno de los bloques que generan los campos de las tramas.

Este bloque recibe las señales de control y genera aquellas señales necesarias en cada momento para la habilitación de cada uno de los bloques generadores de campos. También proporciona las señales necesarias para la selección de los datos propiamente dichos de estos bloques generadores.

- **Select Generated Data (SELECTOR DATOS):** En cada momento de la secuencia de generación, está generando datos un bloque en concreto. Es necesario que a través del bus que entrega los datos a los siguientes bloques, se obtenga de forma síncrona y correcta los datos procedentes del bus de salida de dicho bloque.

Mediante este bloque y las señales de control procedentes del bloque “Select Generating Block”, se realiza esta función.

7.6. Bloque exportador de datos generados

Para estudiar y analizar el sistema sometido a la prueba, es necesario conocer el funcionamiento que debería tener. Por ello, es necesario generar de forma paralela información de diagnóstico válida, para su comparación con los resultados finales obtenidos.

Este bloque se encarga de ello, recibiendo los datos totales generados, a partir del bloque ensamblador de tramas, y exportando los mismos a un fichero de texto plano, indicando los tiempos de transmisión, la longitud de los paquetes y los datos que forman cada paquete.

8. Descripción de tareas, fases, y procedimientos

Tabla 3: Descripción de tareas del proyecto

EDT	Nombre de tarea	Fecha inicio	Fecha fin	Duración
PT.1	Investigación de la tecnología	08/01/2019	30/01/2019	17
T.1.1	Analizar el contexto	08/01/2019	14/01/2019	5
T.1.2	Conocer el estado actual	15/01/2019	21/01/2019	5
T.1.3	Investigar los entornos de desarrollo	22/01/2019	24/01/2019	3
T.1.4	Detectar los riesgos	25/01/2019	28/01/2019	2
T.1.5	Identificar los beneficios de la solución	29/01/2019	30/01/2019	2
PT.2	Estudio de conceptos técnicos necesarios	31/01/2019	14/02/2019	11
T.2.1	Estudiar y comprender el lenguaje de programación utilizado	31/01/2019	05/02/2019	4
T.2.2	Conocer y familiarizarse con las funcionalidades del entorno de desarrollo	06/02/2019	07/02/2019	2
T.2.3	Consolidación de conceptos técnicos	08/02/2019	14/02/2019	5
H.1	Inicio del desarrollo conceptual de la solución	15/02/2019	15/02/2019	0
PT.3	Conceptualización de la solución	15/02/2019	13/03/2019	19
T.3.1	Elaboración del esquema global de la solución	15/02/2019	22/02/2019	6
T.3.2	Elaboración del esquema individual de los componentes internos de la solución	25/02/2019	06/03/2019	8
T.3.3	Desarrollo conceptual del proceso de comunicación entre los bloques internos de la solución	07/03/2019	13/03/2019	5
H.2	Finalización de conceptualización de la solución	14/03/2019	14/03/2019	0
H.3	Inicio del desarrollo funcional de la solución	14/03/2019	14/03/2019	0
H.4	Inicio de la redacción de la memoria	14/03/2019	14/03/2019	0

PT.4	Desarrollo de la solución	01/04/2019	27/05/2019	38
T.4.1	Desarrollo del bloque encargado de la generación de datos	01/04/2019	10/04/2019	8
T.4.2	Desarrollo del bloque encargado del control de la secuencia de generación	11/04/2019	26/04/2019	10
T.4.3	Desarrollo del bloque encargado del ensamblado de tramas	29/04/2019	20/05/2019	15
T.4.4	Construcción de la arquitectura global constituida por la interconexión de los bloques internos	21/05/2019	27/05/2019	5
PT.5	Simulación de solución desarrollada	28/05/2019	06/06/2019	8
T.5.1	Simulación del bloque encargado de la generación de datos	28/05/2019	29/05/2019	2
T.5.2	Simulación del bloque encargado del control de la secuencia de generación	30/05/2019	31/05/2019	2
T.5.3	Simulación del bloque encargado del ensamblado de tramas	03/06/2019	04/06/2019	2
T.5.4	Simulación de la arquitectura global constituida por la interconexión de los bloques internos	05/06/2019	06/06/2019	2
PT.6	Redacción	14/03/2019	25/06/2019	71
T.6.1	Introducción en la memoria de los conceptos teóricos obtenidos durante la investigación y el estudio	14/03/2019	18/03/2019	3
T.6.2	Elaboración de los diagramas propios de la conceptualización de la solución	19/03/2019	21/03/2019	3
T.6.3	Redacción de la ingeniería básica de la solución	22/03/2019	25/03/2019	2
T.6.4	Redacción de la ingeniería de detalle de la solución	26/03/2019	29/03/2019	4
T.6.5	Descripción y redacción de los resultados obtenidos de la implementación de la solución	17/06/2019	19/06/2019	3
T.6.6	Redacción de las conclusiones e ideas obtenidas de los resultados de la implementación de la solución	20/06/2019	21/06/2019	2
T.6.7	Adecuación de la memoria para respetar los aspectos de estilo y formato requeridos	24/06/2019	24/06/2019	1

T.6.8	Introducción y relación de las referencias utilizadas para la elaboración de la memoria	25/06/2019	25/06/2019	1
PT.7	Verificación de solución desarrollada	07/06/2019	14/06/2019	6
T.7.1	Detección y solución de errores	07/06/2019	13/06/2019	5
T.7.2	Simulación global para obtención de resultados	14/06/2019	14/06/2019	1
H.5	Finalización del desarrollo de la solución	17/06/2019	17/06/2019	0
H.6	Finalización de la redacción de la memoria	26/06/2019	26/06/2019	0
H.7	Finalización del proyecto	26/06/2019	26/06/2019	0

Se obtiene una duración total del proyecto de 118 días, teniendo en cuenta que se inicia el 8 de enero de 2019 y se finaliza el 26 de junio de 2019. El trabajo completo es realizado por una única persona.

Estableciendo una media de dos horas de trabajo por día planificado, se estima que la duración total del desarrollo del proyecto ha sido de unas 236 horas, aproximadamente.

9. Diagrama de Gantt/Cronograma

Para su mejor visualización, se incluye en el directorio “Diagramas” dentro del directorio “Ficheros”.

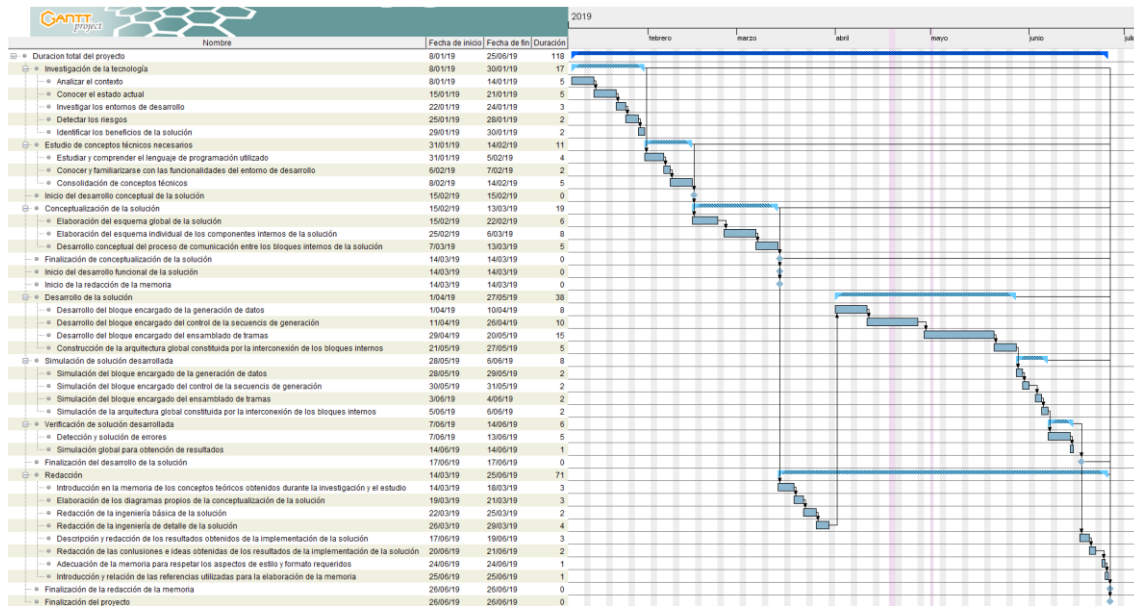


Figura 22: Diagrama de Gantt del proyecto

10. Análisis de costes

Se trata de un proyecto de investigación sin ánimo de lucro, cuyo único objetivo es educativo y de desarrollo tecnológico. Por ello, no requiere en principio de un análisis de los costes propios de su desarrollo. Los únicos gastos destacables, han sido aquellos procedentes de la adquisición de material de oficina. El equipo utilizado para realizar el proyecto fue adquirido con antelación por el alumno y las licencias de los programas utilizados eran propiedad del laboratorio APERT, por lo que no han supuesto coste adicional.

Sin embargo, es extrapolable a una situación particular de una empresa española, suponiendo que lo hubiera realizado por su cuenta, bajo una serie de condiciones.

En el caso de un único trabajador, suponiendo que es graduado en ingeniería técnica de telecomunicaciones, se estima que el coste horario es de unos 11€. Este valor se obtiene del salario bruto anual estimado para un recién graduado que se incorpora al mercado laboral, realizando las labores técnicas propias de su titulación (22000€/año, repartidos entre los 248 días laborables que tiene el año 2019, estimando una jornada de 8 horas diarias). Por tanto, las horas internas del proyecto se pueden observar en la Tabla 4.

En cuanto a los gastos, se pueden observar en la Tabla 5, siendo el único concepto el correspondiente al material de oficina (fotocopias, papel, bolígrafos...).

El apartado más interesante sería el de las amortizaciones, ya que aparecen tres conceptos diferenciados. El resultado de dicho cálculo se puede ver en la Tabla 6. Las suposiciones realizadas para este apartado son las siguientes:

- El ordenador portátil es adquirido para la realización del proyecto por 800€. La vida útil estimada para un equipo de estas características es de tres años, suponiendo un uso intensivo (para el cálculo, serán tres años, siendo utilizado en todo momento). Como se utiliza durante todo el proyecto, el uso del ordenador será de unas 236 horas. Calculando la tasa horaria y multiplicando por dicho número de horas, se obtiene su amortización.
- La licencia de Xilinx Vivado, se adquiere también para el proyecto, realizando la inversión de 2995€. La vida útil es de 12 meses, suponiendo que caduca al pasar dicha fecha. Suponiendo el total de horas de dicho periodo y calculando la tasa horaria, se puede obtener la amortización.
- Para la licencia de Microsoft Office, se realiza la compra por un valor de 170€ y tiene la misma vida útil que la licencia mencionada anteriormente. Aplicando estos valores y las horas de uso, se obtiene también la amortización.

Para el cálculo del total se utiliza la Tabla 7.

Tabla 4: Presupuesto. Horas internas

Horas Internas						
Concepto			N.º de recursos (Uds.)	Coste horario (€)	N.º de horas	Coste Total (€)
Ingeniero Técnico en						
Telecomunicaciones			1	11	236	2596
SUBTOTAL						2596

Tabla 5: Presupuesto. Gastos

Gastos			
Concepto	N.º de recursos (Uds.)	Coste unitario (€)	Coste Total (€)
Material de oficina	-	-	50
		SUBTOTAL	50

Tabla 6: Presupuesto. Amortizaciones

Amortizaciones					
Concepto	Coste adquisición (€)	Vida útil (horas)	Uso (horas)	Coste Total (€)	
Licencia Xilinx Vivado	2995	8760	92	31,4543379	
Licencia Microsoft Office	170	8760	38	0,737442922	
Ordenador Portátil	800	26280	236	7,184170472	
			SUBTOTAL	39,37595129	

Tabla 7: Presupuesto. Coste total del proyecto

Coste Total	
Horas Internas	2596
Amortizaciones	50
Gastos	39,37595129
Total	2685,375951

13. Descripción de los resultados

Teniendo en cuenta la solución descrita del sistema global, se define la siguiente secuencia de tramas a generar:

- Trama Express completa.
- Trama Preemptable completa.
- Trama Preemptable, que es interrumpida.
- Trama Express, que interrumpe a la anterior Preemptable.
- Trama Preemptable, que consiste en la continuación de la anterior Preemptable interrumpida.

La secuencia definida se puede observar de forma gráfica en la Figura 23.

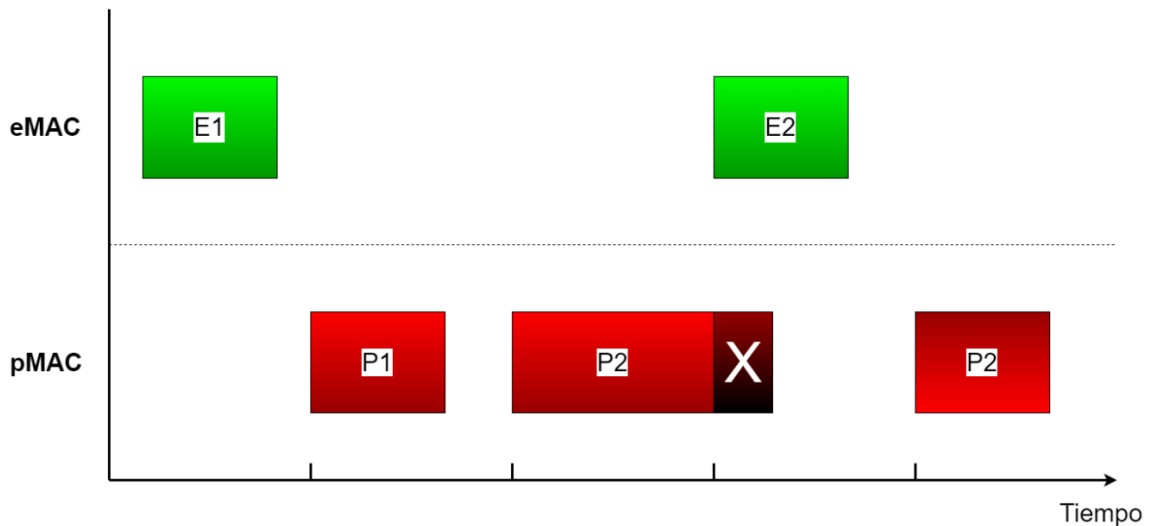


Figura 23: Secuencia de generación

Utilizando el entorno de simulación de Xilinx Vivado y estableciendo las señales necesarias en el testbench (o fichero de simulación) correspondiente del bloque Banco de Pruebas, se obtienen los datos propios del AXI Stream para la secuencia de tramas deseada. Es decir, los datos octeto a octeto, y la

confirmación de la validez de dichos datos. Estos resultados se muestran en la Figura 24.

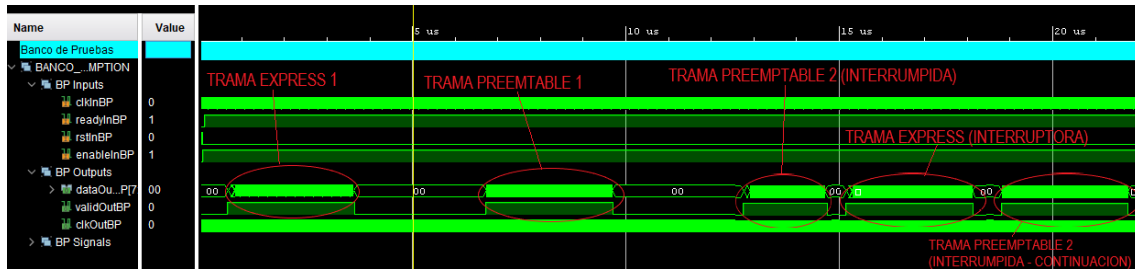


Figura 24: Secuencia generada por el Banco de Pruebas en el entorno de simulación

Si se analiza cada trama por separado, se puede observar el comportamiento de la funcionalidad con detalle, pudiendo diferenciarse los diferentes campos definidos por el estándar 802.3br para la estructura de los paquetes.

En la Figura 25, se puede observar el inicio de la primera trama, correspondiente al paquete Express completo. Primero aparece el valor “55” que se corresponde con el valor hexadecimal del Preamble, durante los siete octetos necesarios. A continuación, se genera el campo SMD-E (“0xD5”). A partir de ese momento, se generan los datos de la trama, incluyendo la dirección MAC Destino (“FF:FF:FF:FF:FF:FF”, valor correspondiente a difusión), la dirección MAC Origen (“AB:BC:CD:DE:EF:FA”, valor elegido para una mejor identificación del campo), valor Ethertype (“0x0800”, protocolo IPV4 de internet) y los datos propiamente dichos o Payload. Se genera el Payload hasta el final de trama, momento en el cual se genera el FSC, que consiste en el cálculo del CRC32 de los datos comprendidos desde la MAC Destino y el último octeto del Payload (ambos incluidos). Esta finalización de la trama se puede observar en la Figura 26.

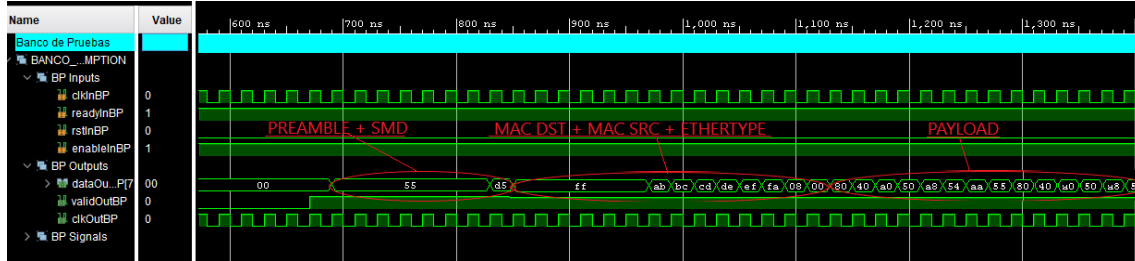


Figura 25: Inicio de primera trama generada por el Banco de Pruebas

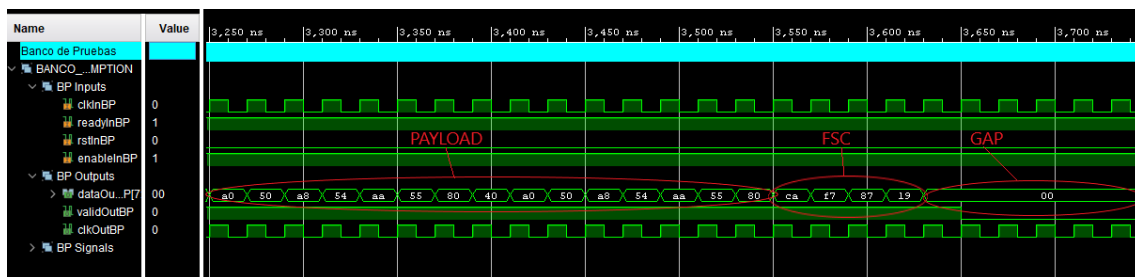


Figura 26: Final de primera trama generada por el Banco de Pruebas

Para el caso de la trama Preemptable completa, la estructura es la misma que en la situación anterior. La única variación es el valor que toma en el campo SMD, siendo éste “0x4c”, que indica que en el momento en que se genera, el contador de tramas Preemptable se encuentra en el número 1, como se puede observar en la Tabla 1 (SMD-S1). Los datos de esta trama se pueden observar en las Figuras 27 y 28.

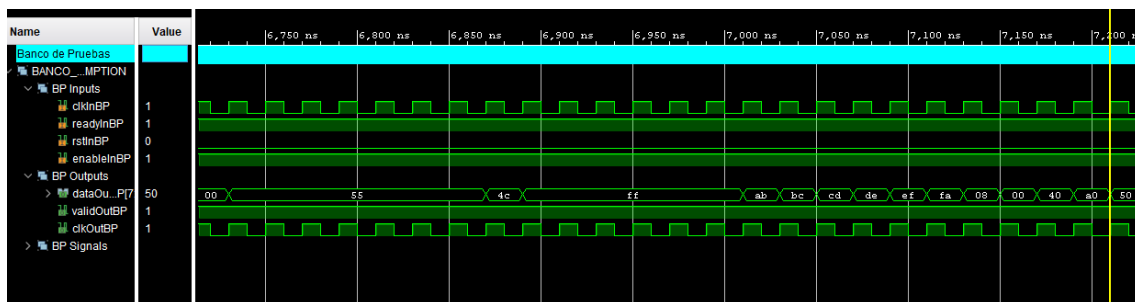


Figura 27: Inicio de segunda trama generada por el Banco de Pruebas

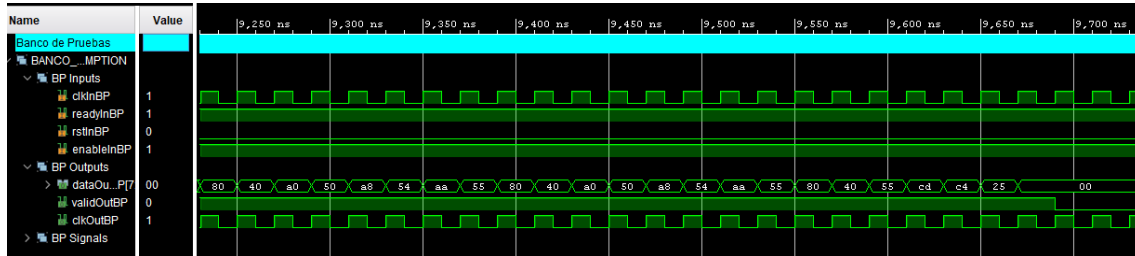


Figura 28: Final de segunda trama generada por el Banco de Pruebas

El tercer paquete generado, mostrado en las Figuras 29 y 30, muestra ciertas particularidades, ya que se corresponde con el paquete Preemptible que es interrumpido. Es el paquete de mayor interés de la secuencia definida, ya que demuestra la funcionalidad del intercalado de tráfico prioritario.

En el caso del inicio de la trama, el funcionamiento es el mismo que para el caso Preemptible completo. Ahora, el valor de SMD es "0x7F", el cual se corresponde con el siguiente valor del contador, como se puede observar en la Tabla 1 (SMD-S2). Sin embargo, al finalizar la trama, aparece una característica interesante, ya que, al finalizar la emisión de Payload, no se emite el FSC, sino el valor mCRC. Este valor se obtiene mediante la operación indicada en el estándar 802.3br e indica que el paquete ha sido interrumpido.

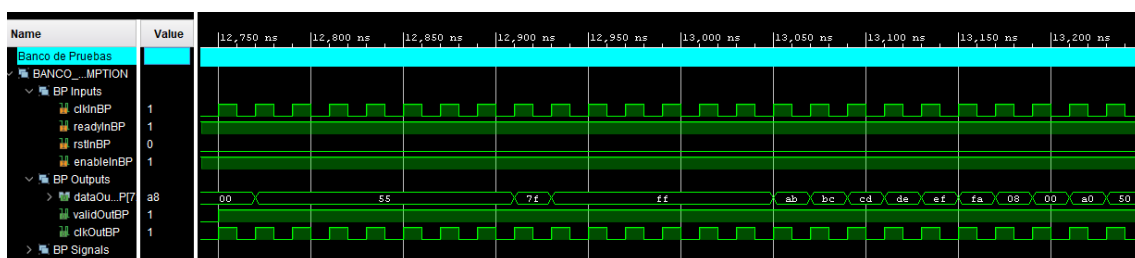


Figura 29: Inicio de tercera trama generada por el Banco de Pruebas

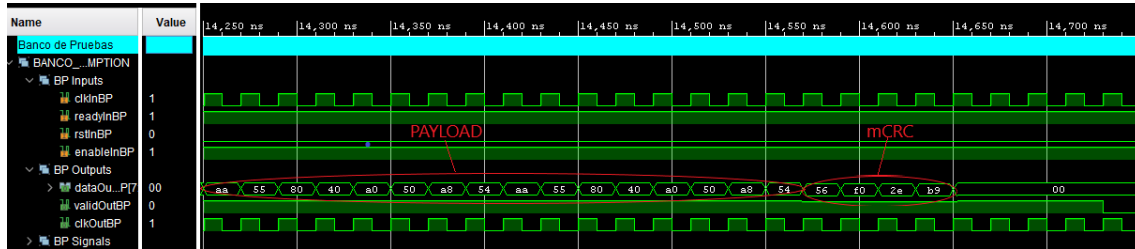


Figura 30: Final de tercera trama generada por el Banco de Pruebas

Una vez es interrumpido el paquete Preemptable, se emite el paquete Express que lo interrumpe sin ninguna particularidad nuevas. Mismo caso que primer paquete de la secuencia. Comportamiento de esta trama en las figuras 31 y 32

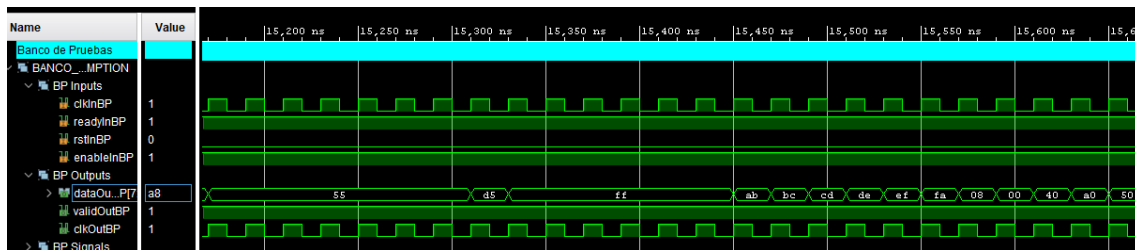


Figura 31: Inicio de cuarta trama generada por el Banco de Pruebas

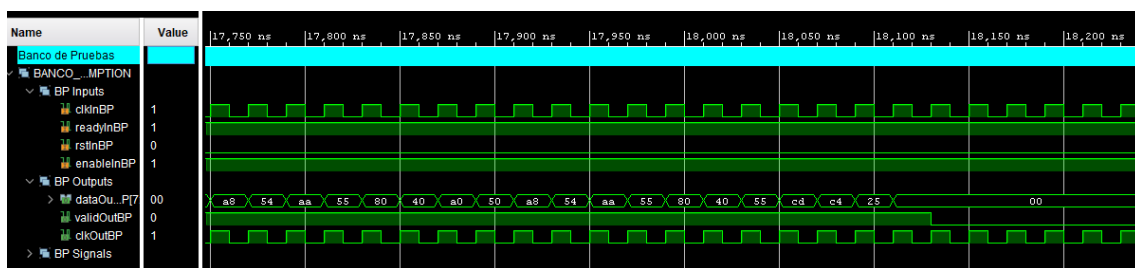


Figura 32: Final de cuarta trama generada por el Banco de Pruebas

Cuando finaliza el paquete Express interruptor, se continúa con la generación de los datos que conforman la trama correspondiente al paquete Preemptable anterior interrumpido. La generación del paquete de continuación se puede observar en las Figuras 33 y 34.

En este caso, se puede observar que el valor de SMD (“0x9E”), se corresponde con el valor del fragmento de continuación de trama equivalente al contador con valor dos, como se puede observar en la Tabla 1 (SMD-C2). Esto indica al receptor que ambos paquetes forman parte de la misma trama y, por tanto, que debe reensamblarlas cuando las reciba. Para el campo de FRAG_COUNT, se tiene un valor de “0xE6”, correspondiente al primer valor del número de fragmentos indicado en la Tabla 2.

Una vez finalizada la generación de la trama, se introduce el valor del FSC. Este valor se obtiene mediante el cálculo del CRC32 de Ethernet, para todos los datos, de todos los fragmentos que conforman la trama completa. Los campos cuyos datos son utilizados para el cálculo el FSC se muestran en la Figura 16 y se corresponden con los datos a partir del octavo octeto de los paquetes (sin tener en cuenta el preámbulo).

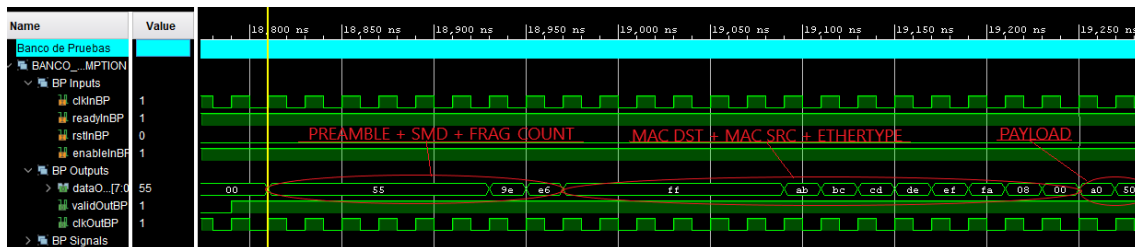


Figura 33: Inicio de quinta trama generada por el Banco de Pruebas

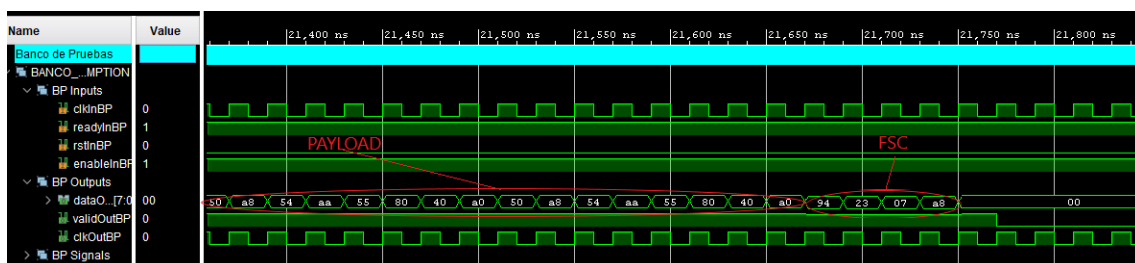


Figura 34: Final de quinta trama generada por el Banco de Pruebas

De esta forma, se obtiene la secuencia deseada y se observa el correcto funcionamiento del protocolo, permitiendo demostrar que la funcionalidad es válida e introduce importantes beneficios a la comunicación. Se pueden detectar interrupciones, asegurando que el tráfico prioritario llega con latencia mínima y que el tráfico interrumpido, puede ser reensamblado en destino, no perdiendo datos en el proceso.

Una vez se han generado los datos de la secuencia de tramas, se exportan a un fichero de texto plano, en el cual se introducen los datos de cada paquete, incluyendo marca de tiempo, longitud de paquete y datos de paquete. Se utiliza el valor decimal de los datos para su correcto procesado. El contenido del fichero de texto plano, para los datos de las tramas de la secuencia definida en este mismo apartado, se observa en la Figura 35.

```

1 3 148 85 85 85 85 85 85 85 213 255 255 255 255 255 171 188 205 222 239 250 8 0 128 64
160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80
168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84
170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85
128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64
160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 202 247 135 25

2 9 148 85 85 85 85 85 85 85 76 255 255 255 255 255 171 188 205 222 239 250 8 0 64 160 80
168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84
170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85
128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64
160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80
168 84 170 85 128 64 160 80 168 84 170 85 128 64 85 205 196 37

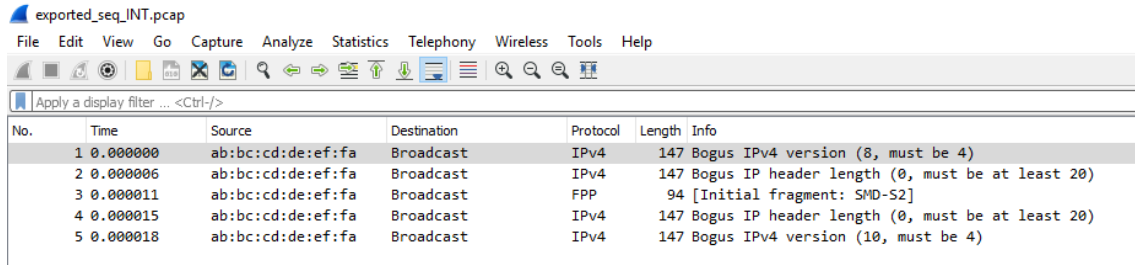
3 14 98 85 85 85 85 85 85 85 127 255 255 255 255 255 171 188 205 222 239 250 8 0 160 80
168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84
170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85
128 64 160 80 168 84 170 85 128 64 160 80 168 84 86 240 46 185

4 18 148 85 85 85 85 85 85 85 213 255 255 255 255 255 171 188 205 222 239 250 8 0 64 160
80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168
84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170
85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128
64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160
80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 85 205 196 37

5 21 148 85 85 85 85 85 85 158 230 255 255 255 255 255 171 188 205 222 239 250 8 0 160 80
168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84
170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85
128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64
160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 80
168 84 170 85 128 64 160 80 168 84 170 85 128 64 160 148 35 7 168
  
```

Figura 35: Datos de secuencia generada exportados a fichero de texto plano

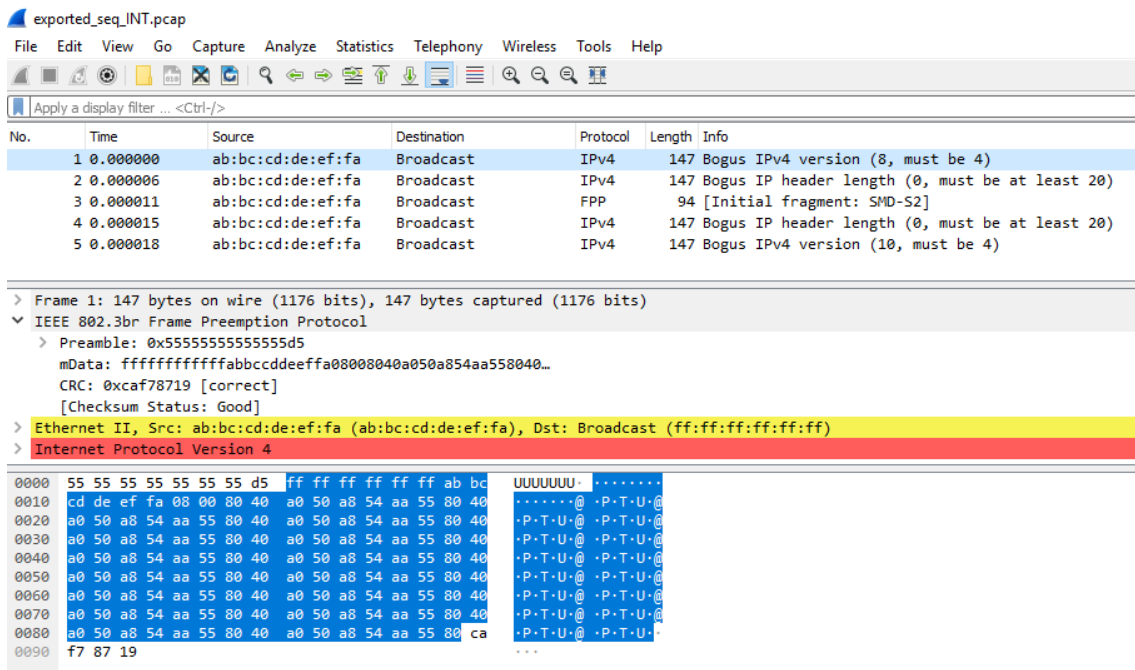
Mediante el script en Python, se extraen los datos del fichero de texto y se genera un nuevo fichero de captura de datos en formato pcap. Se puede observar el contenido del mismo en la Figura 36, utilizando el programa Wireshark. Se puede entonces analizar en detalle los valores de los campos de cada paquete.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (8, must be 4)
2	0.000006	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
3	0.000011	ab:bc:cd:de:ef:fa	Broadcast	FPP	94	[Initial fragment: SMD-S2]
4	0.000015	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
5	0.000018	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (10, must be 4)

Figura 36: Datos de secuencia generada exportados a fichero de capturas PCAP

Se puede observar que los campos se corresponden con los esperados tras la simulación. El programa Wireshark verifica que los campos son correctos y permite validar el proceso de desarrollo y simulación llevado a cabo. Los detalles de cada paquete se observan en las Figuras 37, 38, 39, 40 y 41, siguiendo la secuencia de generación definida al principio del apartado.



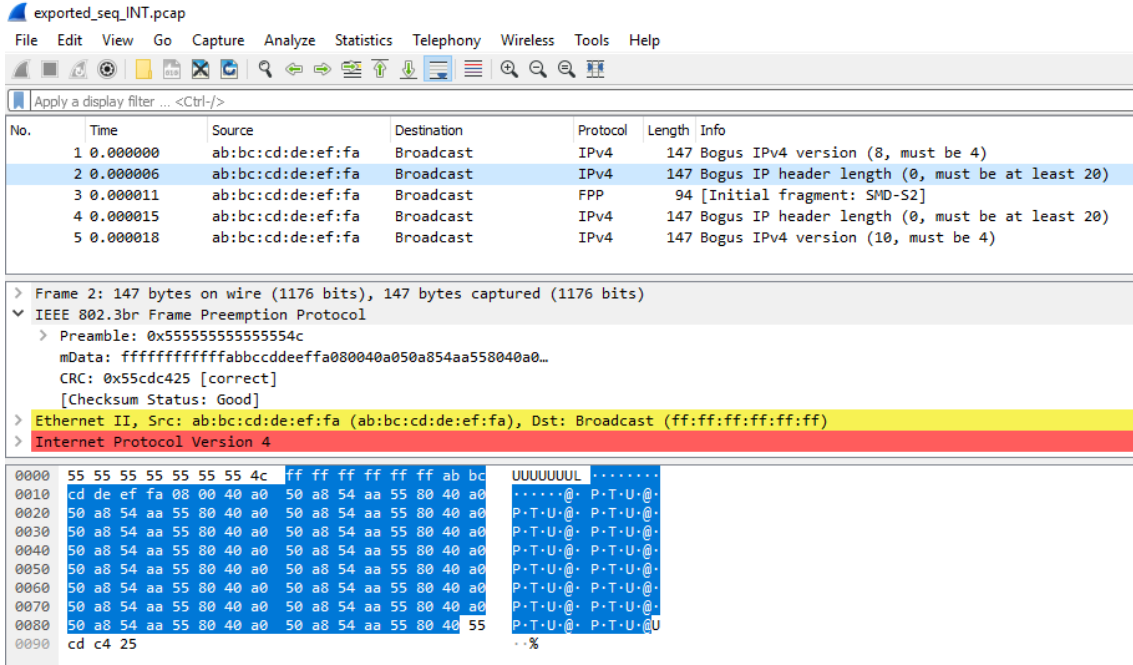
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (8, must be 4)
2	0.000006	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
3	0.000011	ab:bc:cd:de:ef:fa	Broadcast	FPP	94	[Initial fragment: SMD-S2]
4	0.000015	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
5	0.000018	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (10, must be 4)


```

> Frame 1: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits)
  > IEEE 802.3br Frame Preemption Protocol
    > Preamble: 0x55555555555555d5
      mData: ffffffffabccddeeffa0800804a050a854aa558040...
      CRC: 0xcaf78719 [correct]
      [Checksum Status: Good]
    > Ethernet II, Src: ab:bc:cd:de:ef:fa (ab:bc:cd:de:ef:fa), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    > Internet Protocol Version 4
  
```

0000	55 55 55 55 55 55 d5	ff ff ff ff ff ab bc	UUUUUUU ······
0010	cd de ef fa 08 00 80 40	a0 50 a8 54 aa 55 80 40	·····@ ·P·T·U·@
0020	a0 50 a8 54 aa 55 80 40	a0 50 a8 54 aa 55 80 40	·P·T·U·@ ·P·T·U·@
0030	a0 50 a8 54 aa 55 80 40	a0 50 a8 54 aa 55 80 40	·P·T·U·@ ·P·T·U·@
0040	a0 50 a8 54 aa 55 80 40	a0 50 a8 54 aa 55 80 40	·P·T·U·@ ·P·T·U·@
0050	a0 50 a8 54 aa 55 80 40	a0 50 a8 54 aa 55 80 40	·P·T·U·@ ·P·T·U·@
0060	a0 50 a8 54 aa 55 80 40	a0 50 a8 54 aa 55 80 40	·P·T·U·@ ·P·T·U·@
0070	a0 50 a8 54 aa 55 80 40	a0 50 a8 54 aa 55 80 40	·P·T·U·@ ·P·T·U·@
0080	a0 50 a8 54 aa 55 80 40	a0 50 a8 54 aa 55 80 ca	·P·T·U·@ ·P·T·U·
0090	f7 87 19		···

Figura 37: Detalles de primera trama obtenidos mediante Wireshark



exported_seq_INT.pcap
 File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

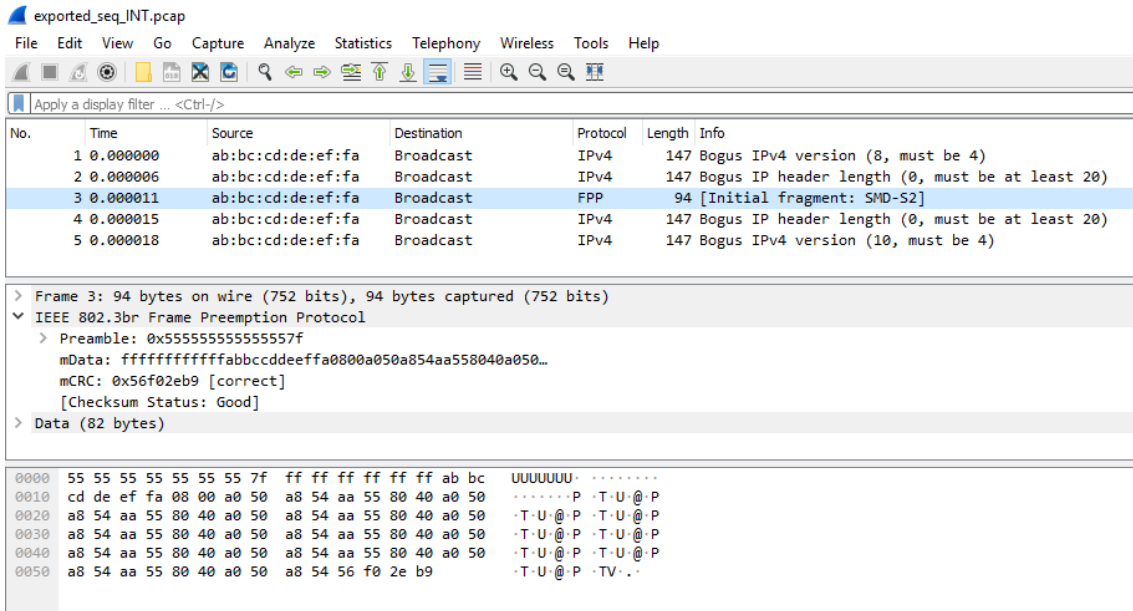
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (8, must be 4)
2	0.000006	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
3	0.000011	ab:bc:cd:de:ef:fa	Broadcast	FPP	94	[Initial fragment: SMD-S2]
4	0.000015	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
5	0.000018	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (10, must be 4)

> Frame 2: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits)
 IEEE 802.3br Frame Preemption Protocol
 > Preamble: 0x555555555555554c
 mData: ffffffffabccddeeffa080040a050a854aa558040a0...
 CRC: 0x55cdc425 [correct]
 [Checksum Status: Good]
 > Ethernet II, Src: ab:bc:cd:de:ef:fa (ab:bc:cd:de:ef:fa), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Internet Protocol Version 4

```

0000 55 55 55 55 55 55 55 4c ff ff ff ff ff ab bc UUUUUUL .....
0010 cd de ef fa 08 00 40 a0 50 a8 54 aa 55 80 40 a0 .....@ P.T.U.@
0020 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@ P.T.U.@
0030 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@ P.T.U.@
0040 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@ P.T.U.@
0050 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@ P.T.U.@
0060 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@ P.T.U.@
0070 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@ P.T.U.@
0080 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 55 P.T.U.@ P.T.U.@
0090 cd c4 25 ..%
  
```

Figura 38: Detalles de segunda trama obtenidos mediante Wireshark



exported_seq_INT.pcap
 File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

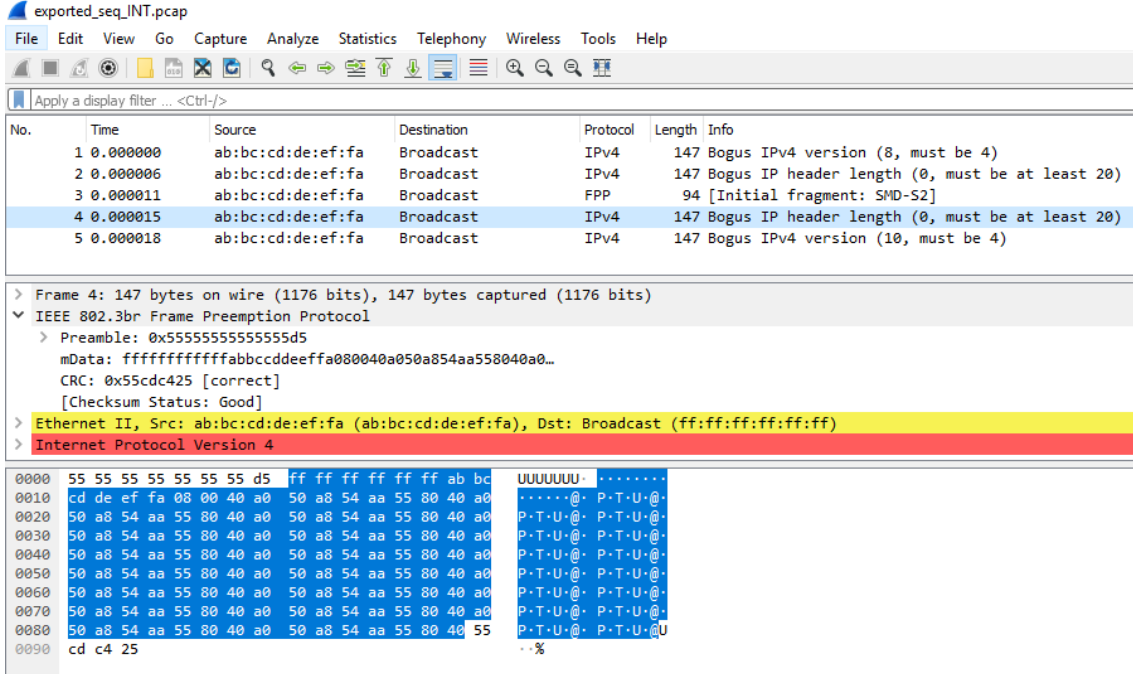
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (8, must be 4)
2	0.000006	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
3	0.000011	ab:bc:cd:de:ef:fa	Broadcast	FPP	94	[Initial fragment: SMD-S2]
4	0.000015	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
5	0.000018	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (10, must be 4)

> Frame 3: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
 IEEE 802.3br Frame Preemption Protocol
 > Preamble: 0x555555555555557f
 mData: ffffffffabccddeeffa0800a050a854aa558040a050...
 mCRC: 0x56f02eb9 [correct]
 [Checksum Status: Good]
 > Data (82 bytes)

```

0000 55 55 55 55 55 55 55 7f ff ff ff ff ff ab bc UUUUUUL .....
0010 cd de ef fa 08 00 a0 50 a8 54 aa 55 80 40 a0 50 .....@ P.T.U.@P
0020 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@P .T.U.@P
0030 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@P .T.U.@P
0040 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@P .T.U.@P
0050 a8 54 aa 55 80 40 a0 50 a8 54 56 f0 2e b9 .T.U.@P .TV..
  
```

Figura 39: Detalles de tercera trama obtenidos mediante Wireshark



exported_seq_INT.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (8, must be 4)
2	0.000006	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
3	0.000011	ab:bc:cd:de:ef:fa	Broadcast	FPP	94	[Initial fragment: SMD-S2]
4	0.000015	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
5	0.000018	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (10, must be 4)

> Frame 4: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits)

IEEE 802.3br Frame Preemption Protocol

- Preamble: 0x555555555555d5
- mData: ffffffffabccddeeffa080040a050a854aa558040a0...
- CRC: 0x55cdc425 [correct]
- [Checksum Status: Good]

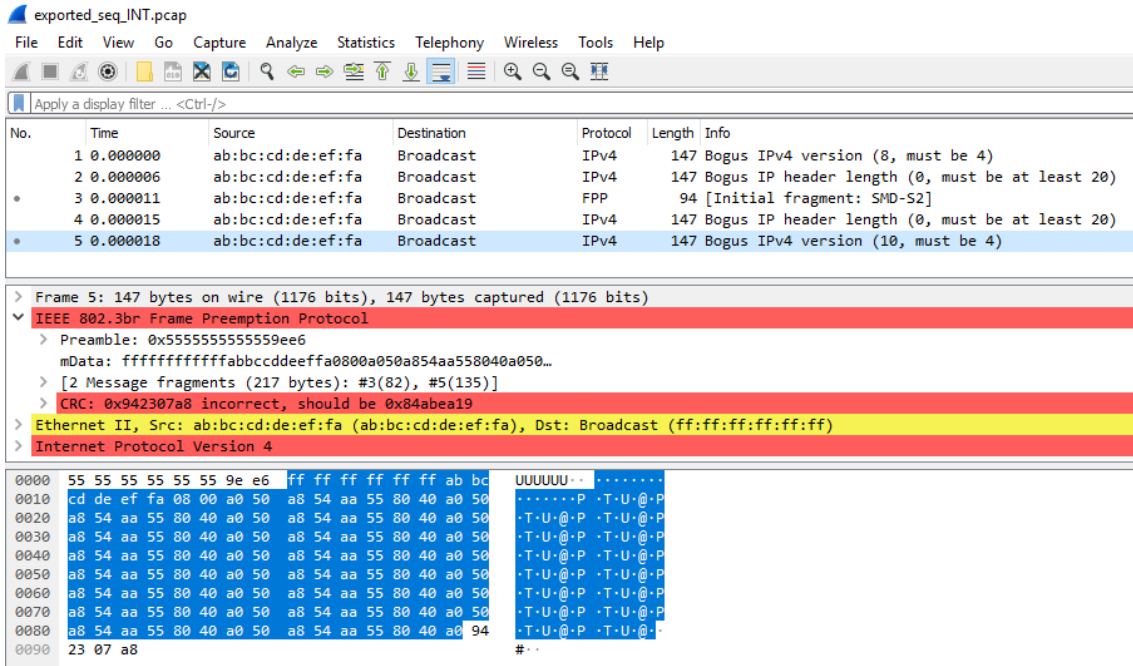
> Ethernet II, Src: ab:bc:cd:de:ef:fa (ab:bc:cd:de:ef:fa), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Internet Protocol Version 4

```

0000 55 55 55 55 55 55 55 d5 ff ff ff ff ff ab bc UUUUUU . . . . .
0010 cd de ef fa 08 00 40 a0 50 a8 54 aa 55 80 40 a0 . . . . .@. P.T.U.@.
0020 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@. P.T.U.@.
0030 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@. P.T.U.@.
0040 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@. P.T.U.@.
0050 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@. P.T.U.@.
0060 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@. P.T.U.@.
0070 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 P.T.U.@. P.T.U.@.
0080 50 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 55 P.T.U.@. P.T.U.@.U
0090 cd c4 25 ..%
  
```

Figura 40: Detalles de cuarta trama obtenidos mediante Wireshark



exported_seq_INT.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (8, must be 4)
2	0.000006	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
3	0.000011	ab:bc:cd:de:ef:fa	Broadcast	FPP	94	[Initial fragment: SMD-S2]
4	0.000015	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IP header length (0, must be at least 20)
5	0.000018	ab:bc:cd:de:ef:fa	Broadcast	IPv4	147	Bogus IPv4 version (10, must be 4)

> Frame 5: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits)

IEEE 802.3br Frame Preemption Protocol

- Preamble: 0x5555555555559ee6
- mData: ffffffffabccddeeffa0800a050a854aa558040a050...
- [2 Message fragments (217 bytes): #3(82), #5(135)]
- CRC: 0x942307a8 incorrect, should be 0x84abea19

> Ethernet II, Src: ab:bc:cd:de:ef:fa (ab:bc:cd:de:ef:fa), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Internet Protocol Version 4

```

0000 55 55 55 55 55 55 9e e6 ff ff ff ff ff ab bc UUUUUU . . . . .
0010 cd de ef fa 08 00 a0 50 a8 54 aa 55 80 40 a0 50 . . . . .P.T.U.@.P
0020 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@.P .T.U.@.P
0030 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@.P .T.U.@.P
0040 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@.P .T.U.@.P
0050 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@.P .T.U.@.P
0060 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@.P .T.U.@.P
0070 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 50 .T.U.@.P .T.U.@.P
0080 a8 54 aa 55 80 40 a0 50 a8 54 aa 55 80 40 a0 94 .T.U.@.P .T.U.@.P
0090 23 07 a8 #. .
  
```

Figura 41: Detalles de quinta trama obtenidos mediante Wireshark

Como detalle interesante, en el caso del último paquete, el propio programa Wireshark relaciona los paquetes “inicial” y “de continuación”, reensamblado la trama completa como se observa en la Figura 41. En este caso, el CRC calculado no es válido ya que no se ha implementado en la solución y queda pendiente para versiones futuras.

Gracias a la elaboración del sistema completo, junto a la exportación de los datos, se logra un ecosistema de simulación idóneo para la naturaleza del “Banco de Pruebas”. Permite poner a prueba circuitos de interés, definiendo en la totalidad el entorno en el que se va a trabajar.

14. Conclusiones

Tras la realización del proyecto y observando los resultados finales, se concluye en primera medida, que los objetivos planteados se han logrado y que el alcance del proyecto ha sido el adecuado. Se ha logrado desarrollar un sistema capaz de generar los datos necesarios para poner a prueba circuitos bajo el estándar de interés y se ha elaborado un entorno completamente controlado y versátil acorde con las necesidades de la situación. Este sistema supone el núcleo del banco de pruebas y permite ser escalable en función de los requerimientos, como se esperaba.

Después de llevar a cabo los procesos de investigación, estudio, desarrollo, análisis y simulación, se pueden extraer las siguientes conclusiones:

- En un entorno globalizado, como es el caso de la Industria 4.0 actual, es necesario que todo el conjunto de elementos funcione de forma adecuada y organizada, siendo muy importante que cada aspecto de cada funcionalidad actúe buscando la interoperabilidad. Esto permite una evolución y un desarrollo apropiado, permitiendo no solo el avance tecnológico, sino también el reciclaje tecnológico y la retrocompatibilidad.
- Para el desarrollo de sistemas electrónicos de prueba, como es el caso de este banco de pruebas, es apropiado utilizar una plataforma que permita la detección de errores y el análisis del comportamiento de cada una de las características de dicho circuito. Por ello, es importante la elección inicial sobre cómo se implementa la solución, siendo acertada la elección de utilizar el lenguaje VHDL y el entorno de simulación de Xilinx Vivado. Permite elaborar el sistema desde todos los aspectos posibles, otorgando un control completo sobre el proceso, desde la fase conceptual, hasta la etapa final, pasando por la detección de errores y el desarrollo.

- En redes y sistemas críticos en el tiempo, es necesario que la sincronización sea perfecta, ya que cualquier desfase de un ciclo de reloj, puede desencadenar un error fatal que impida el correcto funcionamiento del circuito.
- Para garantizar una serie de características en las comunicaciones es necesario investigar e innovar. Este estándar es un caso de ello y gracias a este tipo de iniciativas, se plantean nuevas funcionalidades que mejoran la calidad del servicio en gran medida. A pesar de estar en una etapa inicial, es inminente su aplicación en un gran número de sectores gracias a las enormes ventajas que aporta sobre la latencia y la fiabilidad del sistema.

Para concluir, este proyecto permite no sólo poner a prueba el estándar citado durante la memoria, sino que abre las puertas hacia nuevas tecnologías que permitirán mejorar las comunicaciones a nivel global, hasta lograr una red de comunicaciones segura, fiable y en tiempo real, evitando errores y permitiendo que las comunicaciones sean prácticamente instantáneas en cualquier momento y lugar.

Referencias

- [1] G. Martz, “INDUSTRIAL TRANSFORMATION: IOT & OUR LEARNINGS FROM NETWORKING,” 2018.
- [2] J. Farkas and J. Woods, “An Interoperable Ecosystem Through Common Standards and Testing,” 2018.
- [3] K. Forum, “Keyretsu Forum: Blockchain y la industria 4.0.” [Online]. Available: <https://www.keiretsuforum.es/blockchain-y-la-industria-4-0/>.
- [4] M. Wollschlager, T. Sauter, and J. Jasperneite, “The future of industrial Communication,” *IEEE Ind. Electron. Mag.*, vol. 1, no. 1, pp. 9–11, 2017.
- [5] C. Simon, M. Máté, M. Maliosz, and N. Bella, “Ethernet with time sensitive networking tools for industrial networks,” *Infocommunications J.*, vol. 9, no. 2, pp. 6–14, 2017.
- [6] T. Bürger, F. Essler, X. Schmidt, S. Schönegger, and S. Zuponvic, “The Migration-Plans to TSN. Panel on the Battle of Fieldbus Protocols,” 2018.
- [7] A. Rojko, “Industry 4.0 Concept: Background and Overview,” *Int. J. Interact. Mob. Technol.*, vol. 11, no. 5, pp. 77–90, 2017.
- [8] R. Blanco, J. Fontrodona, and C. Poveda, “La industria 4.0: el estado de la cuestión,” *Econ. Ind.*, no. 406, pp. 151–164, 2017.
- [9] Federal Ministry for Economic Affairs and Energy and Federal Ministry of Education and Research, “What is Industrie 4.0?” [Online]. Available: <https://www.plattform-i40.de/PI40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html>.

- [10] K. Schweichhart, “Reference Architectural Model Industrie 4.0 (RAMI 4.0).”
- [11] Plattform Industrie 4.0, “RAMI4.0 – a reference framework for digitalisation.”
- [12] C. Tecnológica, C. Berenice, J. M. Izar, J. Guadalupe, F. Aguilar, and M. Larios, “El Entorno de la Industria 4.0: Implicaciones y Perspectivas Futuras,” *Concienc. Tecnológica*, no. 54, p. 23, 2017.
- [13] T. Burke, “Secured IoT Interoperability. From Sensor to IT Enterprise,” 2018.
- [14] M. P. M. y F. P. M. D. E.-C. I. P. ETI Tudela, “Comunicaciones Industriales: Conceptos generales,” 2008.
- [15] G. Murcia and L. S. Jorge, “Instrumentación Avanzada: Introducción a las Comunicaciones Industriales Scada con LabVIEW.” .
- [16] José M. Hurtado, “Introducción a las Redes de Comunicación Industrial.” p. 19, 2017.
- [17] A. Warren, “Testing Systems with Automotive Ethernet. TSN Overview for Automotive Applications,” 2018.
- [18] G. A. Ditzel and P. Didier, “Time Sensitive Network (TSN) Protocols and use in EtherNet / IP Systems Table of Contents,” *2015 ODVA Ind. Conf. 17th Annu. Meet.*, pp. 1–24, 2015.
- [19] IEEE Computer Society, *IEEE Standard for Ethernet 802.3*, vol. 2015. 2015.
- [20] E. Brockard, “Everything You Need To Know About TSN Sub-Standards &

How To Combine Them.” [Online]. Available:
<https://blog.ebv.com/combining-tsn-sub-standards-knowhow/>.

- [21] IEEE Computer Society, *IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic. IEEE Std 802.3br-2016*. 2016.
- [22] M. Adetunji, P. Meumeu, and G. Nelissen, “On Multi-Level Preemption In Ethernet,” p. 4249, 2016.

Anexos

A. Descripción detallada de la solución

Partiendo de lo expuesto anteriormente en la memoria, se procede a explicar de forma detallada el funcionamiento de los bloques que conforman el sistema.

Se explican inicialmente las funcionalidades globales de cada bloque general, explicando la forma en que se interconectan entre sí los bloques internos. A continuación, se explican las funcionalidades individuales de cada bloque interno.

A.1. Perspectiva global del sistema

A.1.1. Banco de pruebas

El funcionamiento global del bloque consiste en el envío de las tramas generadas a partir de la secuencia preestablecida mediante el bus de datos “data” utilizando el sistema AXI Stream. Se utiliza como reloj de la comunicación “clk”. Para la sincronización entre el banco de pruebas y el circuito testado, se utiliza las señales “ready” y “valid”. “Ready” se utiliza para indicar que el circuito externo está preparado para recibir datos y “valid”, para indicar que el banco de pruebas está emitiendo datos válidos.

De forma paralela, se exportan los datos de las tramas a un fichero con extensión “txt”, es decir, de texto plano, en el que se indican, para cada uno de los paquetes, el tiempo en el que se emite, la longitud de sus datos y los datos propiamente dichos.

A.1.2. Exportación de datos

Los datos exportados, son procesados mediante un script en lenguaje Python que lee los campos indicados y mediante la librería “scapy” se generan los datos del fichero de captura con extensión “pcap”. Se utiliza para ello un proceso iterativo para cada paquete, en el que se obtienen los datos necesarios y se escriben en la salida, con el formato apropiado. El fichero de capturas está preparado entonces para su visualización en el software apropiado, en el caso de la solución, Wireshark.

A.2. Banco de Pruebas

Este bloque consiste en la interconexión de cuatro bloques, permitiendo la correcta coordinación interna del sistema. Permite la coordinación tanto en las directrices de control para avanzar en la secuencia de estados necesaria, como en el intercambio de datos, que son el resultado de dichas directrices.

Para inicializar el sistema, es necesario realizar un “reset” en el momento inicial mediante “rstInBP” y aportar una señal de reloj, que será la que defina los ciclos de operación del sistema. Esta señal de reloj, debe adaptarse a la velocidad del caso particular del protocolo Ethernet utilizado. Aportando entonces la señal de habilitación “enableInBP” y tras la notificación por parte del circuito de prueba mediante “readyInBP”, se procede a iniciar la secuencia de generación.

Para ello, se inicializa cada uno de los bloques en el instante inicial. Tras la inicialización, el bloque de control de generación, comienza a realizar las operaciones internas que definen la secuencia de generación de tramas del entorno controlado. Mediante las señales “eGen” y “pGen” indica el tipo de paquete a generar en cada instante y mediante “mLengthTotal” se indica su longitud total. Mientras no reciba las señales “eTx” o “pTx”, seguirá indicando la

generación de datos, ya que son estas señales, las que indican que la generación/transmisión ha finalizado. Mediante este bloque, se controlan los bloques de generación y de ensamblado de tramas.

El bloque ensamblador de tramas, recibe las señales del bloque de control y en función de las mismas, realiza la sucesión de operaciones requeridas para cumplir con las especificaciones de la secuencia. Mediante “eGen” y “pGen” recibe el tipo de paquete que se debe generar y, una vez el dispositivo externo de prueba indica el “ready”, comienza a generar cada uno de los campos de la trama: preamble, payload, fsc y gap. Mediante “dataOut” proporciona los datos de cada octeto de la trama indicando que el valor es válido mediante “validOut”. “ClkOut” se corresponde con la señal de reloj propia de la comunicación. Cuando se debe generar la información útil de la trama, es decir, el campo payload a partir de las direcciones MAC y el tipo de protocolo de niveles superiores, se habilita el bloque generador a través de la señal “data_gen”. Cuando finaliza la generación/transmisión, se indica mediante “eTx” o “pTx” dependiendo del tipo de la trama generada. De esta forma, se notifica al bloque de control que debe comenzar con la generación de la siguiente trama.

El bloque generador es, por tanto, un subordinado de los bloques de control y ensamblador, ya que depende de sus directrices y habilitadores para comenzar a funcionar. Recibe del bloque de control la longitud total del paquete a generar mediante “mLengthIn”. Cuando el bloque ensamblador decide generar los datos, indica mediante “rstGen” que debe tomar el valor total del paquete del registro “mLengthIn”. A continuación, cuando se habilita el bloque con “data_gen”, se realizan las iteraciones necesarias para proporcionar “mLengthIn” octetos de datos. Los datos se transmiten mediante “mData”, indicando la longitud de cada octeto con “mLength”. Si el dato proporcionado es válido, se indica con “mValid” y en el último octeto de la generación, se activa la señal “mLast”. De esta forma,

se controla de forma completa la obtención de datos adecuados por el resto de bloques.

El bloque encargado de la exportación se explica en un apartado propio, en este mismo anexo.

A.3. Bloque de control de generación

A.3.1. Máquina de Estados

Se corresponde con el bloque “MAQUINA ESTADOS CONTROL GENERACIÓN”. Se encarga de realizar la sucesión de estados de forma ordenada, permitiendo controlar forma segura y completa, las diferentes fases de generación de datos del sistema.

Para ello, debe ser inicializado de forma correcta (“rst” para establecer estado inicial y eliminar posibles estados no deseados en las diferentes variables y señales internas). A continuación, solamente será necesario introducir la señal síncrona del reloj y habilitar el bloque para que comience la etapa de generación (“clk” y “enable”, respectivamente).

El estado inicial será el de “espera”, en el cual habilitará “enable_DataGen” para activar el bloque encargado de determinar la secuencia y naturaleza de los paquetes. Esperará a las directrices de dicho bloque.

Una vez determine qué tipo de paquete se debe generar (“sel”), pasará al siguiente estado, según lo siguiente:

- Si recibe “sel” = “00”: no hace nada y sigue esperando.
- Si recibe “sel” = “01”: pasa al estado de generación de paquete Preemtable.

- Si recibe “sel” = “10”: pasa al estado de generación de paquete Express.
- Si recibe “sel” = “11”: pasa al estado de generación de paquete Express.

En cada uno de estos estados, generará la señal adecuada “eGen” o “pGen”, en función del tipo de paquete a generar.

Esperará entonces la finalización de la generación del paquete completo o la interrupción de la generación en caso de ser necesario (interrupción de Preemtable para envío de Express).

Una vez reciba la señal adecuada, pasará al siguiente estado en función del tipo de paquete que se vaya a generar o pasará a esperar de nuevo en caso de que no se deba generar nada.

A.3.2. Control tipo de paquete

Se corresponde con el bloque “GENERADOR SECUENCIA PAQUETES” del diagrama de bloques. Se encarga de definir la secuencia en que se van a generar los paquetes (el tipo y la longitud de los mismos).

Debe ser inicializado al comienzo y únicamente pasará a generar cuando se habilite “enable”, mediante la salida correspondiente de “Máquina de Estados”.

Una vez se active, seguirá la secuencia de generación establecida internamente. Actúa por tanto como caja negra para definir dicha secuencia de generación, permitiendo mayor flexibilidad al resto de bloques del sistema.

Es de tipo determinista y sigue una secuencia preestablecida (tipo de paquete y longitud del mismo están predefinidos).

La secuencia inicial es la siguiente:

- Paquete Express (completo).
- Paquete Preemptable (completo).
- Paquete Preemptable (interrumpido por Express).
- Paquete Express (que interrumpe al Preemptable anterior).
- Paquete Preemptable (continuación del Preemptable interrumpido).
- Fin.

Proporciona las señales “mSel” que indicarán a la máquina de estados el tipo de paquete a generar y dará la longitud total de dicho paquete “mLength”.

A.4. Bloque de generación

A.4.1. Contador

Se encarga de realizar cuentas en sentido descendiente, partiendo del valor inicial proporcionado mediante “start_count_val”, hasta cero. Este valor debe estar definido antes de realizar el “reset”.

A partir de ese momento, una vez se habilite “enable”, en cada ciclo de reloj realizara un conteo descendiente de una unidad (correspondiente a 1 byte de datos o un octeto), hasta llegar a cero. En cada uno de estos ciclos de reloj, se transmitirá la longitud del paquete a generar “mLength”. En cada ciclo en que se envía información válida, se activa el habilitador de generación de datos “generating”. La señal que se comporta como indicador de final de paquete es “ending” (activa en el último octeto a generar).

A.4.2. Generador datos

Se encarga de generar los datos “mData” y las señales de validación “mValid” y “mLast”, para cada uno de los ciclos de reloj, en función de las señales aportadas por el bloque “Contador” (“generating”, “ending” y “mLength”).

Los datos generados siguen una secuencia cíclica, en función de un contador, también cíclico. Los valores que puede tomar, en función del contador son:

Tabla 8: Valores de datos de la carga útil de las tramas generados por el sistema

Contador	Valor binario	Valor hexadecimal
0	10000000	0x80
1	01000000	0x40
2	10100000	0xA0
3	01010000	0x50
4	10101000	0xA8
5	01010100	0x54
6	10101010	0xAA
7	01010101	0x55

A.5. Bloque ensamblador de tramas

A.5.1. Preamble Generator

Se encarga de generar la secuencia necesaria para indicar el comienzo de la trama ethernet. Es decir, los primeros ocho octetos Esta secuencia consiste en:

- 7 octetos (10101010) + 1 octeto (SMD-XX): Cuando se trata de un paquete Express (SMD-E), un paquete Preemptable completo (SMD-Sx), o el primer fragmento de un Preemptable interrumpido (SMD-Sx).
- 6 octetos (10101010) + 1 octeto (SMD-Cx) + 1 octeto (FRAG_COUNT): Cuando se trata de una continuación de un paquete Preemptable interrumpido. El último octeto corresponde al número del fragmento que se va a transmitir, permitiendo un máximo de 3 paquetes de continuación.

Una vez finaliza la emisión de todos los datos del preamble (mediante “preamble”), se activa la señal “ended” (bit 4 de “endBlock”), para indicar que se debe generar el siguiente campo de la trama.

Para el control de los datos de preamble generados, se utilizan “eGen” y “pGen” para conocer el tipo de paquete a generar, junto a “pInterrupt”, que indica si ha habido una interrupción.

El funcionamiento se activa mediante “enable” (bit 3 de “selBlock”, señal procedente del bloque de selección de bloque generador) y con la señal de reloj “clk”.

A.5.2. Payload Generator

Se encarga de generar la secuencia necesaria para proporcionar la información que va contenida dentro de la trama.

Para ello, se generan sucesivamente los siguientes campos:

- MAC Address Destino (hh:hh:hh:hh:hh:hh): seis octetos, indicando el equipo que al que se desee enviar las tramas.
- MAC Address Origen (hh:hh:hh:hh:hh:hh): seis octetos, indicando el equipo que envía las tramas.

- Ethertype: un octeto, que indica el protocolo utilizado para tecnologías de comunicación de niveles superior al de enlace.
- Data (48-1500 octetos): es la información propiamente dicha que transporta la trama. Son los bytes que se generan en el bloque generador y deberá tomarlos del mismo de la forma adecuada, respetando la secuencia de generación de cada uno de los campos que componen la trama.

Una vez finaliza la emisión de todos los datos del payload (mediante “payload”), se activan las señales “ended_header” (bit 3 de “endBlock”) y “ended_data” (bit 2 de “endBlock”), que indican que ha finalizado la generación de las “cabeceras” (direcciones MAC y Ethertype) y de los “datos”, respectivamente. Indican que se debe generar el siguiente campo de la trama.

Los datos que obtiene del bloque generador, se reciben mediante “mData” y recibe la indicación de final de generación mediante “mLast”.

El funcionamiento se activa mediante “enable” (bit 2 de “selBlock”, señal procedente del bloque de control de generación de trama) y con la señal de reloj “clk”.

A.5.3. Frame Generator Control

Este bloque recibe las señales de control del bloque de control de generación y en función de ellas, realiza la secuencia adecuada de generación de cada uno de los campos que componen la trama ethernet. Es decir, proporciona las señales necesarias para ensamblar la trama de forma adecuada, en función de las condiciones establecidas por el bloque de control.

Genera también las señales necesarias para comunicarse con el bloque de control del sistema, permitiendo un sincronismo adecuado, para la correcta generación de los datos.

Se compone internamente de una máquina de estados, cada uno de ellos definido para los diferentes campos a generar, diferenciando a su vez las diferentes posibilidades de generación.

Mediante su habilitador (“enable”), se podrá controlar el sistema global, ya que será el bloque que controla el ensamblado de las tramas, permitiendo detener y reanudar la generación según las necesidades de las pruebas.

Las señales de control procedentes de los bloques de control y de generación son: “eGen” y “pGen”, que indican el tipo de paquete a generar; “mValid” y “mLast”, que permiten controlar la secuencia de datos generados y “rst”, que permite inicializar el sistema si fuera necesario. Por supuesto, recibe “clk” que es la señal síncrona de reloj del sistema. También cuenta con una señal que indica que la trama ha sido ensamblada al completo y se puede proceder a continuar con la secuencia (“fin_gen”).

Mediante sus salidas controla:

- Inicialización de generador de datos: “rstGen”.
- Control de fragmentos de paquetes interrumpidos: “pInterr”.
- Finalización de transmisión de paquete: “eTx” y “pTx”.
- Validación de dato en buffer para su entrega: “ready”.
- Habilitador para el bloque que se encarga de habilitar de forma consecutiva cada uno de los bloques que generan los diferentes campos de la trama: “mGen”.

A.5.4. FSC Generator

Este bloque recibe los datos generados mediante “mData” y opera sobre cada octeto realizando la operación relativa al cálculo de redundancia cíclico característico de Ethernet, CRC32. Para realizar los cálculos sobre los datos en el momento preciso, es necesario que el habilitador “enable” esté activo en el ciclo en el que se desea operar. Para finalizar el cálculo del CRC, se utilizan las señales “mLast” y “pInterrupt”, las cuales indican si ha finalizado la trama de forma completa o si ha sido interrumpida, respectivamente. La señal de reloj es síncrona con el sistema y se introduce mediante “clk”. Para una mejor sincronización con el bloque de control, se utiliza la señal “data_gen”, conociendo en todo momento el estado de generación del sistema.

Cuando el CRC se ha calculado y emitido completamente a través de “fsc”, se activa la señal “ended” (bit 1 de “endBlock”), para indicar que se debe continuar con la siguiente fase de ensamblado de la trama en cuestión.

A.5.5. Select Generating Block

Se encarga de realizar la secuencia propiamente dicha de generación de campos de la trama de forma adecuada para que los datos se encuentren disponibles en el ciclo de reloj que les corresponde.

Es controlado por “Frame Generator Control” y va habilitando los diferentes bloques de generación mediante “selBlock” (4 bits). Los bits utilizados son:

- Bit 3: preamble.
- Bit 2: payload
- Bit 1: fsc.
- Bit 0: gap (permite inclusión de otro bloque si fuera necesario).

La finalización de generación, es indicada por cada uno de estos bloques que habilita, mediante su correspondiente bit, dentro de la señal “endBlock” (5 bits).

- Bit 4: preamble.
- Bit 3: payload headers (MAC Addr + Ethertype).
- Bit 2: payload data.
- Bit 1: fsc.
- Bit 0: gap.

Mediante las salidas se puede controlar qué tipo de dato se está generando en cada momento: “header_gen” si se trata de cabeceras o campos de control y “data_gen” si se trata de información procedente del bloque generador inicial.

A.5.6. Select Generated Data

Se trata básicamente de un multiplexor que permite obtener en un único bus, los datos de toda la trama, obteniendo los datos del bloque que debe generar en cada momento.

La señal de selección es la procedente del bloque selector “selBlock” y las entradas de datos serán: “preamble”, “payload” y “fsc”. La selección debe ser síncrona, por lo que se le proporciona también la señal de reloj “clk”.

A.6. Bloque exportador de datos generados

Bloque encargado de la exportación de los datos proporcionados por la salida “dataOut” del bloque ensamblador. Cuando el bloque exportador recibe la señal activa “valid”, comienza a almacenar los datos contenidos en “data”. Cuando finaliza la generación de un paquete, se realiza el volcado de dicha información en un fichero de texto plano. Este bloque debe ser síncrono con la comunicación y el sistema, y debe recibir la misma señal de reloj “clk”.

B. Diagramas de bloques

Si se desea visualizar los diagramas de bloques con mayor detalle y calidad, se incluyen en el directorio “Ficheros” dentro del subdirectorio “Diagramas”.

B.1. Perspectiva global del sistema

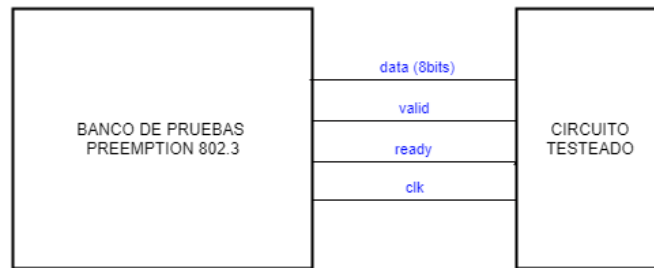


Figura 42: Diagrama de bloques del sistema global

B.2. Banco de Pruebas

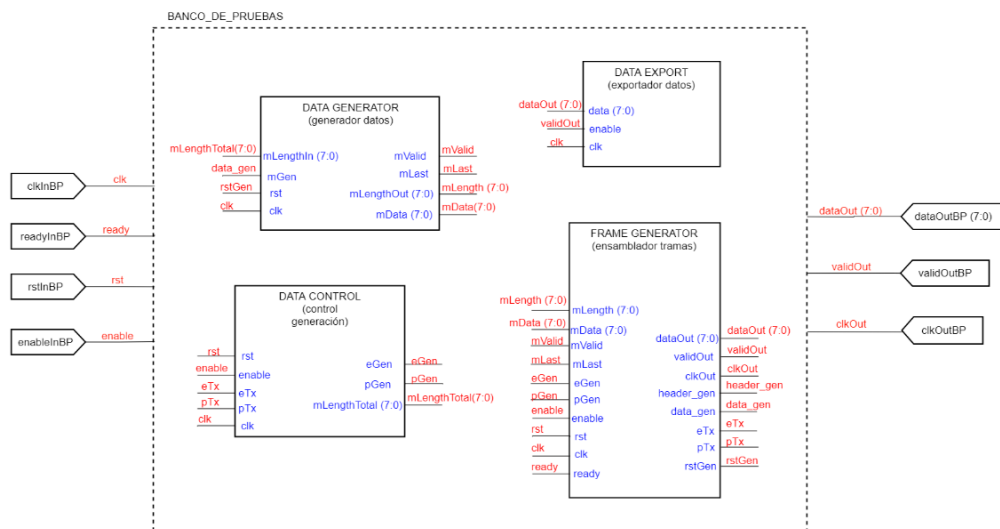


Figura 43: Diagrama de bloques de “Banco de Pruebas”

B.3. Bloque de control de generación

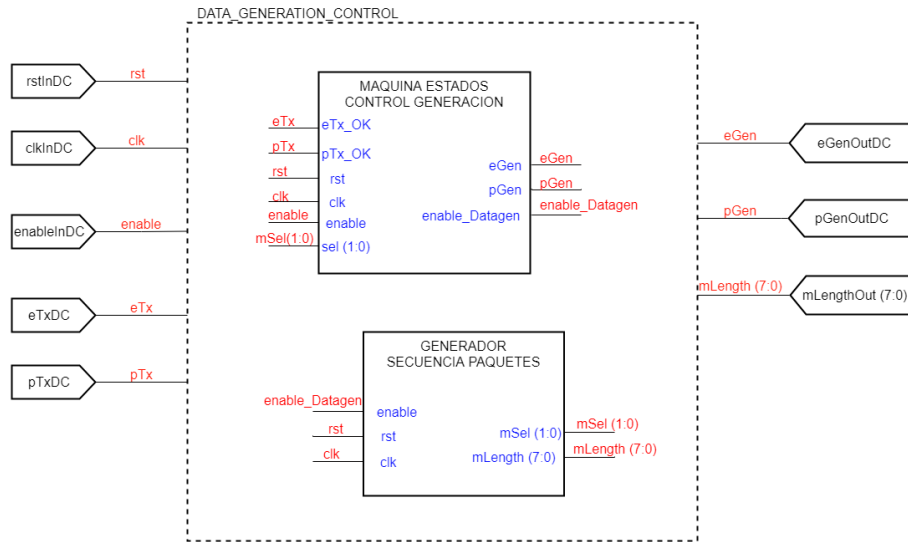


Figura 44: Diagrama de bloques de "Bloque de control de generación"

B.4. Bloque de generación

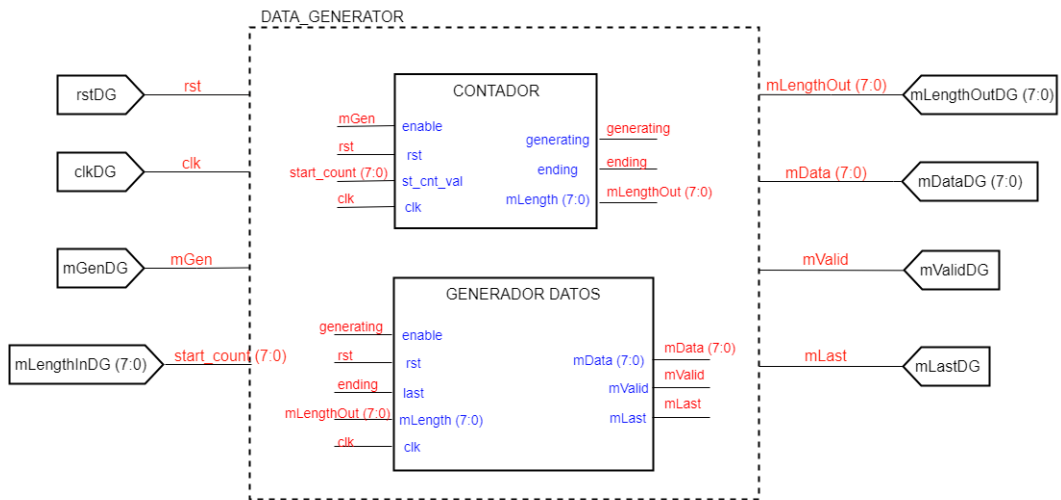


Figura 45: Diagrama de bloques de "Bloque de generación"

B.5. Bloque ensamblador de tramas

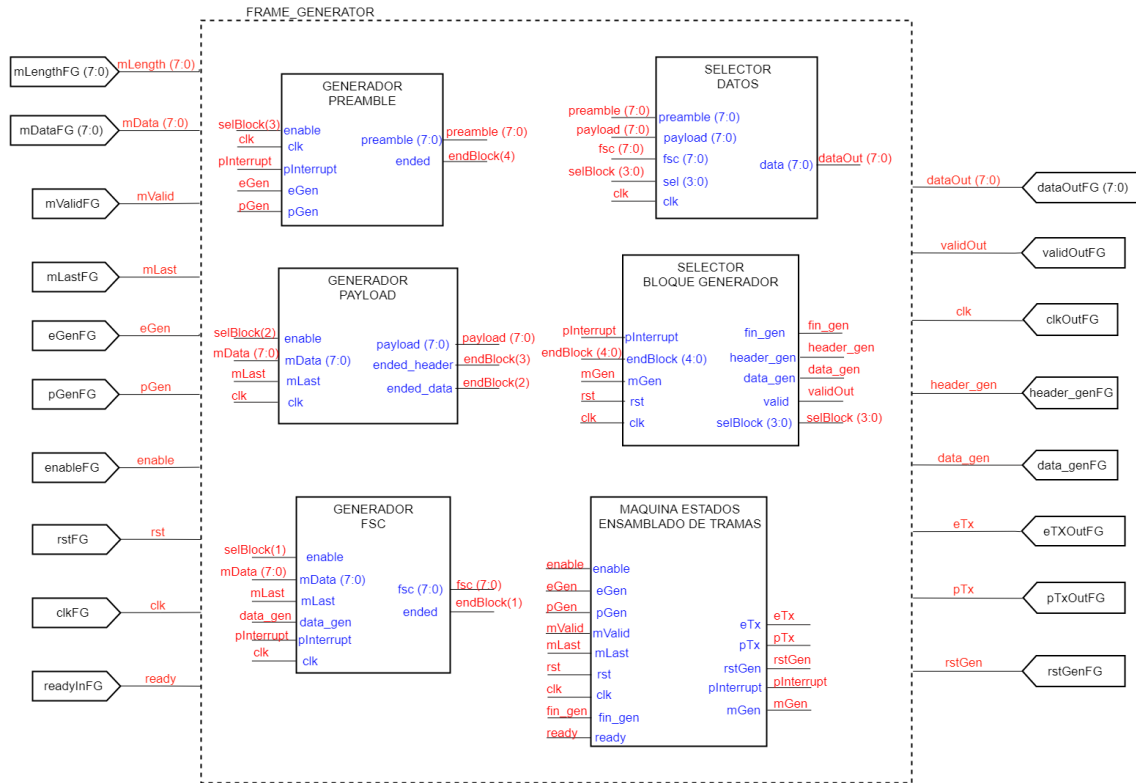


Figura 46: Diagrama de bloques de "Bloque ensamblador de tramas"

C. Diagramas de estados

Si se desea visualizar los diagramas de estados con mayor detalle y calidad, se incluyen en el directorio “Ficheros” dentro del subdirectorio “Diagramas”.

C.1. MAQUINA ESTADOS CONTROL GENERACION

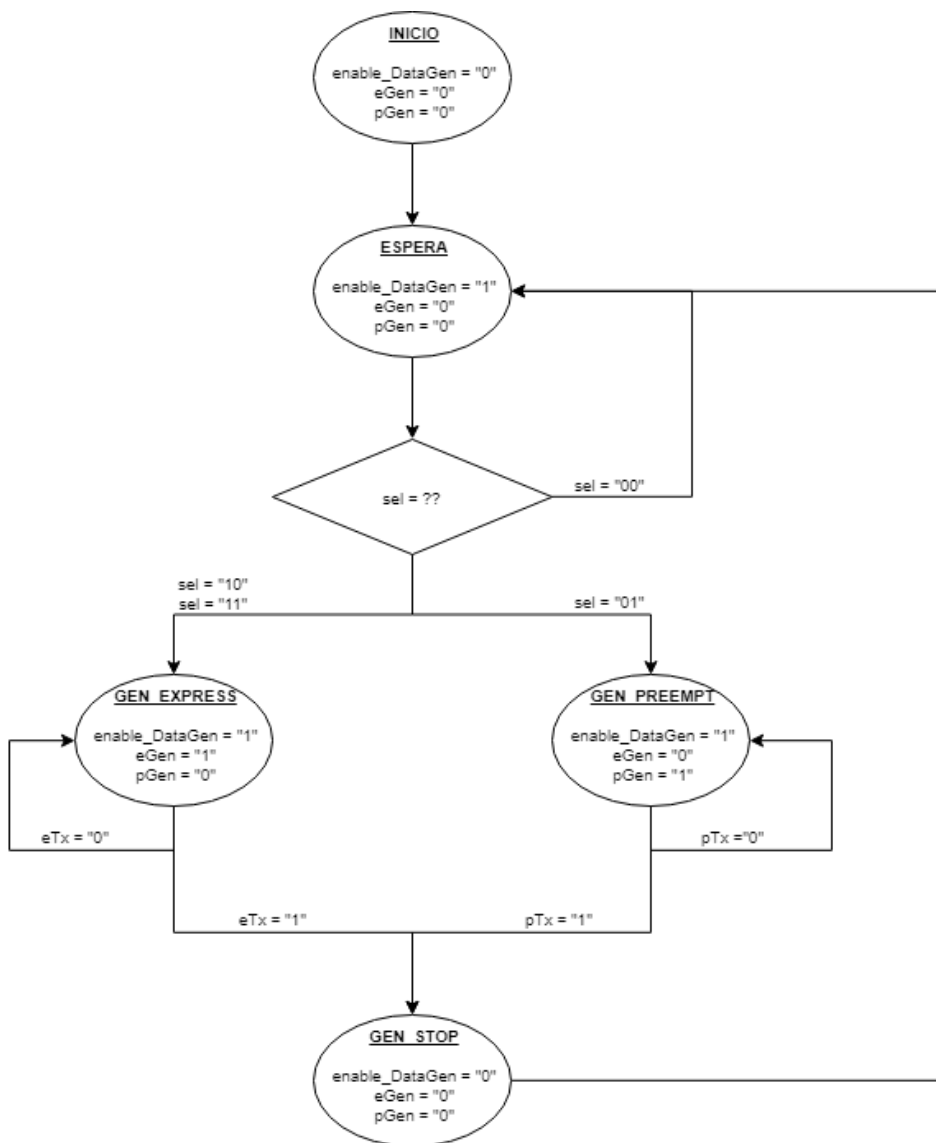


Figura 47: Diagrama de estados de bloque de control de generación

C.2. MAQUINA ESTADOS ENSAMBLADO DE TRAMAS

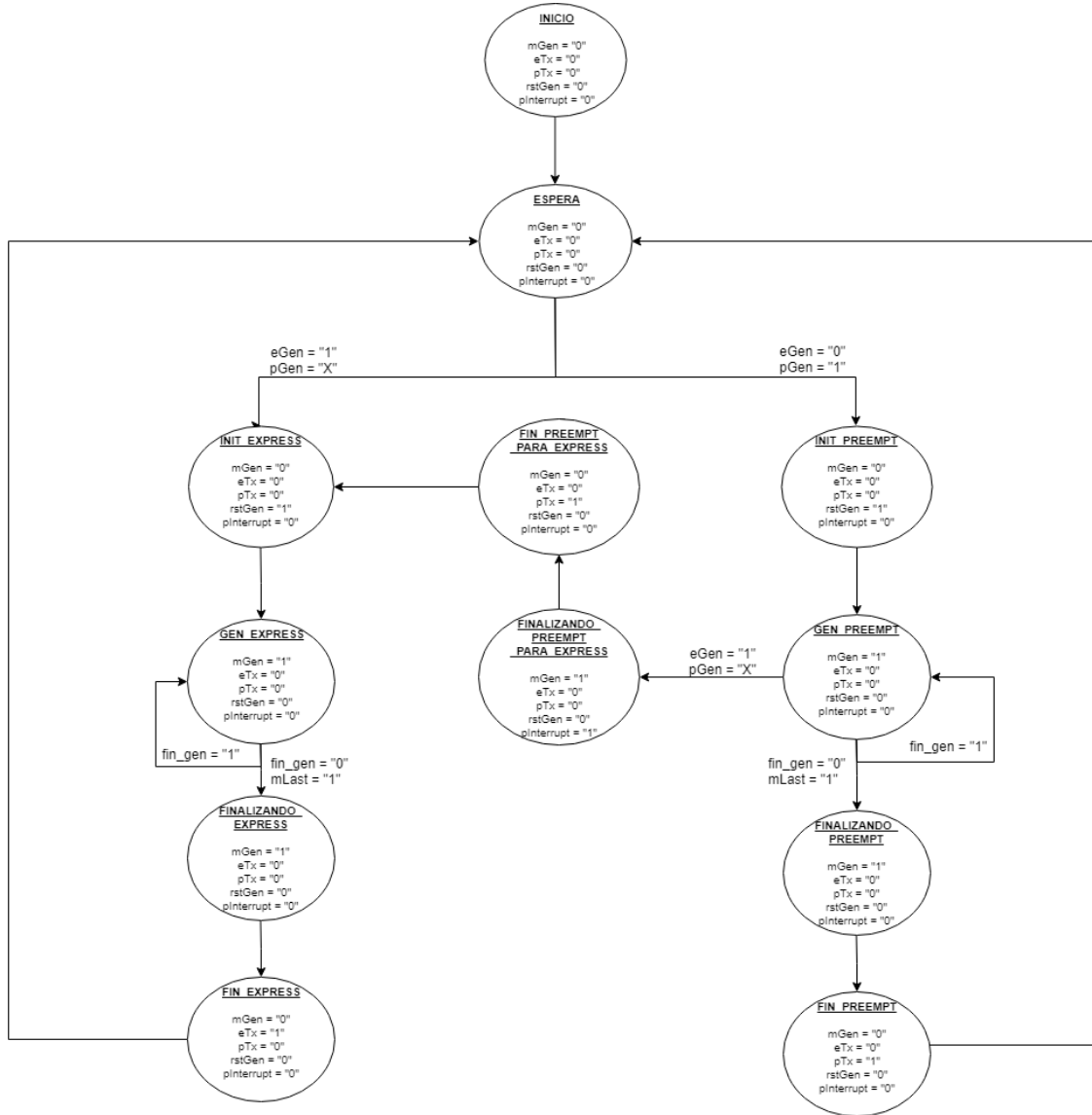


Figura 48: Diagrama de estados de bloque de control de ensamblado de tramas

C.3. SELECTOR BLOQUE GENERADOR

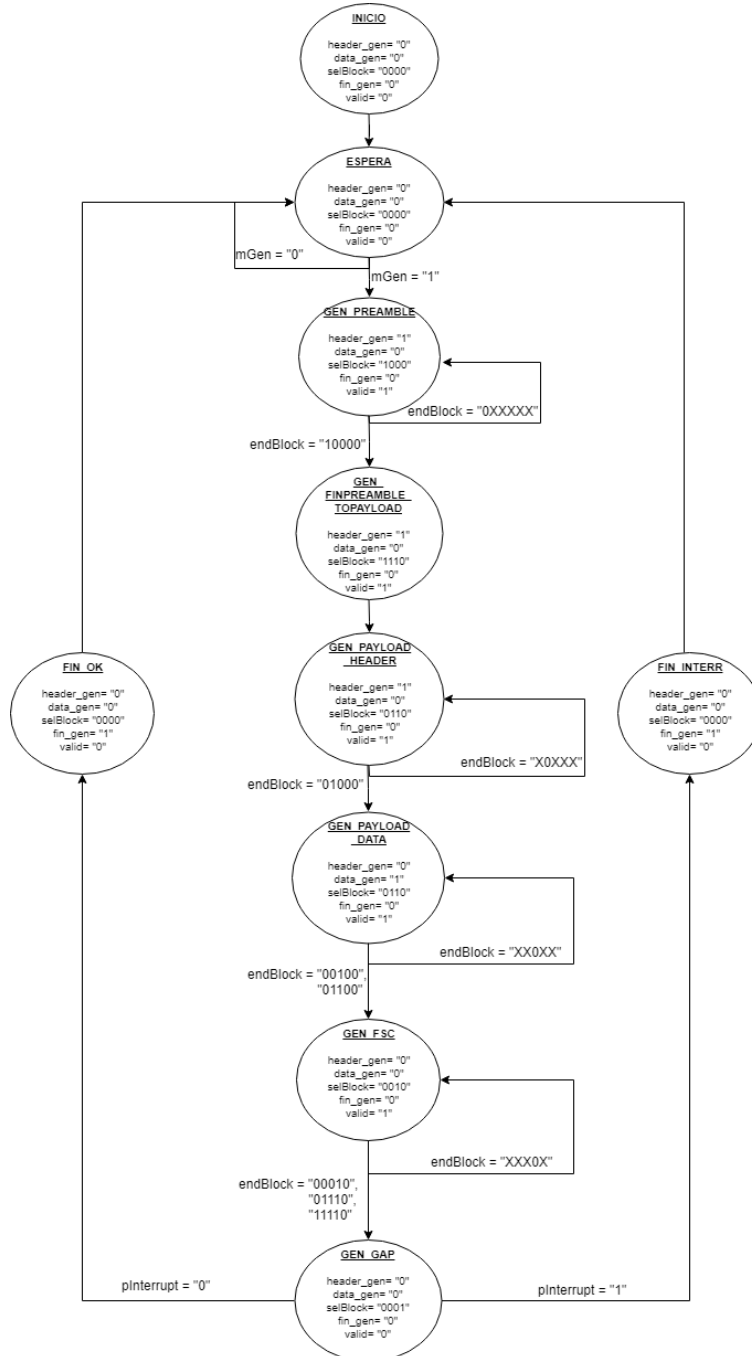


Figura 49: Diagrama de estados de bloque de selección de generador de ensamblado de tramas

D. Código

El código completo, se incluye en el directorio “Ficheros”. Para diferenciar los diferentes tipos de ficheros y funcionalidades del código, se añaden varios subdirectorios:

- “Fuentes”: contiene el código desarrollado en lenguaje VHDL para la definición de los diferentes bloques que conforman el sistema completo.
- “Simulación”: contiene los ficheros denominados “testbenchs”, los cuales permiten simular sus respectivos bloques codificados en los ficheros del directorio “Fuentes”.
- “Exportación”: contiene los ficheros de datos exportados por el sistema para la secuencia definida en los resultados, así como el código en lenguaje Python necesario para procesar dicha información.

Para cada uno de los bloques generales, se incluye una tabla en la que se indica su correspondiente fichero de codificación y simulación de los directorios “Fuentes” y “Simulación” respectivamente.

Tabla 9: Ficheros que contienen el código de los bloques generales del sistema

Bloque	Código VHDL	Código Testbench
BANCO DE PRUEBAS	Banco_Pruebas_Preemption.vhd	testbench_BANCO_PRUEBAS_PREEMPTION.vhd
DATA CONTROL	Data_Control.vhd	testbench_Data_Control.vhd
DATA GENERATOR	Data_Generator.vhd	testbench_Data_Generator.vhd
FRAME GENERATOR	Frame_Generator.vhd	testbench_Frame_Generator.vhd
DATA EXPORT	Data_Export.vhd	

Para cada bloque general, se incluye una tabla en la que se indica el código de desarrollo y de simulación de sus bloques internos.

Tabla 10: Ficheros que contienen el código de los bloques internos del bloque "DATA CONTROL"

Bloque	Código VHDL	Código Testbench
MAQUINA ESTADOS CONTROL GENERACION	maquina_estados_ generador.vhd	testbench_maquina_esta dos_generador.vhd
GENERACION SECUENCIA PAQUETES	control_tipo_paque te.vhd	testbench_control_tipo_p aquete.vhd

Tabla 11: Ficheros que contienen el código de los bloques internos del bloque "DATA GENERATOR"

Bloque	Código VHDL	Código Testbench
CONTADOR	contador_dsc.vhd	testbench_contador.vhd
GENERADOR DATOS	number_generator.v hd	testbench_generador_datos.v hd

Tabla 12: Ficheros que contienen el código de los bloques internos del bloque "FRAME GENERATOR"

Bloque	Código VHDL	Código Testbench
MAQUINA ESTADOS ENSAMBLADO DE TRAMAS	frame_generator_ control.vhd	testbench_frame_generat or_control.vhd
SELECTOR BLOQUE GENERADOR	select_generating _block.vhd	testbench_select_generat ing_block.vhd
SELECTOR DATOS	select_generated_ data.vhd	testbench_select_generat ed_data.vhd
GENERADOR PREAMBLE	preamble_generat or.vhd	testbench_preamble_gen erator.vhd
GENERADOR PAYLOAD	payload_data_gen erator.vhd	testbench_payload_data_ generator.vhd
GENERADOR FSC	fsc_generator.vhd	

Los ficheros correspondientes al código utilizado para el procesamiento de información se incluyen en la siguiente tabla.

Tabla 13: Ficheros que contienen el código Python para procesamiento de datos exportados

Función	Fichero
Conversión fichero de texto plano a fichero de capturas	Convert_Exported_Seq_To_PCAP.py
Cálculo de CRC32 para los datos del fichero de texto plano	CRCcheck.py