

GRADO EN INGENIERÍA EN TECNOLOGÍA DE
TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

ANÁLISIS DE LAS RETRANSMISIONES EN SISTEMAS WIRELESS-NOMA PARA APLICACIONES INDUSTRIALES

Alumno: Albisu González, Mario
Director: Angueira Buceta, Pablo
Codirector: Iradier Gil, Eneko

Curso: 2018-2019

Fecha: 18 de febrero de 2019

ÍNDICE

ÍNDICE	2
2) Lista de tablas, figuras y acrónimos	4
2.1) Tablas	4
2.2) Figuras	5
2.3) Acrónimos	6
3) Resumen	7
4) Introducción	10
5) Contexto	12
6) Objetivos y Alcance	15
6.1) Objetivos	15
6.2) Alcance	16
7) Beneficios	17
7.1) Beneficios técnicos	17
7.2) Beneficios económicos	17
7.3) Beneficios sociales	18
8) Estado del arte	19
8.1) Non-Orthogonal Multiple Access (NOMA)	19
8.2) Familia IEEE 802.11	22
8.2.1) IEEE 802.11g	23
8.2.2) IEEE 802.11n	28
8.3) Synchronous and Hybrid Architecture for Real-time Performance (SHARP)	33
9) Metodología	36
9.1) Sistema de simulación de red en OMNet++	36
9.2) Propuestas MAC de retransmisiones	39
9.2.1) Modelo simple	40
9.2.2) Cadena completa	40
9.3) Codificación de la capa MAC	42
9.3.1) Modelo simple	42
9.3.2) Cadena completa	43
9.4) Escenario de simulación y resultados obtenidos	44
9.4.1) Cadena completa	44
9.4.2) Modelo simple	44

10) Planificación	49
10.1) Equipo de trabajo	49
10.2) Fases del trabajo.....	49
10.3) Diagrama de Gantt.....	53
11) Presupuesto y costes.....	54
11.1) Horas internas.....	54
11.2) Amortizaciones.....	54
11.3) Gastos totales	55
12) Análisis de riesgos	56
12.1) Identificación y evaluación de los riesgos.....	56
12.2) Planes de contingencia	57
13) Conclusiones y visión de futuro.....	59
Referencias	61
Anexo 1: Código OMNET++	63
1.1) Cadena Completa.....	63
1.2) Modelo Simple.....	70

2) LISTA DE TABLAS, FIGURAS Y ACRÓNIMOS

2.1) Tablas

Tabla 1. Requisitos de funcionamiento para servicios industriales.....	13
Tabla 2. Ganancia relativa de LDM sobre TDM en recepción móvil y fija.....	20
Tabla 3. MCS correspondientes a 802.11g.....	24
Tabla 4. Límites teóricos SNR en 802.11g + NOMA (2 capas) vs TDM.....	27
Tabla 5. MCS correspondientes a 802.11n en régimen SISO y canales de 20 MHz ...	29
Tabla 6. Límites teóricos SNR NOMA 802.11n vs TDM.....	31
Tabla 7. Ganancia PHY relativa NOMA vs TDM 802.11n.....	32
Tabla 8. Tasa de datos máxima alcanzable BE (Mbps).....	35
Tabla 9. Tabla comparativa tecnologías de retransmisión.....	39
Tabla 10. Parámetros de simulación del modelo simple.....	45
Tabla 11. Resultados brutos de recepción de paquetes 20x20.....	46
Tabla 12. Resultados PER 20x20.....	46
Tabla 13. Resultados brutos de recepción de paquetes 100x100.....	46
Tabla 14. Resultados PER 100x100.....	47
Tabla 15. Comparativa PER.....	47
Tabla 16. Latencia media 100x100.....	47
Tabla 17. Equipo de trabajo.....	49
Tabla 18. PT0. Coordinación, gestión y supervisión del proyecto.....	49
Tabla 19. PT1. Planificación del proyecto.....	49
Tabla 20. PT2. Estudio del estado del arte.....	50
Tabla 21. PT3. Profundización en el programa OMNet++.....	50
Tabla 22. PT4. Desarrollo de la aplicación.....	51
Tabla 23. PT5. Simulación y obtención de PER y latencia.....	51
Tabla 24. PT6. Redacción de informe.....	51
Tabla 25. Entregables.....	52
Tabla 26. Hitos.....	52
Tabla 27. Horas internas.....	54
Tabla 28. Amortizaciones.....	54
Tabla 29. Presupuesto completo.....	55

2.2) Figuras

Figura 1. Esquema de multiplexación multicapa para dos servicios.....	19
Figura 2. Esquema de funcionamiento LDM	21
Figura 3. Comparativa de eficiencia espectral NOMA vs OMA para $SNR_m = 0$ dB	21
Figura 4. Ganancia relativa máxima LDM sobre TDM/FDM	22
Figura 5. División de canales en la banda de 2.4 GHz con canales no solapables	23
Figura 6. Canales no solapables usados en 802.11g con OFDM.....	24
Figura 7. Formato de paquete en 802.11g/a	25
Figura 8. Curvas PER-SNR de 802.11g para cada MCS en tecnología unicapa.....	26
Figura 9. Curvas límite PER-SNR de 802.11g NOMA.....	27
Figura 10. Canal no solapable en 802.11n con canales de 40 MHz.....	28
Figura 11. Formato de paquete en 802.11n modo mixto	30
Figura 12. Límites SNR teóricos para 802.11n unicapa y $PER = 10^{-2}$	31
Figura 13. Curvas PER-SNR obtenidas para servicio crítico CL	32
Figura 14. Topología de red SHARP.....	33
Figura 15. Capa MAC de SHARP – Supertrama TDMA.....	33
Figura 16. Tramas PHY 802.11g (a) y Agregación de tramas SHARP DL	34
Figura 17. Formato trama SHARP UL.....	34
Figura 18. Resultados PER y PLR en servicio RT	35
Figura 19. Asignación temporal en NOMA	37
Figura 20. Asignación temporal en esquemas TDMA	37
Figura 21. Esquema de la red en OMNet++ y detalle del módulo AP.....	38
Figura 22. Esquema y descripción de casos de análisis	40
Figura 23. Diagramas de flujo del mecanismo de retransmisiones DL	41
Figura 24. Esquema de funcionamiento modelo simple	42
Figura 25. Detalle de la información de paquete y ejemplo de comunicación	44
Figura 26. Diagrama de Gantt.....	53

2.3) Acrónimos

AP	Access Point
ARQ	Automatic Repeat Request
ATSC	Advanced Television Systems Committe
AWGN	Additive White Gaussian Noise
BE	Best Effort
BER	Bit Error Rate
BPSK	Binary Phase-Shift Keying
CL	Core Layer
CM	Condition Monitoring
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DCF	Distributed Coordination Function
DL	Down-Link
FA	Factory Automation
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
HARQ	Hybrid Automatic Repeat Request
HP	Hligh Performance
HT	High Throughput
IE	Industrial Ethernet
ISI	Intersymbol Interference
IWSAN	Industrial Wireless Sensor and Actuatot Network
IWSN	Industrial Wireless Sensor Network
LDM	Layer Division Multiplexing
LDPC	Low-Parity-Check-Codes
LL	Lower Layer
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
NOMA	Non-Orthogonal Multiple Access
OFDM	Orthogonal Frequency Division Multiplexing
PA	Process Automation
PER	Packet Error Rate
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RT	Real Time
SHARP	Synchronous and Hybrid Architecture for Real-time Performance
SNR	Signal-to-Noise Ratio
TDMA	Time Division Multiple Access
UL	Upper Layer
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

3) RESUMEN

La Industria 4.0 escenifica la próxima gran revolución industrial a nivel global. Sus objetivos son hacer llegar una completa transformación en los apartados de la producción, automatización e interconexión dentro de los entornos industriales. Como sabemos, la transmisión y comunicación fiable de datos se antoja un elemento clave a la hora de trabajar en estos entornos. Por ello, hasta ahora se ha venido apostando por soluciones cableadas que garantizan los requisitos de latencia y fiabilidad mínimos con las contrapartidas de la falta movilidad, precio y poca escalabilidad asociadas. Con el desembarco de esta nueva era en el sector industrial, se pretende mitigar estas desventajas mediante tecnologías inalámbricas existentes. Sin embargo, la implementación de estas tecnologías en entornos industriales supone un reto en el aspecto de los requisitos de latencia y fiabilidad. Es por ello por lo que su despliegue a gran escala no se ha hecho efectivo todavía.

Además, si tenemos en cuenta que las aplicaciones industriales no solo están basadas en situaciones de estrictas tasas de latencia y pérdida de paquetes, sino que también existen aplicaciones más flexibles en sus requisitos, podemos distinguir dos tipos de tráfico: el tráfico Real Time (RT) o de Tiempo Real y el Best Effort (BE) o de Mejor Esfuerzo. Por ello, es fundamental definir la solución PHY/MAC inalámbrica que garantice los requisitos mínimos para los dos tipos de servicios de la manera más eficiente posible.

El objetivo de este proyecto es aportar el estudio y análisis de la implementación de retransmisiones de tramas al proyecto de comunicaciones industriales basado en tecnología NOMA (Non-Orthogonal Multiplexing Access) como capa física para transmitir ambos servicios y el estándar IEEE 802.11n. Con ello, se evaluará la eficacia de este mecanismo para reducir el PER (Packet Error Rate) y los posibles compromisos a realizar en términos de latencia con respecto a la solución sin retransmisiones.

Palabras clave: Industria 4.0, tecnologías inalámbricas, comunicación fiable, Real Time, Best Effort, NOMA, IEEE 802.11n, PER, latencia.

Industry 4.0 is set up to be the next great industrial revolution on a global level. Its objectives are to achieve a complete transformation in the areas of production, automation and interconnection within industrial environments. As we know, the transmission and reliable communication of data are key elements when working in these environments. For this reason, until now manufacturers tend to use wired solutions that guarantee the minimum latency and reliability requirements with the counterparts of the lack of mobility, price and low scalability associated. With the landing of this new era in the industrial sector, the intention is to mitigate these disadvantages through existing wireless technologies. However, the implementation of these technologies in industrial environments poses a challenge in terms of latency and reliability requirements. That is why its large-scale deployment has not yet become effective.

In addition, if we take into account that industrial applications are not only based on strict latency and packet loss rates situations, but that there are also more flexible applications in their requirements, we can distinguish two types of traffic: Real Time traffic (RT) and the Best Effort (BE) traffic. Therefore, it is essential to define the wireless PHY/MAC solution that guarantee the minimum requirements for the two types of services in the most efficient way possible.

The objective of this project is to provide the study and analysis of a frame retransmission implementation into the industrial communications project based on NOMA technology (Non-Orthogonal Multiplexing Access) as a physical layer to transmit both services and the IEEE 802.11n standard. With this, the effectiveness of this mechanism to reduce the PER (Packet Error Rate) and the possible trade-offs made in terms of latency with respect to the solution without retransmissions are going to be evaluated.

Keywords: Industry 4.0, wireless technologies, reliable communication, Real Time, Best Effort, NOMA, IEEE 802.11n, PER, latency.

L'Industrie 4.0 met en scène la prochaine grande révolution industrielle au niveau mondial. Ses objectifs sont de réaliser une transformation complète dans les domaines de la production, de l'automatisation et de l'interconnexion dans les environnements industriels. Comme nous savons, la transmission et la communication fiable de données semblent être un élément clé lorsque l'on travaille dans ces environnements. Pour ceci, jusqu'à présent on a misé sur des solutions câblées qui garantissent les exigences minimales de latence et de fiabilité avec les contreparties de la manque de mobilité, du prix et de la faible évolutivité associées. Avec le débarquement de cette nouvelle ère dans le secteur industriel, il est prévu d'atténuer ces inconvénients grâce aux technologies sans fil existantes. Cependant, la mise en œuvre de ces technologies dans des environnements industriels pose un défi en termes de latence et de fiabilité. C'est pourquoi son déploiement à grande échelle n'est pas encore effectif.

De plus, compte tenu du fait que les applications industrielles ne reposent pas uniquement sur des situations strictes de latence et de taux de perte de paquets, mais que leurs applications peuvent être aussi plus flexibles, nous pouvons distinguer deux types de trafic: le trafic Real Time (RT) ou en Temps Réel et le Best Effort (BE) ou Meilleur Effort. Par conséquent, il est essentiel de définir la solution PHY/MAC sans fil qui puisse garantir les exigences minimales pour les deux types de services de la manière la plus efficace possible.

L'objectif de ce projet est de fournir l'étude et l'analyse de la mise en œuvre de retransmissions de trames au projet de communications industrielles basé sur la technologie NOMA (Non-Orthogonal Multiplexing Access) en tant que couche physique pour la transmission des services et de la norme IEEE 802.11n. Cela permettra d'évaluer l'efficacité de ce mécanisme pour réduire le PER (taux d'erreur de paquet) et les

éventuels engagements à prendre en termes de latence vis-à-vis de la solution sans retransmissions.

Mots-clés: Industrie 4.0, technologies sans fil, communication fiable, Real Time, Best Effort, NOMA, IEEE 802.11n, PER, latence.

4) INTRODUCCIÓN

Las redes industriales de sensores inalámbricos o IWSN (Industrial Wireless Sensor Networks) reposan en la total interconexión de los elementos en un entorno industrial. Generalmente, estas redes consisten en pequeños dispositivos distribuidos llamados nodos que recolectan parámetros tales como temperatura, humedad o presión fundamentales para la monitorización de los procesos industriales. A su vez y mediante saltos entre nodos, estos datos se hacen llegar hasta una estación central o AP (Access Point) donde se analizan y se procesan. Estas aplicaciones demandan por lo tanto una gran eficiencia energética y unos estrictos estándares de transporte de datos inalámbrico. En especial, los datos deben llegar a un destino de forma fiable y en un tiempo establecido para garantizar determinismo en los procesos. Esto último constituye uno de los principales obstáculos a la hora de implementar estas redes en casos de uso reales.

Por ello, uno de los grandes apartados en los que se trabaja actualmente en el ámbito de la investigación es en satisfacer los requisitos del sector industrial mediante tecnologías y estándares inalámbricos ya existentes actualmente que no fueron diseñados para trabajar en estos entornos. En concreto, se realizaron trabajos sobre opciones basadas en la familia IEEE 802.11/802.15 para la capa física e innovaciones en la capa MAC sobre TDMA. Los intentos dejaron claro que hasta el momento los resultados son muy dependientes de la aplicación en cuestión y que no existe una tecnología ganadora.

En este sentido, este proyecto se encuadra dentro de un novedoso sistema basado en NOMA del estándar IEEE 802.11. NOMA pretende ser la solución a los defectos que poseen otros métodos de asignación de recursos como TDMA (Time Division Multiple Access) o FDMA (Frequency Division Multiple Access). En él, la señal transmitida se compone de un conjunto de señales divididas por capas, cada una con diferente robustez, capacidades y sensibilidad. Así, en situaciones de tráfico asimétrico (un servicio más rígido en sus requisitos que el otro), esta tecnología presenta una mejora con respecto a las anteriormente propuestas. Además, en este proyecto se detallaron las primeras etapas hacia una posible estandarización: contexto teórico, diseño de las arquitecturas receptor-transmisor, simulaciones y análisis a nivel de enlace. Se propuso además una capa MAC (Medium Access Control) que se adecuase a las especificaciones de NOMA.

Uno de los puntos que se incluyó en este proyecto como vía de profundización fue la implementación de retransmisiones en este sistema. En este documento, se abordará este tema con el estudio de un modelo de retransmisiones de paquetes en caso de error en la transmisión. El principal objetivo es valorar los beneficios y contrapartidas de la inclusión de un sistema de recuperación a nivel MAC a partir del análisis de parámetros de calidad de la comunicación (PER, latencia) y su comparación con los requisitos de la industria.

Para ello, en primer lugar, se detallarán los conocimientos técnicos necesarios para acotar el proyecto en base a literatura y temas relacionados. Cabe destacar el análisis de la tecnología NOMA, del estándar IEEE 802.11n y del software empleado, OMNet++, además de detallar casos de uso reales mediante otras tecnologías.

A continuación, se detallará la metodología seguida para realizar la implementación, haciendo especial hincapié en los pasos seguidos y su integración en el proyecto de referencia. Asimismo, se simulará un caso práctico con el nuevo modelo creado. Esto constituirá la base para analizar los resultados en comparación con los obtenidos sin las retransmisiones y evaluar la idoneidad de la solución.

Más adelante, se enumerarán los beneficios que se pretenden conseguir con la realización de este proyecto, así como el desglose de la planificación y viabilidad económica del proyecto. En concreto, se indicarán el presupuesto y la organización seguidas, además del análisis de los riesgos acecidos durante la realización del trabajo.

Finalmente, en las conclusiones se procederán a debatir los resultados y a conformar las posibles líneas de profundización. Llegados a este punto, se llegará a una resolución acerca de los compromisos a realizar entre el positivo aumento de fiabilidad y el dañino aumento de latencia.

La realización del proyecto se ha llevado a cabo de manera conjunta con el grupo de investigación TSR (Tratamiento de la Señal y Radiocomunicaciones) de la Escuela de Ingeniería de Bilbao.

5) CONTEXTO

La Industria 4.0 presenta un nuevo paradigma de interconexión de los dispositivos involucrados en el proceso industrial. En él, entidades distribuidas con capacidades de sensado, procesado y ejecución de tareas son monitorizadas y controladas en tiempo real por medio de algoritmos de decisión [5]. Si bien en algunas de estas aplicaciones se dependa de conexiones cableadas, el verdadero reto recae en la adopción de redes inalámbricas o IWSN.

Hasta el momento, las redes industriales basadas en Industrial Ethernet (IE) copan la mayoría de la cuota de mercado. En estas [2], se consigue una alta tasa de datos con ciclos de comunicación muy cortos, algo indispensable en estas redes. Sin embargo, estos canales de comunicación llevan inherentemente asociados un bajo grado de adaptabilidad y escalabilidad. En un sector industrial en el que la movilidad de equipos cobra cada vez más importancia, las redes cableadas no son óptimas: cables expuestos a cargas mecánicas y altos costes de mantenimiento.

Para subsanar estos inconvenientes se está optando entonces por la tecnología inalámbrica, que reduce substancialmente los costes de instalación, diseño y mantenimiento además de proporcionar soluciones en entornos donde instalar cableado es imposible. Asimismo, con el rápido crecimiento de estas tecnologías los dispositivos inalámbricos son en general más baratos que sus homólogos cableados. Sin embargo, la mera instalación de las tecnologías de propósito general en los entornos industriales no es garantía de que se satisfagan los estrictos requisitos del sector.

Se necesitan hacer uso por lo tanto de las redes inalámbricas de altas prestaciones, o Wireless HP (High Performance) [14], una familia de tecnologías que garantice los siguientes apartados y características:

- **Determinismo:** los paquetes deben entregarse en una franja de tiempo (deadline) y con una periodicidad establecidas. Esto se debe a que normalmente el controlador debe conocer los datos actualizados de los elementos distribuidos para enviar las órdenes a los actuadores.
- **Tamaño de paquetes pequeño:** los mensajes intercambiados en aplicaciones industriales son normalmente muy cortos, con una carga útil del orden de 10^2 o inferior.
- **Baja latencia:** se refiere al intervalo de tiempo entre el envío y la correcta recepción de un paquete. Se mide en términos de jitter o de retardo de transmisión y tiene que estar en el orden de los pocos milisegundos.

- Alta fiabilidad: hace referencia a la cantidad de datos correcta recibida en destino. Los parámetros asociados son el BER (Bit Error Rate) y PER (Packet Error Rate). Debido a la naturaleza crítica para la seguridad de los mensajes que se envían, estos deben entregarse de forma fiable.
- Control y gestión centralizados: una arquitectura de red centralizada donde un controlador envíe comandos y reciba datos de nodos distribuidos. Si es necesario se utilizarán subredes con la misma arquitectura en instalaciones a gran escala.
- Canal semiestático y despliegue conocido: la posición y número de nodos es conocido de antemano en una instalación industrial, al contrario de lo que ocurre en otros escenarios. Además, a pesar de que ciertos elementos requieran de movilidad, su velocidad suele ser baja. Por ello, el canal presenta variaciones más lentas, con lo que es menos selectivo en tiempo.

Por otro lado, podemos clasificar [2] las comunicaciones industriales en diferentes áreas de servicios: Condition Monitoring (CM) o monitorización de estado, Process Automation (PA) o Automatización de Procesos y Factory Automation (FA) o Automatización Industrial. Cada una presentará necesidades específicas en términos de sincronización y capacidad de tiempo real.

	Clase de Tiempo Real	Latencia [ms]	PER	Tiempo de ciclo [ms]	Número de nodos
Condition Monitoring	Sin RT	> 100 ms	$< 10^{-3}$	100 – 10000 ms	100 - 1000
	RT Suave	10 - 100 ms	$< 10^{-4}$		
Process Automation	RT Suave	10 - 100 ms	$< 10^{-5}$	1 - 5000 ms	100 - 300
	RT Fuerte	1- 10 ms	$10^{-9} - 10^{-8}$		
Factory Automation	RT Fuerte	1- 10 ms	$10^{-9} - 10^{-8}$	0,25 - 30 ms	2 - 50
	Isócrono	< 1 ms	$< 10^{-9}$		

Tabla 1. Requisitos de funcionamiento para servicios industriales

Condition Monitoring hace referencia a las aplicaciones industriales en las que se monitoriza el estado de los elementos mecánicos, neumáticos o hidráulicos mediante sensorica distribuida (temperatura, corriente, vibraciones). El carácter de tiempo real pasa a un segundo plano en estos servicios, aunque se requiere sincronismo entre los nodos para asegurar un flujo periódico de datos. [19] ZigBee y Bluetooth con las tecnologías más usadas en este campo. ZigBee está basada en IEEE 802.15.4 como PHY e CSMA/CA como MAC con lo que no se puede asegurar determinismo debido a las colisiones.

Process Automation engloba todos los procesos de producción continuos de circuito cerrado. En ellos, grandes cantidades de un producto se producen o procesan. Estos son relativamente lentos y continuos, como fluidos, cables o tejidos. Para asegurar

la calidad, se obtienen y transmiten multitud de datos. Por ello, es necesario sistemas de comunicación con altas tasas de datos. El sistema WirelessHART [19] es típico en este campo, también basado en 802.15.4 y haciendo uso de esquema de acceso TDMA. Este último es garantía de determinismo, pero presenta una insuficiente fiabilidad y baja tasa de datos.

Factory Automation describe los procesos discretos de producción en circuito cerrado. En ellos los productos son manufacturados, ensamblados, testados, embalados, etc. Se hacen uso de sensores y actuadores que trabajan con movimientos cortos y rápidos en un espacio limitado [4]. Requieren de control en tiempo real y son las aplicaciones más críticas en las que interviene el protocolo de comunicación. Centran los mayores esfuerzos de la comunidad científica debido a los requisitos tan estrictos y la gran diferencia entre casos de aplicación. Entre las propuestas que se presentan actualmente encontramos la de IsoMAC, [19] basado en 802.11g y TDMA o la novedosa SHARP (Synchronous and Hybrid Architecture for Real-Time Performance in IWSAN) basada en 802.11g OFDM (Orthogonal Frequency Division Multiplexing) y TDMA.

En este documento, se trabaja sobre una propuesta que nace de la combinación de las capas PHY/MAC actuales de 802.11 con la incorporación de la técnica NOMA de multiplexación no ortogonal. Esta técnica permite compartir un canal (en contraposición con FDMA) y enviar varios flujos de datos simultáneamente [3] (sin necesidad de esquema TDMA). Cada servicio se asigna a una porción de la potencia total transmitida, donde la capa superior más robusta proporciona bajas tasas de error y de datos y la capa inferior menos robusta experimentará más propensión a fallos y también más bitrate.

6) OBJETIVOS Y ALCANCE

En este apartado, se resumen el objetivo principal y los secundarios, así como el alcance de este trabajo.

6.1) Objetivos

El objetivo principal de este proyecto es diseñar y elaborar una propuesta para el mecanismo de retransmisiones dentro del trabajo de NOMA con 802.11g/n. Mediante implementación software, haciendo uso del entorno de simulación OMNeT++, se pretende ampliar las prestaciones del modelo NOMA de comunicaciones industriales. En concreto, a raíz de este trabajo, se busca analizar la mejora de fiabilidad obtenida para satisfacer los criterios de la industria y estudiar las contrapartidas que subyacen de la inclusión de retransmisiones.

Para alcanzar el objetivo principal, se han definido varios objetivos secundarios que se detallan a continuación:

- Analizar y comprender el estado del arte correspondiente a la tecnología NOMA, con especial atención a su implementación con IEEE 802.11g/n. Estudiar los requisitos técnico del sector industrial en términos de fiabilidad, robustez y latencia. Caracterizar las soluciones y estudios previos acerca de los modelos de retransmisiones y de capa MAC en entornos industriales. Elaborar una descripción sobre las contribuciones que se realizarán mediante el trabajo.
- Aprendizaje del uso de la herramienta de modelado de red OMNet++ e interpretación del modelo del sistema NOMA previo sobre el que se basará el trabajo. Adquisición de conocimientos en lenguaje C++.
- Definir las propuestas MAC para la inclusión de retransmisiones, valorando su complejidad y estimando las mejoras a obtener.
- Integrar la solución propuesta en el programa, respetando el modelo previo e introduciendo los mecanismos para analizar los resultados a nivel MAC.
- Realizar las simulaciones sobre el modelo a evaluar, recuperando resultados mediante diferentes casos. Se precisará de la obtención de datos con vistas a una posible validación para la arquitectura PHY-MAC del modelo NOMA.
- Sintetizar los resultados y obtener conclusiones, atendiendo a la comparación de los parámetros obtenidos con los requisitos de la industria y argumentando la utilidad de la propuesta. Se tomarán en cuenta las implicaciones favorables y desfavorables.

6.2) Alcance

El alcance final de este proyecto es, en primer lugar, el desarrollo de un mecanismo que aumente la fiabilidad del sistema NOMA de comunicaciones industriales. Para ello, se ha optado por la aportación a la capa MAC rudimentaria de base un nuevo sistema de retransmisiones. Con ello, se sentarán las bases para extender este estudio en siguientes trabajos con la intención de aproximar lo máximo posible el modelo a situaciones reales. Por ejemplo, una vertiente interesante sería incluir el enlace ascendente en las simulaciones, con sus consiguientes retransmisiones.

Así pues, se detallará el proceso de desarrollo realizado con la herramienta OMNet++ tanto del modelo de referencia *NOMA-based 802.11g/n* como del código de las aportaciones propuestas. Además, se considerará un caso práctico y se analizarán los resultados obtenidos para evaluar la idoneidad de la solución planteada. Se centrarán esfuerzos en analizar el comportamiento a nivel MAC del diseño en un entorno de simulación industrial.

7) BENEFICIOS

Las comunicaciones industriales se encuentran en un periodo de transición hacia un modelo de conexiones inalámbrico. La falta de soluciones que cumplan con los estrictos requisitos del sector ha propiciado que este tipo de tecnologías se implanten en entornos industriales. Con ese fin, la comunidad científica está desarrollando tecnologías que mejoren las prestaciones de los estándares inalámbricos cotidianos. En ese aspecto, este proyecto se encuadra en la búsqueda de mayor fiabilidad en las comunicaciones.

En esta sección, se describirán los beneficios que aportaría este proyecto desde el punto de vista técnico, económico y social.

7.1) Beneficios técnicos

Algunos de los beneficios más importantes que ofrecen las tecnologías inalámbricas en el seno de la industria son la movilidad, flexibilidad y escalabilidad que aportan. En el contexto de la Industria 4.0, estos dos aspectos cobran cada vez más relevancia. Hasta el momento, las instalaciones cableadas suponían un problema a la hora de manejar elementos móviles: la exposición de los cables a altas cargas de trabajo los hacía vulnerables a fallos y averías. La utilización de modelos inalámbricos permitirá sobreponer estos inconvenientes y aportará sistemas de comunicación en escenarios donde la utilización de cableado sea difícil, como en altas temperaturas o alturas.

El principal beneficio técnico que se persigue en este proyecto es el aumento de la fiabilidad del sistema *NOMA-based 802.11g/n* de comunicaciones industriales. Con ello, se pretende que esta propuesta se sitúe más cerca de cumplir los requisitos de PER más restrictivos (los de FA), y por consiguiente de convertirse en un modelo capaz de superar las pruebas de validación y estandarización.

En relación al uso de tecnología NOMA, en especial de su vertiente LDM (Layered Division Multiplexing), se consiguen mejorar [3] los resultados de otros esquemas de asignación de recursos como TDM y FDM en cuanto a robustez y fiabilidad, así como permitir una mayor eficiencia espectral. El uso de IEEE 802.11 permite la interoperabilidad con la estructura de protocolos de Ethernet, lo que facilitará su implementación en entornos WLAN.

7.2) Beneficios económicos

El reemplazo de los medios cableados por enlaces radio, debido a su fácil diseño, instalación y mantenimiento, acarreará un abaratamiento de costes a corto plazo. En las soluciones cableadas, [5] los elementos distribuidos deben estar conectados físicamente en todo momento, lo que aumenta los costes de material y de aislamiento de los equipos de comunicación

Con este proyecto, se da cabida a una nueva tecnología inalámbrica, propiciando con ello una reducción de costes en las comunicaciones industriales, otro de los pilares de la Industria 4.0. Además, fomentando una mayor escalabilidad, se consigue abaratar costes a la hora de ampliar o reformar una cadena de producción: en vez de desechar o cambiar el equipamiento antiguo, simplemente es necesario añadir nuevos nodos inalámbricos con nuevas funciones.

7.3) Beneficios sociales

Entre los beneficios sociales [9] que se consiguen en este proyecto, el más significativo es el concerniente a la seguridad en los entornos industriales. Teniendo en cuenta el carácter crítico de los mensajes que deben enviarse, como alarmas o señales de emergencia, que suponen un peligro en caso de no llegar a su destino, el aumento de la fiabilidad en las comunicaciones propiciará la creación de entornos de trabajo más seguros. Además, puede que en algunas aplicaciones se evite la necesidad de manipulación humana peligrosa en situaciones de difícil acceso o en presencia de sustancias peligrosas.

Otro de los apartados de la tecnología inalámbrica que aportan beneficios en este sentido es el aumento de la movilidad: los fabricantes estarán en disposición de utilizar cadenas de producción más complejas y eficientes, pudiendo mejorar sus perspectivas de negocio y de contratación.

8) ESTADO DEL ARTE

En este apartado se describen las principales áreas de conocimiento sobre las que reposa el trabajo. En primer lugar, se detallan los dos principales elementos que conforman la propuesta de sistema de comunicaciones sobre el que se basa el proyecto: el estándar 802.11 y la técnica NOMA. Seguidamente, se explicará otra novedosa tecnología, SHARP (Synchronous and Hybrid Architecture for Real-Time Performance), una propuesta del grupo IK4-Ikerlan que rediseña el estándar 802.11g para adaptarlo a las IWSAN (Industrial Wireless Sensor and Actuator Networks).

8.1) Non-Orthogonal Multiple Access (NOMA)

La tecnología NOMA (Non-Orthogonal Multiple Access) hace referencia a un mecanismo de superposición de espectro por el cual se consiguen transmitir múltiples flujos de datos con diferentes robusteces, capacidades, servicios y entornos de recepción por un mismo canal. Esto permite una mejora en el uso del espectro y en la capacidad de los sistemas de radiodifusión [7].

Dado que el espectro se comparte en tiempo y frecuencia entre los diferentes servicios, el 100% del ancho de banda y el 100% del tiempo se destinan únicamente a transmitir la señal. Esto contribuye a un uso más eficiente y flexible del espectro: es el concepto de reutilización jerárquica del espectro.

En recepción, se utilizan técnicas de cancelación de señal [8] para recuperar los flujos de datos, llamados capas, que conforman la señal total. Para ello, se adquiere primero la capa superior, más robusta, la cual se resta a la señal total recibida, cancelándola. Así, se puede proceder a decodificar la capa inferior ya que es la capa que se ha mantenido después de la cancelación.

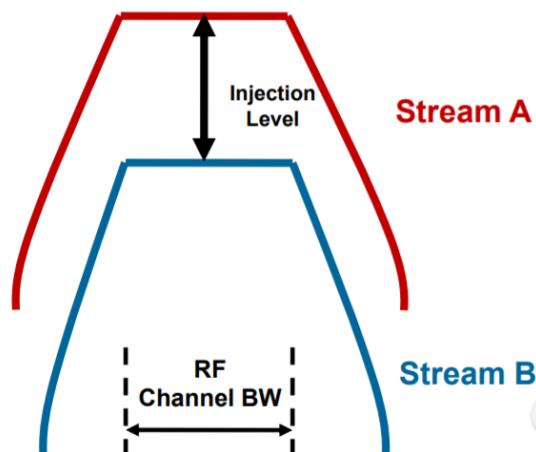


Figura 1. Esquema de multiplexación multicapa para dos servicios

Cada una de las capas viene definida por diferentes niveles de robustez [10], capacidades y umbrales de decodificación como resultado del uso de diferentes modulaciones, códigos y potencias destinadas a cada capa. Esta distribución de potencia viene definida por el nivel de inyección Δ (medido en dB). Este parámetro describe la relación en la asignación de potencia entre dos capas.

La señal resultante viene definida en el dominio de la frecuencia por la siguiente expresión:

$$x_{NOMA}(k) = x_{s1}(k) + g \cdot x_{s2}(k)$$

Dónde $x_{s1}(k)$ y $x_{s2}(k)$ son los flujos de datos que se combinan en la señal resultante $x_{NOMA}(k)$. El parámetro k define el índice del canal y el parámetro g la desviación lineal de potencia entre dos capas ($\Delta=10 \log(g)$).

Los principales requisitos [6] de esta tecnología son:

- Un potente código de prevención y corrección de errores para dotar a la capa superior de una SNR (Signal-to-Noise Ratio) negativa, cercana al límite de Shannon. En este aspecto la aparición de códigos LDPC (Low Density Parity Check) han permitido la implantación de NOMA en equipamiento real.
- Un esquema de cancelación de señal eficiente y simple, que minimice los errores en la cancelación para que sean viables altas tasas de datos en la capa inferior. Esto implica baja complejidad en las estimaciones del canal y algoritmos de equalización para simplificar la complejidad del receptor.

Estudios como [7], presentan una propuesta de la utilización de tecnología NOMA como capa física para el estándar de nueva generación ATSC 3.0 norteamericano, denominado LDM (Layered Division Multiplexing). En él, se pretendía elaborar un sistema de difusión de servicios de alta capacidad (recepción fija de TV Digital) y de baja (servicios móviles de recepción). Se demostraba que LDM proporcionaba una mejora entre 4 y 8 dB en términos de capacidad respecto a TDM. Esta mejora se hacía más evidente cuando los umbrales de recepción de los servicios fijos y móviles estaban más separados el uno del otro.

LDM vs TDM: MOBILE SERVICE GAIN (AWGN)			
	50%	33.3%	25%
3.1 Mbps	0.8 dB	4.1 dB	5.7 dB
5.46 Mbps	2.0 dB	6.2 dB	9.3 dB
LDM vs TDM: HIGH-CAPACITY GAIN (AWGN)			
	50%	67.7%	75%
17.5 Mbps	3.4 dB	- 0.9 dB	- 2.4 dB
26.3 Mbps	N/A	1 dB	- 1.2 dB
24.6 Mbps	N/A	N/A	- 0.3 dB

Tabla 2. Ganancia relativa de LDM sobre TDM en recepción móvil y fija

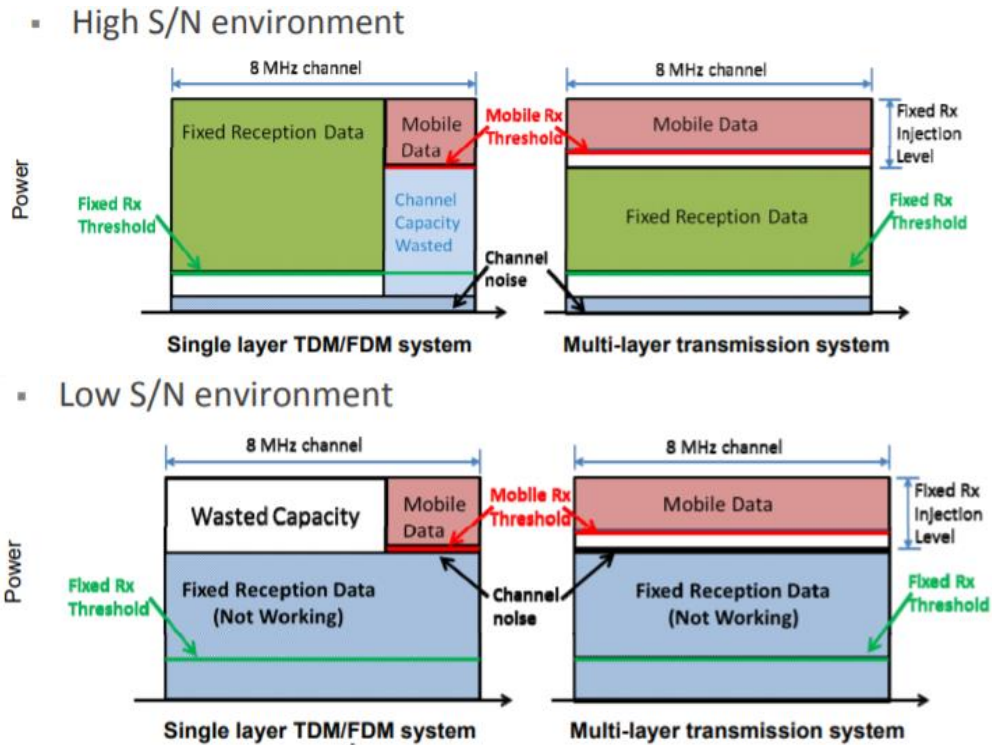


Figura 2. Esquema de funcionamiento LDM

En la misma línea, estudios que comparan la eficiencia espectral de NOMA/LDM respecto a los esquemas tradicionales TDM/FDM desde un punto de vista teórico [3] demuestran que NOMA proporciona mejores resultados cuanto mayor sea la diferencia en los requisitos de bitrate entre servicios. La asimetría en los umbrales de recepción hace que NOMA supere a TDM/FDM siempre que se consideren códigos FEC (Forward Error Coding) eficientes. Los resultados que se arrojan, por ejemplo, hablan de una mejora de 0.1 bps/Hz para casos donde la SNR del servicio móvil es de 1 dB y la del servicio fijo es de 4 dB. Sin embargo, en casos más asimétricos (1 dB, 13 dB) la mejora ofrecida por NOMA/LDM puede llegar a 2.5 bps/Hz [7].

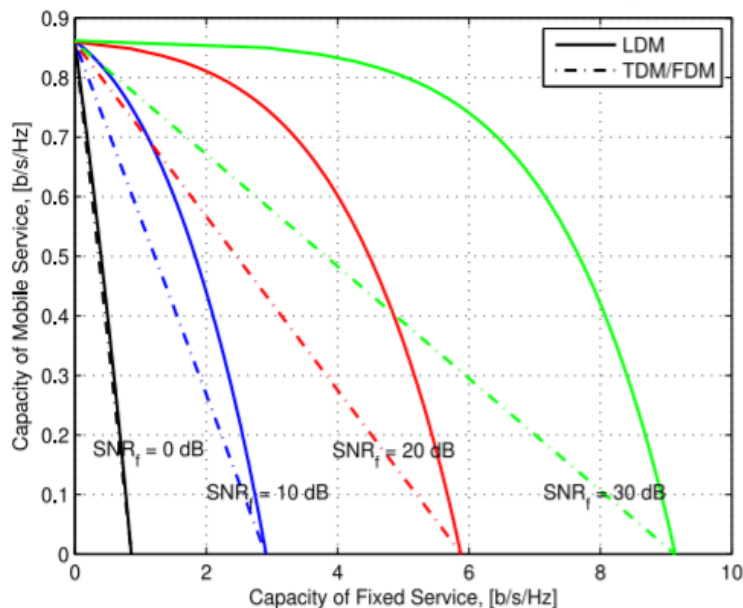


Figura 3. Comparativa de eficiencia espectral NOMA vs OMA para $SNR_m = 0$ dB

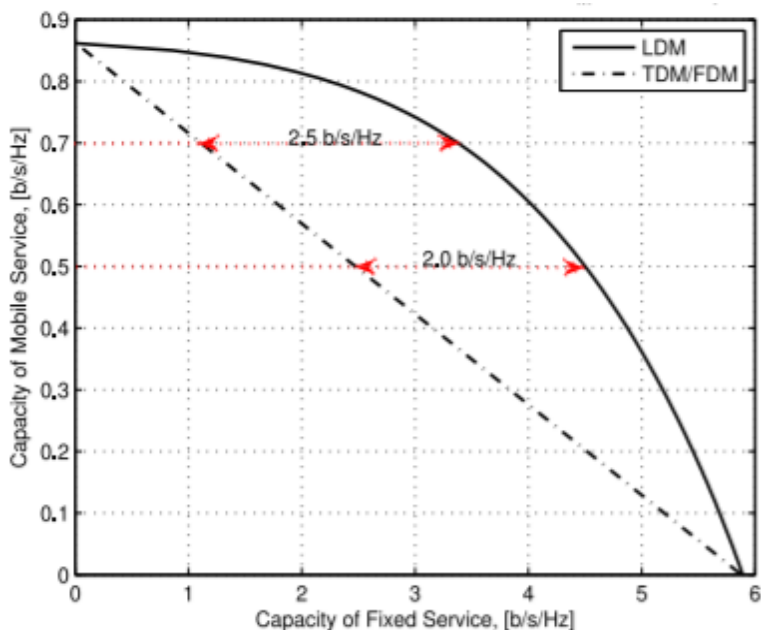


Figura 4. Ganancia relativa máxima LDM sobre TDM/FDM

Por ello, la utilización de NOMA en aplicaciones industriales se presenta como adecuada: en un mismo entorno se deben proveer servicios de altos requerimientos, en tiempo real y altamente robustos, con otros menos críticos. La fiabilidad del sistema y la seguridad en la comunicación son fundamentales en los servicios en tiempo real, algo que se aborda en NOMA. Con LDM [1], por ejemplo, se consiguen umbrales de recepción de los servicios móviles de 0 dB o incluso negativos. Además, LDM también contribuye a dotar de flexibilidad a la red: puede usarse con TDM y variando el nivel de inyección permite asignar el nivel potencia de cada servicio (un nivel de inyección de 6 dB reparte la potencia transmitida en un 80% para RT y 20% para BE).

Sin embargo, la cancelación asociada a la decodificación de la capa inferior entraña un procesamiento en el extremo del receptor que haría aumentar la latencia. [3] Conociendo lo crítico que es éste parámetro en las comunicaciones industriales, es fundamental minimizar la complejidad de los receptores, donde se produce la detección y cancelación. En este aspecto, desde el estudio de LDM [6] se recalca que la mayor fuente de retardo se produce en la decodificación LDPC. Se comprueba cómo la complejidad de este mecanismo disminuye con la SNR obtenida en recepción: apenas 5 iteraciones son necesarias con SNR altas por encima de 7 dB, que provocan un tiempo extra asociado insignificante.

8.2) Familia IEEE 802.11

El estándar IEEE 802.11 es, junto al IEEE 802.15, una de las bases para la capa física en las propuestas de redes inalámbricas industriales.

El 802.15, definido para dar cabida a las WPAN [4] (Wireless Personal Area Networks), se postula como un gran candidato a aplicaciones industriales debido a sus bajos índices de consumo eléctrico y flexibilidad. Estudios en este sentido han

confirmado su idoneidad en escenarios de tiempo real con estrictas especificaciones de latencia para aplicaciones con bajos requisitos de tasa de datos y de cobertura.

A su vez, el 802.11 especifica el funcionamiento de las redes WLAN (Wireless Local Area Network) y también está considerado en tareas industriales. Permite mayores *throughput* o caudal de datos que el 802.15 además de un mayor rango de cobertura. En 2009 [15] se estandarizó la versión IEEE 802.11n como evolución de la existente 802.11g, permitiendo más tasas de transmisión hasta los 600 Mbps en comparación con los 54 Mbps de máxima de 802.11g.

En la actualidad, no hay una clara tecnología ganadora para definir la capa PHY de las comunicaciones industriales y en cada aplicación deberá estudiarse la mejor opción.

En esta sección se describirán los dos estándares por los que se apuesta en este proyecto, el IEEE 802.11g y el IEEE 802.11n

8.2.1) IEEE 802.11g

El estándar 802.11g se popularizó rápidamente a partir de enero de 2013 [13], mucho antes de su ratificación por la IEEE, debido a la demanda de mayores velocidades y reducción de costes de producción.

El 802.11g trabaja en la banda de 2,4 GHz como 802.11b, pero permite usar un esquema OFDM (Orthogonal Frequency-Division Multiple Access) como 802.11a. Puede operar en DSSS (Espectro Ensanchado por Secuencia Directa) aunque actualmente esta configuración está en desuso para el 802.11g. Consigue unas tasas de datos máximas a nivel físico de 54 Mbps haciendo uso de códigos FEC, corrigiendo errores sin necesidad de retransmisión.

A pesar de su gran calado [5] en los dispositivos actuales, 802.11g sufre de las interferencias asociadas a una banda tan saturada como la de 2.4 GHz. Por ello, se destinan 3 canales no solapables de uso común (canales 1,6,11 con 25 MHz de separación para evitar interferencia por lóbulos laterales).

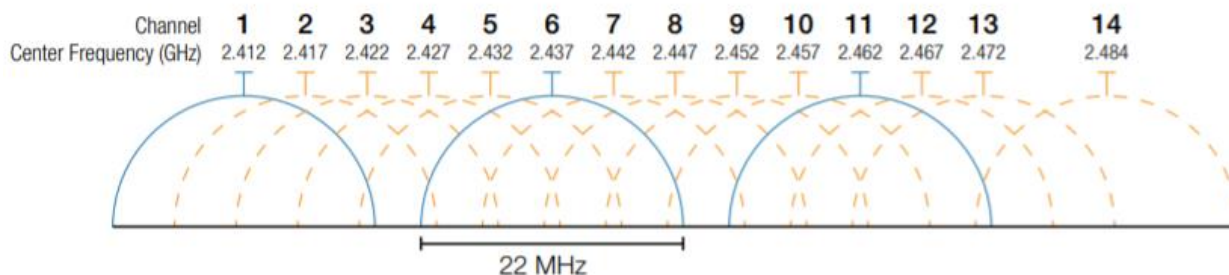


Figura 5. División de canales en la banda de 2.4 GHz con canales no solapables

La banda de 2,4 GHz se divide en 14 canales espaciados 5 MHz entre sí, los cuales en 802.11g son de alrededor de 20 MHz cada uno, aunque también admite anchos de banda de 5 y 10 MHz.

Data Rate (Mbps)	Modulación	Code Rate	Bits por subportadora	Bits de datos por símbolo
6	BPSK	1/2	1	24
9	BPSK	3/4	1	36
12	QPSK	1/2	2	48
18	QPSK	3/4	2	72
24	16-QAM	1/2	4	96
36	16-QAM	3/4	4	144
48	64QAM	2/3	6	192
54	64-QAM	3/4	6	216

Tabla 3. MCS correspondientes a 802.11g

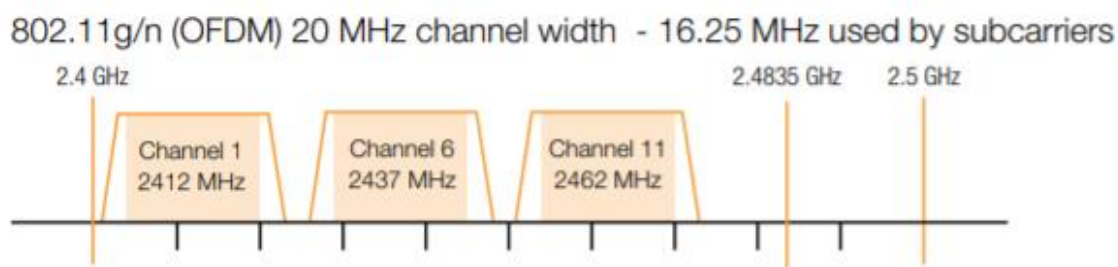


Figura 6. Canales no solapables usados en 802.11g con OFDM

En cuanto a la distribución OFDM para cada canal, se usan 64 subportadoras, 48 de datos, 4 subportadoras piloto y 12 subportadoras vacías, con una separación entre éstas de 312,5 KHz (20 MHz/64). [13] Cada subportadora puede ser modulada en BPSK, QPSK, 16-QAM o 64-QAM. El ancho de banda de cada canal es de 20 MHz con un ancho de banda efectivo de 16,25 MHz por cada portadora. Esto nos arroja una duración de símbolo de 3,2 μ s a los que se suma un intervalo de guarda de 0,8 μ s para reducir el ISI (Intersymbol interference).

La principal ventaja de la utilización de OFDM [5] sobre esquemas de portadoras únicas es la capacidad de adaptación a las condiciones del canal sin necesidad de filtros de ecualización. Para ello, se definen 8 modos de funcionamiento, denominados MCS (Modulation and Coding Scheme) en los que se garantiza una tasa de datos para una modulación y *code rate* dados. Éstas tasas de datos van desde los 6 Mbps hasta los 54 Mbps.

El formato de paquete en 802.11g se compone [13] de los siguientes elementos:

- Preámbulo: 20 μ s, 5 símbolos
 - STF (Short Training Field): 2 símbolos, sincronización inicial
 - LTF (Long Training Field): 2 símbolos, sincronización fina, estimación de canal
 - SIGNAL: 1 símbolo, 24 bits de configuración, modulación BPSK
 - RATE: 4 bits, indica MCS utilizada (modulación y FEC)
 - LENGTH: 12 bits, indica el número de bytes de carga útil
 - PARITY: 1 bit, bit de paridad par para RATE + LENGTH
 - TAIL: 7 bits, usado para decodificación FEC de SIGNAL
- Carga útil: 64 subportadoras, 48 de datos + 4 pilotos + 12 vacías, portadoras de datos moduladas en BPSK, QPSK, 16-QAM o 64-QAM (misma modulación para todos los símbolos).

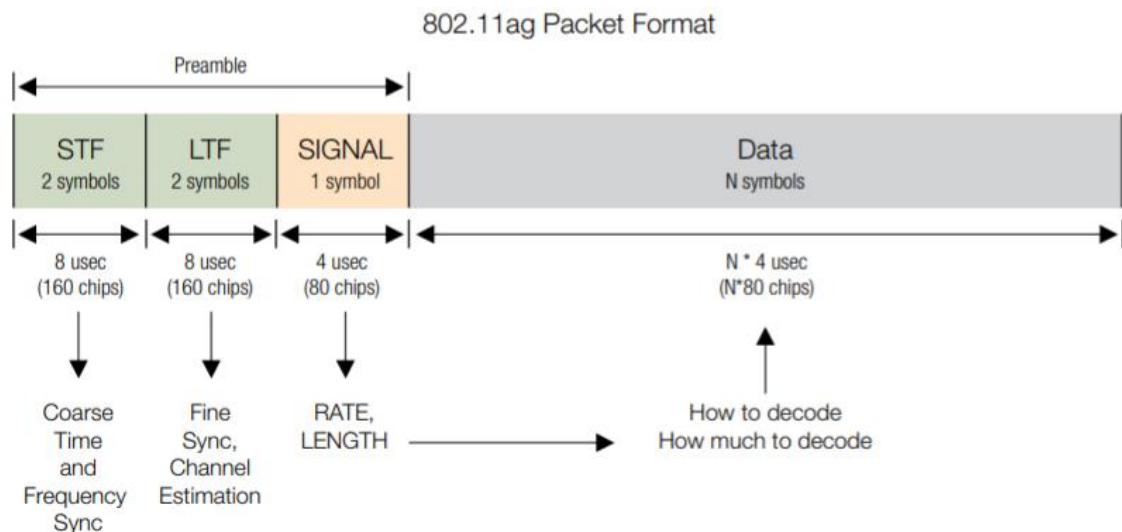


Figura 7. Formato de paquete en 802.11g/a

Como método de acceso al medio, o MAC (Medium Access Control), todos los estándares de la familia 802.11 utilizan DCF (Distributed Coordination Function) CSMA/CA (Carrier-Sense Multiple Access with Collision Avoidance). Este mecanismo consiste en comprobar que el canal no esté en uso por alguna otra estación en el momento de la transmisión [5]. Si el canal está libre, se envía el paquete. Si el canal está ocupado, la estación espera un tiempo aleatorio y vuelve a comprobar el canal. Este tiempo se denomina tiempo de *backoff*. Si el canal está libre cuando el tiempo de *backoff* finaliza, se procede a transmitir la trama. Si el canal está ocupado cuando finaliza el tiempo de *backoff*, se repite el proceso con un nuevo tiempo de *backoff* cada vez más grande.

En los resultados de [3], se comparan, para cada MCS, las curvas PER-SNR a nivel PHY que asocian la relación señal a ruido en el receptor con la probabilidad de pérdida de paquetes en las simulaciones de entorno industrial para tecnologías de una capa (no NOMA).

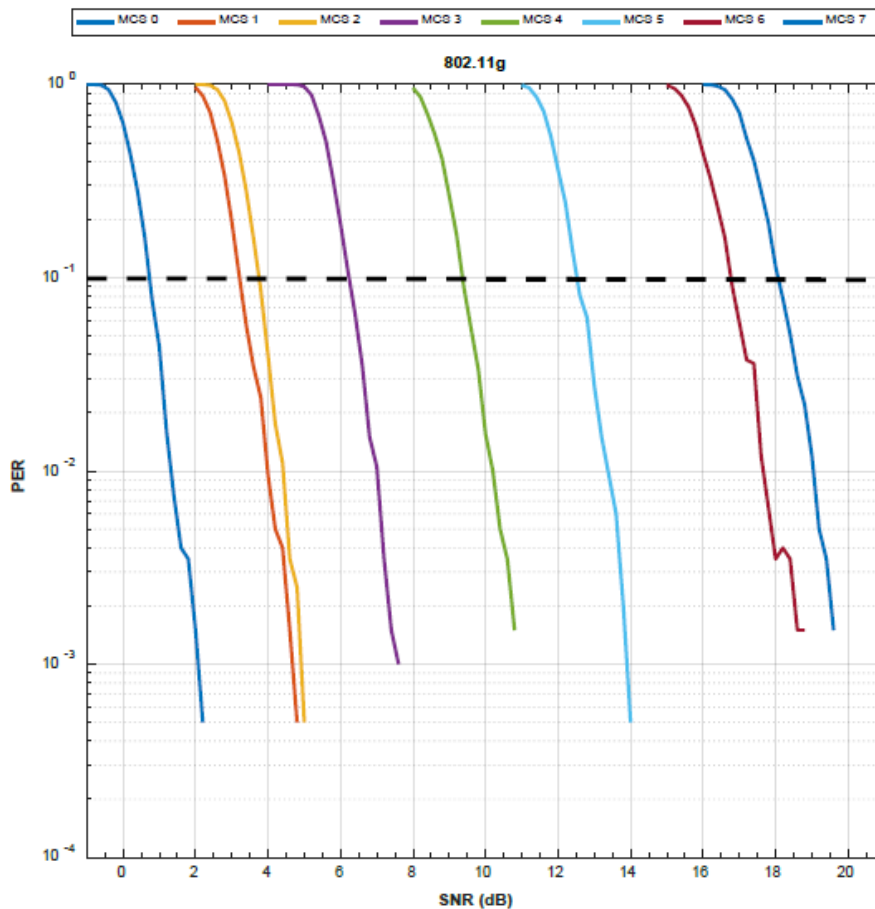


Figura 8. Curvas PER-SNR de 802.11g para cada MCS en tecnología unicapa

A partir de estos resultados [3] se comprueba que la SNR varía en un rango de 1 a 20 dB dependiendo del esquema de modulación y codificación seleccionado. A su vez, se concluye que sólo en MCS 0 se cumplen los requisitos de robustez [14] mínimos para aplicaciones industriales en tecnologías tradicionales unicapa (no NOMA).

Después, se pretende [3] evaluar la eficacia de NOMA respecto a la tecnología unicapa más utilizada, TDMA. Para ello, se evalúan unos casos de uso teóricos en los que se intentan equiparar las capacidades de los MCS de TDM a los valores de capacidad que se quieren obtener con NOMA. En concreto, se requiere un *throughput* mínimo de 6 Mbps en el Core Layer para tiempo real de NOMA y para *Best Effort* unos *throughput* de 12 a 36 Mbps en las Enhanced Layer. Por ejemplo, en un reparto TDMA 66%-33% no se pueden satisfacer los 24 Mbps de la Enhanced Layer 2 con sólo 33% del tiempo para *Best Effort* (tendría que existir una MCS que proporcionase 73 Mbps ya que $73 \times 0.33 = 24$ Mbps).

	NOMA		Real Time (66%)		Real Time (50%)		Real Time (25%)	
	Data Rate	SNR	Data Rate	SNR	Data Rate	SNR	Data Rate	SNR
Core Layer 1	6 Mbps BPSK 1/2	3.0 dB	6 Mbps BPSK 3/4	3.6 dB	6 Mbps QPSK 1/2	4.0 dB	6 Mbps 16-QAM 1/2	9.5 dB
	IL = -7 dB		Best Effort (33%)		Best Effort (50%)		Best Effort (75%)	
Enhanced Layer 1	12 Mbps QPSK 1/2	11.97 dB	12 Mbps 16-QAM 3/4	13 dB	12 Mbps 16-QAM 1/2	9.5 dB	13.5 Mbps QPSK 3/4	6.5 dB
Enhanced Layer 2	24 Mbps 16-QAM 1/2	17.54 dB	N.A N.A	N.A	24 Mbps 64-QAM 2/3	17 dB	27 Mbps 16-QAM 3/4	13 dB
Enhanced Layer 3	36 Mbps 16-QAM 3/4	20.79 dB	N.A N.A	N.A	N.A N.A	N.A	40.5 64-QAM 3/4	18.3 dB

Tabla 4. Límites teóricos SNR en 802.11g + NOMA (2 capas) vs TDM

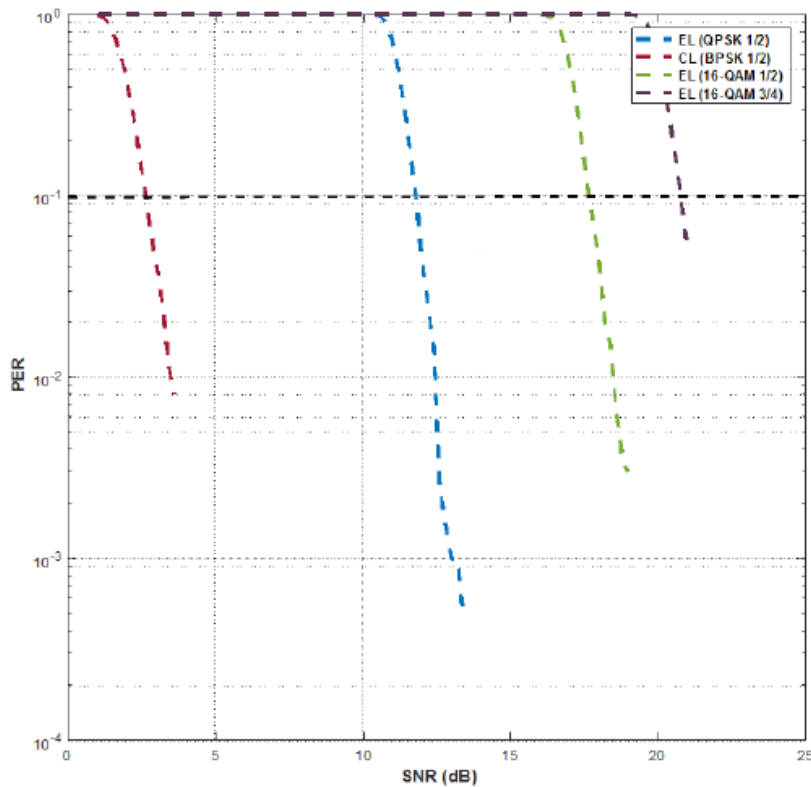


Figura 9. Curvas límite PER-SNR de 802.11g NOMA

A la luz de los resultados, se puede concluir que la solución no ortogonal proporciona mejores resultados de robustez, siendo la única capaz de garantizar SNR de 3 dB. Además, se observa que la mejora se vuelve más sustancial a medida que los servicios se vuelven más asimétricos. Este punto se vuelve determinante en el caso 75%-25%: el servicio crítico RT en TDMA deja un valor de SNR de 9.5 dB, 6.5 dB peor que con NOMA. Cabe recordar que la importancia de una mejora en la robustez del servicio RT es mayor si cabe [15], pues de la correcta transmisión de este tipo de servicios (información de sincronización, control), depende el resto de la supertrama.

8.2.2) IEEE 802.11n

El estándar 802.11n nació inicialmente como una versión del 802.11g para ofrecer mejoras en los apartados de la cobertura, fiabilidad y *throughput*. En su capa física (PHY), este estándar introduce procesado de señal y nuevas técnicas de modulación para sacar partido a canales más anchos y el uso de múltiples antenas (MIMO) en las transmisiones [13]. En la capa MAC, la incorporación de nuevas funcionalidades a protocolos existentes permiten un uso más eficiente del ancho de banda disponible. Conjuntamente, estos desarrollos HT (High Throughput) permiten incrementar la tasa de datos hasta los 600 Mbps, más de 10 veces lo que ofrecían los 54 Mbps de 802.11g.

802.11n puede operar en las bandas de 2.4 y 5 GHz, siendo el soporte para la banda de 5 GHz opcional. Si bien conserva muchas de las características de sus predecesores, como el acceso al medio CSMA/CA, el gran avance de 802.11n se consigue gracias a la técnica MIMO (Multiple-Input Multiple-Output), al posible uso de canales de 40 MHz y a la utilización de agregación de tramas en la capa MAC.

Detrás de la técnica MIMO se introduce la capacidad de recibir y transmitir simultáneamente desde múltiples antenas [16]. En 802.11n se definen varias configuraciones de antenas “M x N”, desde “1 x 1” hasta “4 x 4”. Gracias a SDM (Spatial Division Multiplexing), un mecanismo que multiplexa espacialmente flujos independientes de datos, se consiguen transmitir todos ellos en un mismo canal espectral. Esta innovación eleva por tanto el *throughput* total al disponer de un mayor número de flujos de datos en la transmisión. Si bien constituye un gran avance, en este proyecto sólo se tendrá en cuenta 802.11n bajo la configuración SISO (Single-Input Single-Output) con el fin de facilitar el análisis.

802.11n también provee la posibilidad de canales de 40 MHz, teniendo en cuenta que el AP (Access Point), permita esta funcionalidad. Canales que operan en este ancho de banda ofrecen el doble de tasas de datos a nivel físico que sus homólogos de 20 MHz y pueden trabajar en 2.4 o 5 GHz.

802.11n (OFDM) 40 MHz channel width - 33.75 MHz used by subcarriers

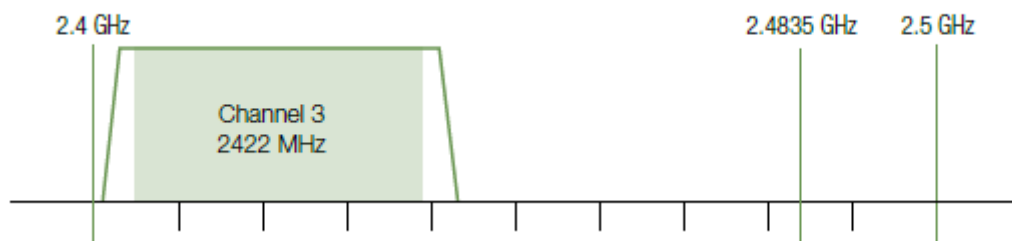


Figura 10. Canal no solapable en 802.11n con canales de 40 MHz

Nótese que un canal de 40 Mhz consume 9 canales ISM de 5 MHz: uno central y 4 a cada lado. Dado que es raro encontrar canales vacíos en esta banda 802.11n

necesita de la comunicación con dispositivos *legacy* para proporcionar mecanismos de coexistencia.

802.11n posee 2 modos de operación [13]: Greenfield (HT) y Mixto (no HT). Greenfield solamente se utiliza en entornos sin dispositivos *legacy*, mientras que el modo mixto sirve para esas situaciones. El AP en 802.11n mixto envía todas las tramas en el formato *legacy* 802.11a/g para que todos los nodos puedan entender la información, además de usar canales de 20 MHz y de no contar con las mejoras del estándar. Con ello, no se consiguen mejores prestaciones que en 802.11a/g, pero se asegura la retrocompatibilidad.

En cuanto a la distribución OFDM para cada canal [5], se usan 64 subportadoras, 52 de datos, 4 subportadoras piloto y 8 subportadoras vacías, con una separación entre éstas de 312,5 KHz (20 MHz/64). Cada subportadora puede ser modulada en BPSK, QPSK, 16-QAM o 64-QAM. El ancho de banda de cada canal es de 20 MHz con un ancho de banda efectivo de 17,8 MHz por cada portadora. Esto nos arroja una duración de símbolo de 3,2 μ s a los que se suma un intervalo de guarda de 0,8 μ s para reducir el ISI (Intersymbol interference).

Data Rate (Mbps)	Modulación	Code Rate	Bits por subportadora	Bits de datos por símbolo
6	BPSK	1/2	1	24
12	QPSK	1/2	2	48
18	QPSK	3/4	2	72
24	16-QAM	1/2	4	96
36	16-QAM	3/4	4	144
48	64-QAM	2/3	6	192
54	64QAM	3/4	6	216
60	64-QAM	5/6	6	240

Tabla 5. MCS correspondientes a 802.11n en régimen SISO y canales de 20 MHz

El formato de paquete en 802.11n para el modo mixto se compone de los siguientes elementos [13]:

- Preámbulo: 20 μ s, 5 símbolos
 - Modo Mixto *Legacy*:
 - L-STF, L-LTF, L-SIG: retrocompatibles con los estándares a/g
 - L-SIG: contiene los campos RATE y LENGTH que informan a los sistemas *legacy* cuanto deben retrasar el siguiente intento de transmisión
 - Modo Mixto HT:
 - HT-SIG (2 símbolos): contiene la MCS, length, y otros parámetros específicos de HT

- HT-STF (1 símbolo), HT-LTF (≥ 1 símbolo): permite sincronización y estimación de canal en ancho de banda HT (40 MHz, se necesitan más subportadoras). Se pueden incluir símbolos HT-LTF adicionales para casos MIMO.
- Carga Útil: 52 (20 MHz) o 110 (40 MHz) subportadoras de datos, moduladas en BPSK, QPSK, 16QAM o 64QAM, usando todos los símbolos la misma. 4 subportadoras piloto en BPSK para adaptar las variaciones de frecuencia y amplitud en la transmisión.

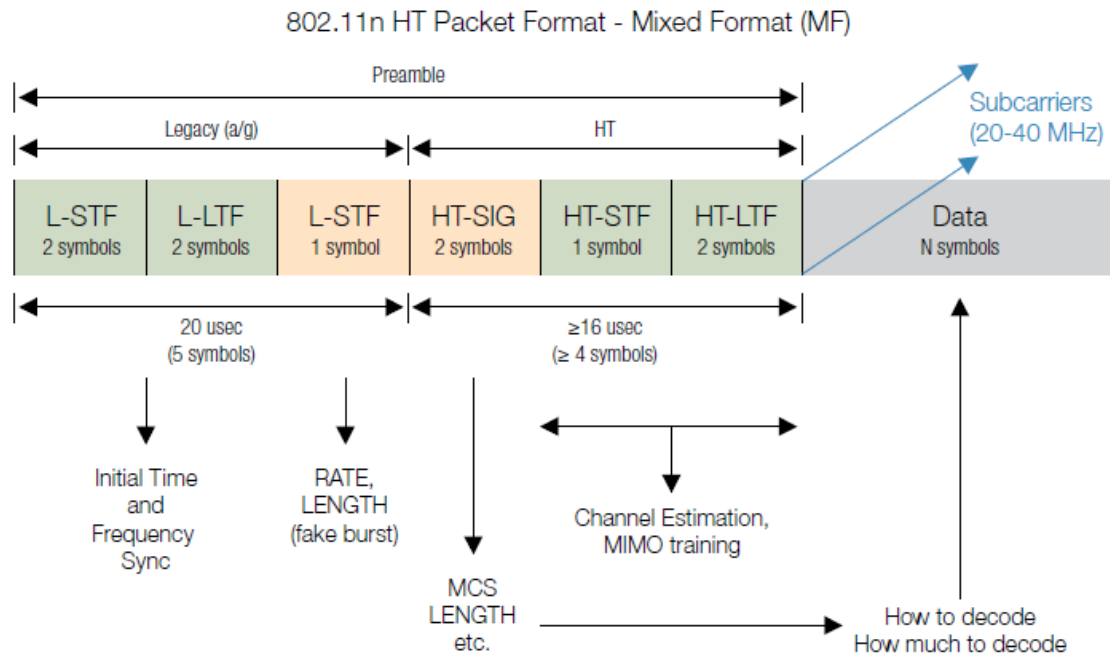


Figura 11. Formato de paquete en 802.11n modo mixto

De nuevo, en [3], se estudian las prestaciones unicapa de 802.11n para un único servicio como punto de partida para efectuar la comparativa NOMA vs TDM. En la figura 12, se recogen los límites de SNR mínimos necesarios para cada MCS tanto para un canal industrial (con multitrayectos) como para uno Gaussiano (AWGN). Además, se añade la curva de capacidad ideal de Shannon para referenciar los resultados y se fija un PER de 10^{-2} .

Se puede observar que la diferencia entre la curva de Shannon y el caso AWGN oscila entre 1 y 2 dB dependiendo de la MCS en cuestión. Esta buena prestación se logra gracias al reemplazo de códigos convolucionales por códigos LDPC en colaboración con 802.11n. Para el entorno con multitrayecto industrial, las diferencias con respecto al canal AWGN son 0.5 a 1 dB, lo que confirma que la propuesta NOMA puede implementarse en estos entornos.

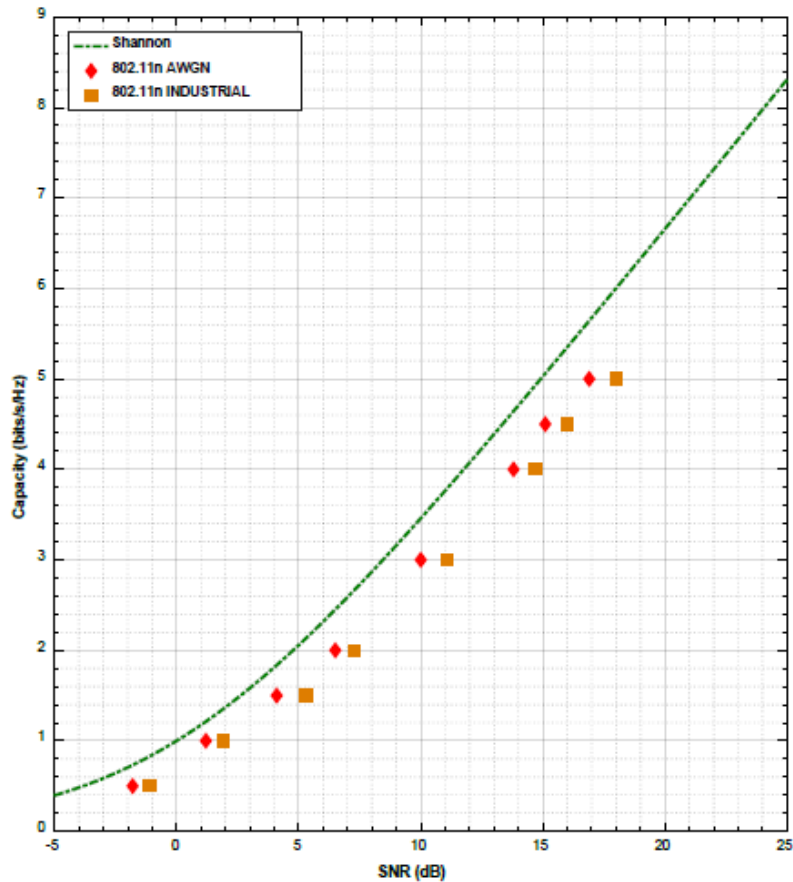


Figura 12. Límites SNR teóricos para 802.11n unicapa y PER = 10⁻²

		NOMA (IL = -4 dB)						
CASE A	MCS	AWGN	INDUSTRIAL					
CL (6 Mbps)	BPSK 1/2	1.0	1.8					
EL (12 Mbps)	QPSK 1/2	6.8	7.4					
		NOMA (IL = -6.5 dB)			TDM CL (50 %) / EL (50 %)			
CASE B1	MCS	AWGN	INDUSTRIAL	CASE A	MCS	AWGN	INDUSTRIAL	
CL (6 Mbps)	BPSK 1/2	-0.3	0.5	CL (6 Mbps)	QPSK 1/2	1.3	1.9	
EL (24 Mbps)	16QAM 1/2	14.1	14.9	EL (12 Mbps)	16QAM 1/2	6.7	7.5	
		NOMA (IL = -2.5 dB)			TDM CL (50 %) / EL (50 %)			
CASE B2	MCS	AWGN	INDUSTRIAL	CASE B1	MCS	AWGN	INDUSTRIAL	
CL (6 Mbps)	BPSK 1/2	2.2	3.2	CL (6 Mbps)	QPSK 1/2	1.3	1.9	
EL (24 Mbps)	16QAM 1/2	11.1	11.9	EL (24 Mbps)	64QAM 2/3	14.3	15.2	
		NOMA (IL = -4 dB)			TDM CL (25 %) / EL (75 %)			
CASE C	MCS	AWGN	INDUSTRIAL	CASE B1	MCS	AWGN	INDUSTRIAL	
CL (6 Mbps)	BPSK 1/2	1.0	1.5	CL (6 Mbps)	16QAM 1/2	6.7	7.5	
EL (36 Mbps)	16QAM 3/4	16.1	17.4	EL (27 Mbps)	16QAM 3/4	10.6	11.6	
		NOMA (IL = -1.5 dB)			TDM CL (33 %) / EL (66 %)			
CASE D	MCS	AWGN	INDUSTRIAL	CASE B1	MCS	AWGN	INDUSTRIAL	
CL (6 Mbps)	BPSK 1/2	3.3	3.8	CL (6 Mbps)	QPSK 3/4	4.1	5.3	
EL (48 Mbps)	16QAM 2/3	18.1	19.2	EL (36 Mbps)	64QAM 3/4	15.9	16.9	
					TDM CL (20 %) / EL (80 %)			
CASE D	MCS	AWGN	INDUSTRIAL	CASE B1	MCS	AWGN	INDUSTRIAL	
CL (6 Mbps)	BPSK 1/2	3.3	3.8	CL (6 Mbps)	16QAM 3/4	10.6	11.6	
EL (48 Mbps)	16QAM 2/3	18.1	19.2	EL (48 Mbps)	64QAM 5/6	17.6	18.8	

Tabla 6. Límites teóricos SNR NOMA 802.11n vs TDM

De igual forma, se pretende evaluar [3] la eficacia de NOMA respecto a la tecnología unicapa más utilizada, TDMA. Para ello, se evalúan de nuevo unos casos de uso teóricos en los que se intentan equiparar las capacidades de los MCS de TDM a los valores de capacidad que se quieren obtener con NOMA en este caso para 802.11n. La capa crítica, CL o Core Layer, se fija en 6 Mbps como mínimo para garantizar un ancho de banda en una agrupación de sensores industriales [1].

Se puede observar en primer lugar que NOMA es la única solución capaz de ofrecer SNR inferiores a 1 dB. En segundo lugar, TDM requiere aumentar la SNR del servicio crítico hasta los 10.6 dB para un *Best Effort* de 48 Mbps. Finalmente, de nuevo se concluye que la asimetría en las capacidades de los servicios juega en favor de la solución NOMA, si bien con servicios más balanceados la solución TDM es una buena alternativa (caso A).

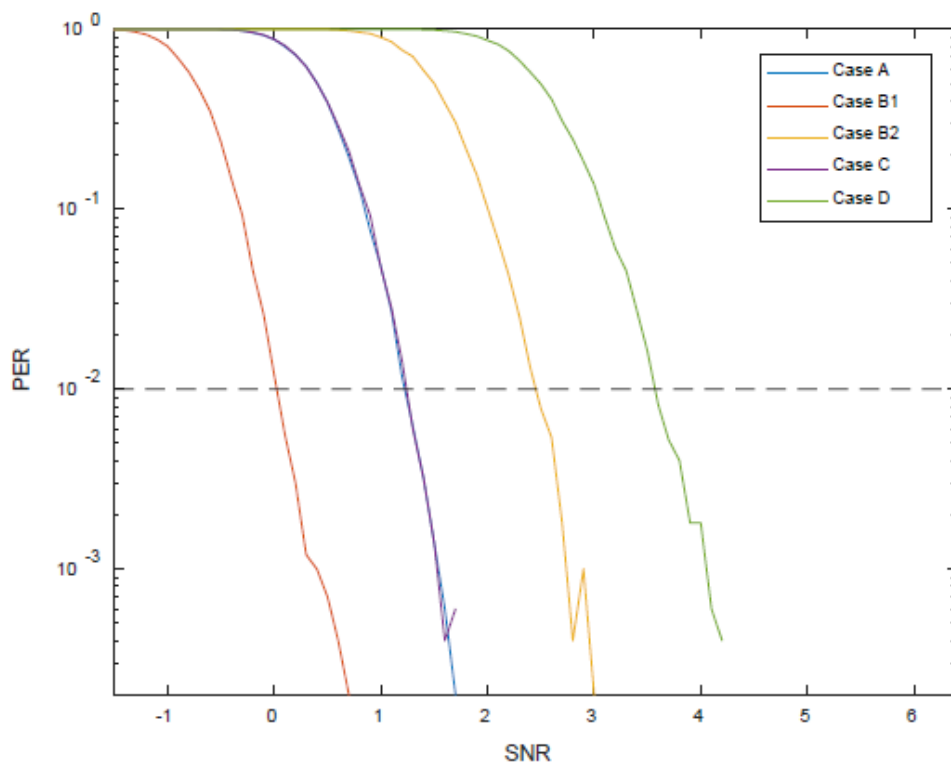


Figura 13. Curvas PER-SNR obtenidas para servicio crítico CL

Las simulaciones de la propuesta NOMA con 802.11n confirman los cálculos teóricos para PER de 10^{-2} , existiendo diferencias entre los valores previstos y los obtenidos de 0.1 a 0.3 dB. En cuanto al servicio BE, [16] NOMA ofrece peores prestaciones, con una mayor tasa de error de paquetes, pero sabiendo que este tipo de tráfico no es tan estricto y que esta desmejora es asumible si tenemos en cuenta el significativo aumento de la fiabilidad en los servicios RT.

	Case A	Case B1	Case B2	Case C	Case D
AWGN - CL	0.2	0.3	4.3	3.1	6.7
AWGN - EL	-0.1	-0.1	-1	-0.5	-0.7
IND - CL	0	2.1	4	3.8	7.2
IND - EL	-0.1	0	-0.6	-0.8	-0.7

Tabla 7. Ganancia PHY relativa NOMA vs TDM 802.11n

8.3) Synchronous and Hybrid Architecture for Real-time Performance (SHARP)

Wireless SHARP es la propuesta del grupo IK-4 Ikerlan para las IWSN. Está basada en una modificación de la capa física OFDM de 802.11g en combinación con una MAC TDMA. Sus principales ventajas son la capacidad de ofrecer servicios inalámbricos RT deterministas, con altas fiabilidades ($PER < 10^{-8}$ en canales industriales con una SNR de 23 dB) y altas tasas de paquetes (hasta 40 nodos RT).

La topología en estrella [11] del sistema SHARP se compone de un AP, como coordinador de la red, nodos RT que soportan SHARP y nodos *legacy* que se comunican mediante 802.11 estándar.

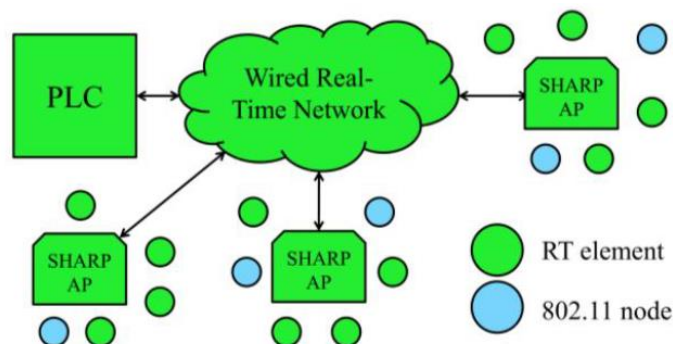


Figura 14. Topología de red SHARP

Entrando en detalle, la capa MAC de Wireless SHARP está basada en un esquema TDMA que garantiza determinismo en las comunicaciones y alta fiabilidad. La supertrama TDMA se divide en 2 grandes periodos de tiempo [12]: uno dedicado al tráfico RT y otro dedicado a servicios *legacy* para tramas 802.11. Los servicios *legacy* irán encuadrados en el periodo BE, y para garantizar que estos servicios no se den en el periodo RT se hacen uso de mecanismos de 802.11 para privar de acceso al canal a los nodos *legacy*.

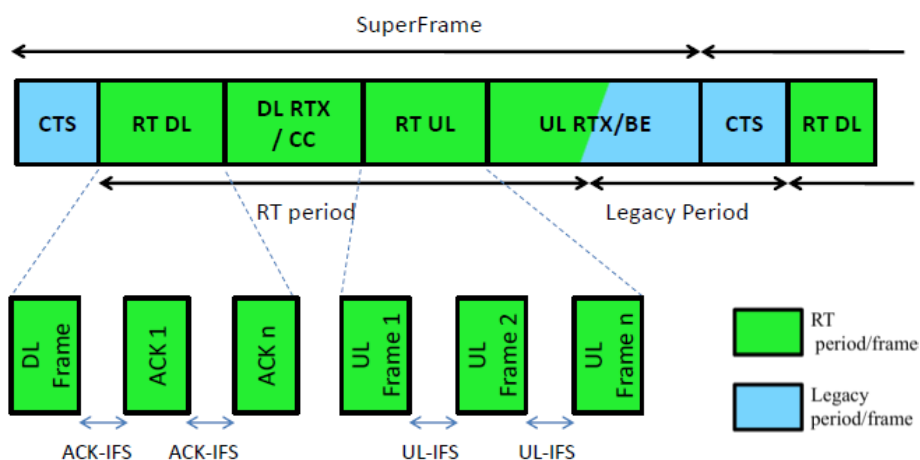


Figura 15. Capa MAC de SHARP – Supertrama TDMA

La supertrama se divide en varios *slots* temporales [12]. En el RT DL, enlace descendente RT, se transmiten los paquetes RT desde el AP hasta los esclavos. Estos paquetes se agrupan con agregación de tramas y los esclavos deben confirmar la recepción con ACK si recibieron el mensaje en el tiempo establecido por el planificador. En el DL RTX/CC se producen las posibles retransmisiones de los paquetes RT hasta que se reciban todos los ACK correspondientes. También se comprueba el estado de cada canal DL y UL para garantizar su calidad y variar si es necesario las MCS.

En el RT UL, los mensajes desde los esclavos hasta el AP se envían mediante paquetes con preámbulo acortado (sensórica sobre todo) sin necesidad de confirmación y de uno en uno con una sincronización precisa. Finalmente, en el UL RTX/BE, se producen las posibles retransmisiones de los paquetes desde los esclavos y en el tiempo restante los nodos *legacy* 802.11 pueden hacer uso del canal para transmitir sus tramas por contienda.

La capa PHY está basada en 802.11g OFDM dual para aumentar la tasa de datos: una forma de onda para el enlace descendente DL con agregación de tramas y otra para el enlace ascendente UL con preámbulo acortado [11].

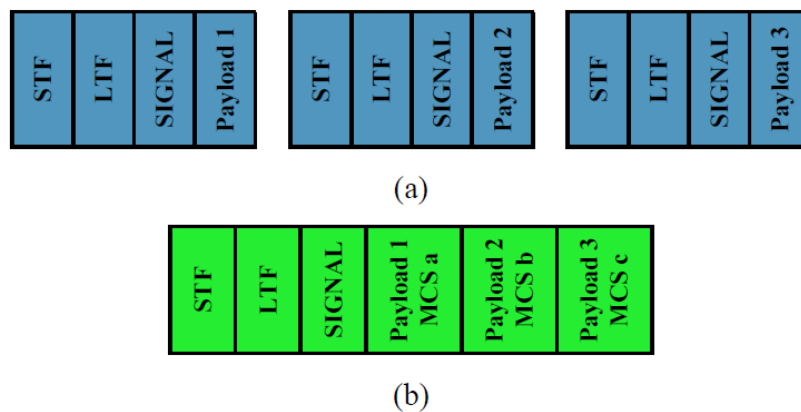


Figura 16. Tramas PHY 802.11g (a) y Agregación de tramas SHARP DL

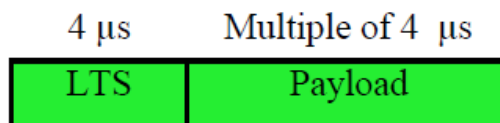


Figura 17. Formato trama SHARP UL

Esta distinción de formatos entre enlace ascendente y descendente se debe a los inconvenientes de la utilización del formato original de 802.11g para transmisiones de mensajes cortos como los usados en entornos industriales [12].

En concreto, con el formato original, se pierde eficiencia temporal puesto que en muchos casos el tiempo consumido para transmitir el paquete entero es alto comparado con el tiempo consumido en transmitir la información (pocos símbolos de carga útil).

Esta capa física dual explota por tanto la asimetría entre los canales UL y DL, la periodicidad en el tráfico RT y la poca variabilidad de los canales de comunicación

Los resultados expuestos en [12], muestran que un sistema SHARP puede ofrecer un PER de 10^{-8} bajo una SNR media de 23 dB sin el uso de retransmisiones bajo la MCS 0 de 802.11g (BPSK 1/2) para tráfico RT. Sin embargo, esta MCS se queda corta en términos de *bitrate*. Se consiguen unas latencias inferiores a 26 μ s para 10 nodos DL/10 nodos UL e inferiores a 600 μ s para 20 nodos UL/20 nodos DL. En cuanto al tráfico BE, se describe que la duración de este servicio debe ser superior a 200 μ s para asegurar el acceso al medio de las estaciones *legacy*. En escenarios industriales, normalmente con un BE inferior a 300 μ s, se obtienen *bitrates* máximos de en torno a 0.5 Mbps con MCS 7.

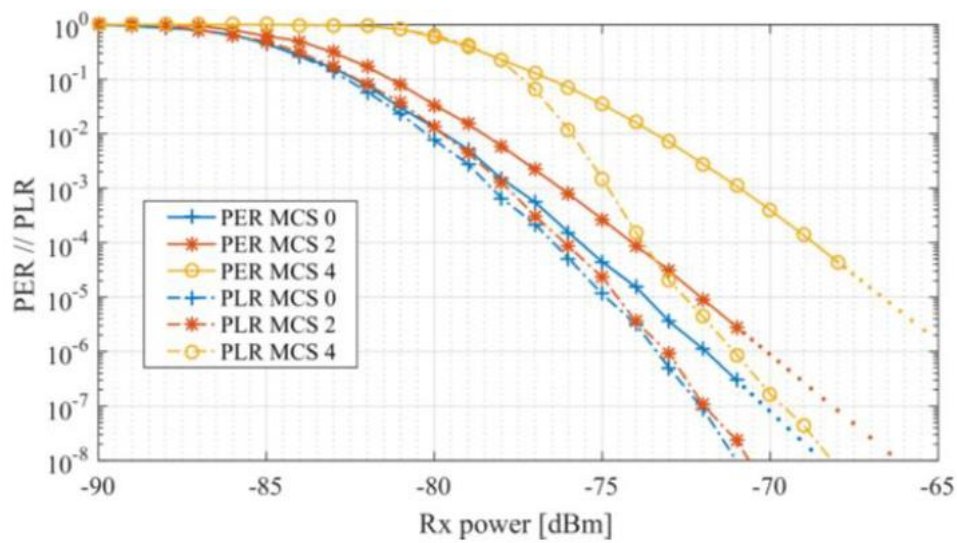


Figura 18. Resultados PER y PLR en servicio RT

		BE (μ s)						
		600	500	400	300	250	200	150
MCS	2	1.08	0.916	0.624	0.488	0.395	0.192	0
	4	1.22	1.01	0.702	0.501	0.394	0.189	0
	7	1.30	1.07	0.80	0.498	0.405	0.195	0

Tabla 8. Tasa de datos máxima alcanzable BE (Mbps)

9) METODOLOGÍA

En esta sección se desarrolla la metodología seguida en la elaboración del proyecto junto a los resultados obtenidos. Constituye la fase práctica del trabajo que tiene como objetivo estudiar una propuesta de retransmisiones para NOMA 802.11.

Para ello, en primer lugar, se describe el modelo de referencia sobre el que se trabaja, detallando el sistema puesto en marcha en OMNet++. En segundo lugar, se evalúan las propuestas MAC de retransmisiones candidatas y se justifica la elección de la ganadora. A continuación, se define el sistema de retransmisiones con la cadena completa UL-DL. Seguidamente, se describe la codificación de los módulos añadidos, tanto de la propuesta MAC como de la cadena completa. Por último, se expondrán los resultados, atendiendo al sistema de simulación y a los casos de uso diseñados.

9.1) Sistema de simulación de red en OMNet++

A fin de validar el modelo de retransmisiones que se propone, se hace uso de un simulador de red, OMNet++. Este software permitirá evaluar la robustez del modelo bajo diferentes parámetros de canal y señal. Para ello, se hace uso del sistema elaborado previamente en [3] para OMNet++.

En él, se diseña un modelo de red que hace uso de las curvas de PER-SNR expuestas en apartados anteriores para generar recepciones correctas o incorrectas de los paquetes transmitidos en la red. En concreto, usando Matlab, las curvas PER-SNR se obtienen atendiendo a los efectos del canal, características de transmisión y recepción. Estas curvas se cargan entonces en OMNet++ que las utiliza para comprobar la correcta recepción de un paquete: mediante la SNR obtenida en el receptor, se extrae desde las curvas su correspondiente PER, y si un número aleatorio creado es inferior al este PER asociado, se asumirá el paquete como erróneo. Por el contrario, si este número aleatorio es superior al PER indicado, se tratará como correcto. Cabe añadir que se trabaja sobre datos discretos de las curvas, con valores de PER para cada 0.1 dB de SNR para cada MCS (desde -1.5 dB hasta 50 dB).

El caso de simulación sobre el que se trabaja es una planta industrial. La arquitectura de red se compone de un nodo central como AP (Access Point) localizado espacialmente en el centro de la instalación y 20 nodos esclavos distribuidos aleatoriamente en cada ejecución. La distancia entre los nodos y el AP cobra especial importancia en la SNR en recepción y por ende en los niveles de PER para cada esclavo, que serán diferentes. Los esclavos pueden trabajar como receptores o transmisores, atendiendo a su uso como sensores o actuadores en las redes IWSN. El AP es el encargado de la sincronización y distribución de información a todos los nodos, mediante paquetes de 1000 Bytes. El tráfico que se modela puede clasificarse en 2 tipos:

- Real-Time (RT): tráfico con requisitos estrictos. Necesidad de alta robustez, esquemas de MCS más bajos. Alta prioridad de entrega, como alarmas, información de sincronización.
- Best Effort (BE): tráfico con más tolerancia a errores. Mayores capacidades, entregadas por las MCS elevadas. Baja prioridad, servicios complementarios, *legacy*.

Para lidiar con esta disparidad de tráfico, se utiliza las propiedades de NOMA: sobre OMNet++, se recrean las propiedades de LDM asignando la capa superior o UL (Upper Layer) a los servicios RT y la capa inferior o LL (Lower Layer) a los servicios BE. Esto se consigue en la práctica encapsulando una trama 802.11 con una robustez inferior (BE) dentro de otra con una robustez mayor (RT). En transmisión, se envían por tanto a la vez ambas capas, respetando el principio de NOMA. En recepción, se podrán recuperar las dos capas. Para ello, se hará uso del anteriormente mencionado mecanismo de la extracción del PER de las curvas y su posterior comparación. Cabe destacar que, puesto que poseen diferente robustez y *bitrate*, a cada capa le corresponde una curva PER-SNR diferente.

En cuanto a la distribución temporal, la inclusión de NOMA permite que se utilice el 100% del tiempo para la transmisión de ambos tipos de tráfico. Sin embargo, se sigue teniendo que hacer distinción temporal en cuanto al enlace descendente, desde el AP a los esclavos, y el enlace ascendente, desde los esclavos hasta el AP. Hay por tanto cuatro tipos que manejar [3]: RT-DL, BE-DL, RT-UL y BE-UL.

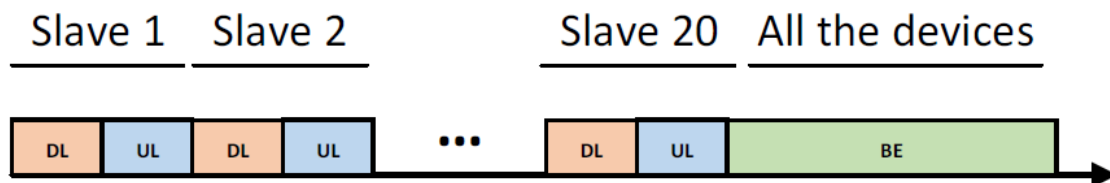


Figura 20. Asignación temporal en esquemas TDMA

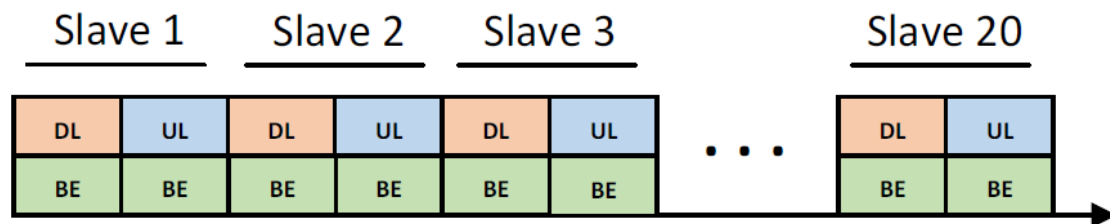


Figura 19. Asignación temporal en NOMA

Como se aprecia en la figura, en TDMA el tiempo tiene que repartirse entre los 2 tipos de tráfico ya que un *slot* sólo puede usarse para un tipo a la vez. De ahí surgen las combinaciones TDM, por ejemplo 50%-50%, donde todo el tiempo designado para RT para los 20 nodos es igual que para el tráfico BE que se deja al final. Como se ve en TDMA sólo el acceso a RT es determinista, puesto que el BE se reparte entre los esclavos por contienda. La duración de cada slot viene dada por el tiempo en el aire del paquete y varía dependiendo del modo de transmisión utilizado.

En NOMA, se reestructura todo el tráfico, convirtiendo toda la supertrama en determinista, incluso el tráfico BE, ya que se transmite conjuntamente con el RT [8]. Para cada esclavo se efectúa la comunicación completa de cada este con el AP, haciendo uso de las 2 capas. En esta técnica, la duración de cada *slot* es igual al tiempo en el aire de la capa superior y, por ello, la cantidad de datos transmitida en la capa inferior se adapta para que coincida con el tiempo en el aire de la capa superior.

En NOMA con 802.11n, que será el que se usará en este trabajo, y a modo de referencia, se define la duración de un slot como la duración de la transmisión de un bloque LDPC bajo la MCS más baja, MCS 0 de 802.11n (BPSK 1/2) [16]. Este tiempo es de 180 μ s. Por lo tanto, en la asignación temporal de NOMA, un ciclo completo de comunicación con los 20 esclavos es siempre de $20 \times 2 \times 180 \mu$ s = 7.2 ms.

En cuanto a los requisitos de tasas de datos para cada tráfico, se utilizan los casos que aparecen en 7.2.2. Se asigna al tráfico RT la MCS 0 de 802.11n, la más robusta, con lo que se asume unos 6 Mbps de *bitrate*. Para el tráfico BE, se utilizan diferentes escenarios con 12, 24, 36 y 48 Mbps.

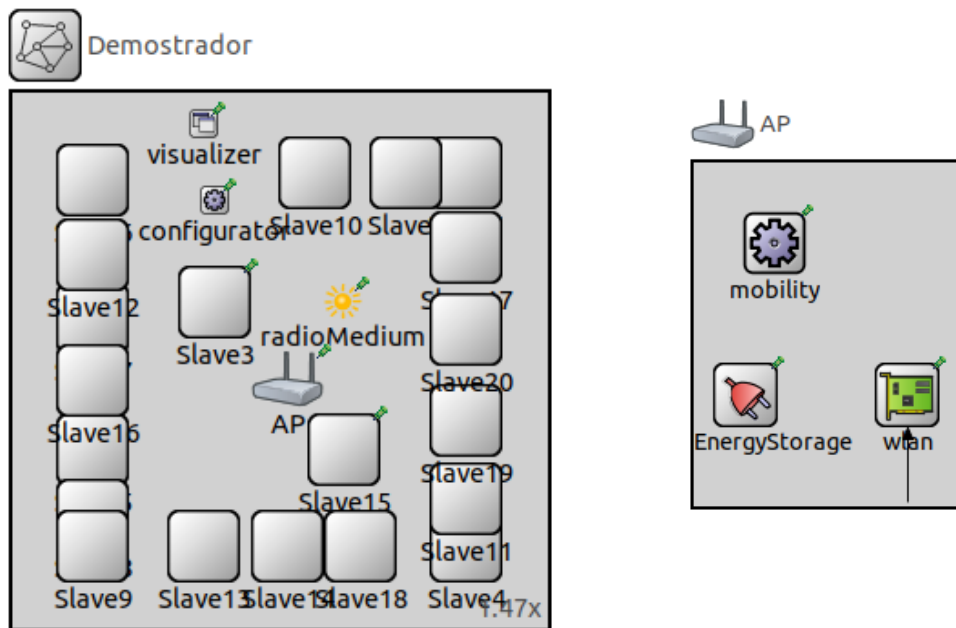


Figura 21. Esquema de la red en OMNet++ y detalle del módulo AP

En OMNet++, se trabaja sobre módulos C++/NED que se interconectan mediante puertas. Los archivos más importantes son los .ned, .cc, .h y .ini. Los archivos .ned contienen las definiciones de la red, de los módulos y de las conexiones, en lenguaje NED. Los archivos .cc contienen las funcionalidades de los módulos, en C++, donde se definen las interacciones entre elementos de red. Las variables de estos ficheros vendrán declaradas en otro fichero, los .h. Finalmente, los ficheros.ini definen los parámetros característicos de la simulación (número de simulaciones, estadísticas a recoger, valores de atributos de los módulos, etc).

9.2) Propuestas MAC de retransmisiones

La dificultad de la incorporación de un sistema de retransmisiones en cualquier capa MAC radica en la definición de la nueva asignación temporal de acceso al medio. La posibilidad de añadir a su vez confirmaciones de paquetes añade aún más complejidad a las transmisiones. Sin embargo, no es sólo en este apartado en el cual las retransmisiones influyen en la comunicación. Si bien estas permiten aumentar la fiabilidad del sistema, su inclusión tiene un fuerte impacto en la latencia y el determinismo de las llegadas de paquetes. Se tienen por tanto que realizar lo más rápido y eficazmente posible para minimizar sus efectos.

Por otro lado, existe el aspecto de la variabilidad del canal y la frecuencia en las retransmisiones. Si la nueva transmisión se realiza bajo la misma banda de frecuencia y el canal tiene una baja variabilidad temporal, es probable que el resultado sea de nuevo el mismo, habiendo desperdiciado recursos de canal para tener que volver a enviar el mensaje. En el marco de las Wireless HP [14], las usadas en ámbitos industriales, se cumple habitualmente este supuesto, en el cual se dispone de un canal aéreo muy estable en el tiempo, con dispositivos repartidos en localizaciones conocidas por el controlador. Además, una típica comunicación de control involucra paquetes de bajo tamaño, con comandos dirigidos a los esclavos desde el controlador y recolección de parámetros desde los esclavos hacia el controlador. Por ello, normalmente sólo se necesitan confirmaciones de los comandos por los esclavos, ya que se prefiere recuperar datos nuevos en el siguiente ciclo antes que retransmitir información antigua.

Desde este proyecto, se pretende abordar por tanto la implementación del modelo de retransmisiones más acorde a las características de un entorno industrial conjuntamente con NOMA. Esto incluye la elección de un candidato que permita aspirar a satisfacer los requisitos que marca la industria en términos de latencia fiabilidad y determinismo.

Categoría	Opciones	Ventajas	Desventajas	Recomendación Wireless HP
Retransmisiones	Fijas ARQ HARQ	Determinista Mejora en fiabilidad Máxima fiabilidad	Peor fiabilidad Mayor latencia Cabecera/Complejidad	Fijas y con bajo número de RTX
Confirmaciones	Sin ACK ACK Inmediato ACK de Bloque Piggyback ACK	Menor latencia Respuesta rápida Cabecera reducida Sin cabecera, menor latencia	Disminuye fiabilidad Incrementa cabecera Aumenta latencia total Paquete de respuesta necesario	Sin ACK para sensórica, piggyback ACK para control

Tabla 9. Tabla comparativa tecnologías de retransmisión

En concreto, se barajan diferentes opciones [17], entre las cuales están las retransmisiones fijas, con ARQ (Automatic Repeat Request) y con HARQ (Hybrid Automatic Repeat Request). Además, se deberá decidir si estas se producen *in-slot* o *out-of-slot*, es decir si se reserva un espacio temporal dentro de la franja de comunicación del paquete para retransmisiones o si se reserva otro espacio sólo para retransmisiones. Es el caso de SHARP, con retransmisiones para UL y DL en periodos

definidos. En lo que respecta a las confirmaciones de paquetes, existe la posibilidad de no utilizar ACK, de utilizar la técnica de ACK inmediato, de usar ACK de bloque o de usar *piggyback* ACK.

9.2.1) Modelo simple

Considerando los aspectos del entorno industrial, se estima que las retransmisiones adaptativas no son una buena opción, ya que añaden una carga excesiva a la red y al tiempo de comunicación [18]. Se concluye por tanto que un número fijo y bajo de retransmisiones [17] in-slot es la alternativa más adecuada para este tipo de escenarios. Además, se confirma que únicamente han de confirmarse los paquetes de control, mediante *piggybacking*.

En este proyecto se trabajará sobre estas bases, creando un modelo basado en la redundancia para satisfacer los niveles de PER y latencia requeridos. En concreto, se trabajará primero sobre un esquema de replicación de paquetes, donde múltiples copias del mismo paquete se envían a todos los nodos (*flooding*). Con ello, si bien consigue mejorar la fiabilidad del sistema de forma sencilla, también conlleva el aumento de la latencia. Usando un nivel de redundancia de 2, es decir el mensaje original y 2 copias, en cada nodo el mensaje puede llegar correctamente en el primer, segundo o tercer envío.

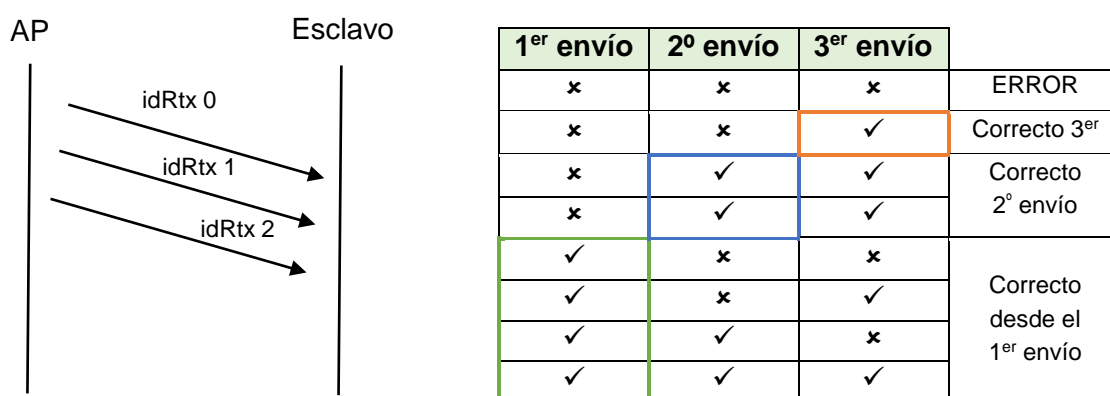


Figura 22. Esquema y descripción de casos de análisis

El análisis en este caso se centrará en obtener las sucesivas mejoras de PER que se obtienen con cada retransmisión, así como calcular la latencia. Esta última estará acotada si tenemos en cuenta la latencia como $Latencia = t_{transmisión} + t_{air}$. El tiempo de transmisión hace referencia a la duración del slot, 180 μ s, y el tiempo de propagación a la duración de la trama en el aire, que varía con cada transmisión.

9.2.2) Cadena completa

En este caso se estudia el envío de retransmisiones únicamente en el caso en el que exista un error en la transmisión. A diferencia del caso anterior, se necesita hacer uso de confirmaciones positivas y negativas desde los esclavos hacia el controlador por

lo que se requiere de un enlace ascendente de comunicación. En el caso anterior, suponíamos que el AP enviaba el mismo comando por vía aérea a todos los nodos (un ejemplo podría ser la recolección de un valor de los sensores) y no se necesitaba un mensaje de respuesta para estudiar la tasa de paquetes erróneos y latencia.

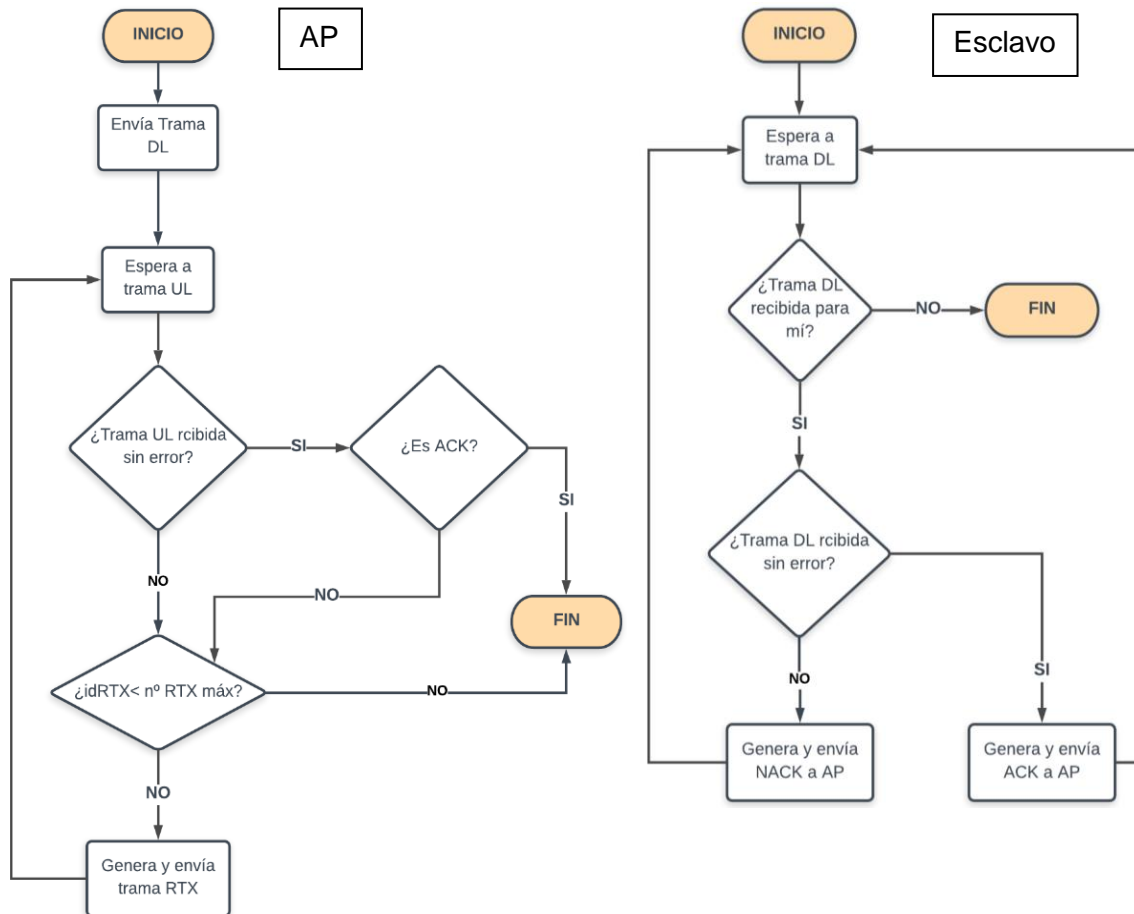


Figura 23. Diagramas de flujo del mecanismo de retransmisiones DL

En cada comunicación, el AP se dirige únicamente a un esclavo. Al tratarse de transmisión inalámbrica, la trama DL llega a todos los nodos y es necesario comprobar la dirección MAC de destino para confirmar el receptor del mensaje. Sólo el AP lleva la cuenta del número de retransmisiones que se han dado para esclavo, con lo que cada nodo únicamente genera las confirmaciones. Una comunicación entre un esclavo y el AP puede acabar por la recepción de un ACK correcto o por excederse el número de intentos de retransmisión, generando un error. Este número de intentos máximo es el mismo que en el anterior caso, admitiéndose hasta 2 retransmisiones.

Cabe recordar que se trabaja sobre NOMA, con lo que en la capa MAC es necesario administrar posibles errores en las 2 capas. Así pues, se fija que un paquete es correcto cuando la capa superior se recupera sin error, con independencia de si la capa inferior ha sufrido errores. Esto es importante, ya que influye en las condiciones de interpretación de un paquete correcto: la capa superior es más robusta, más probabilidad de envío correcto, mientras que la capa inferior requiere de una mayor SNR en el receptor. Se considera correcto un mensaje si su información RT es correcta.

9.3) Codificación de la capa MAC

9.3.1) Modelo simple

En OMNet++, los módulos interaccionan entre sí mediante señales y mensajes. Para crear un modelo de capas, se necesita hacer uso de estas herramientas para delegar tareas que no se pueden realizar en una capa concreta. En este proyecto, se trabaja sobre una capa física ya creada, 802.11n, que será transparente y de la que únicamente tendremos que proveer de los parámetros iniciales para funcionar.

En el programa, todos los dispositivos comparten una misma capa MAC, con lo que tienen que estar definidas las funciones de los esclavos y del AP, que son diferentes. Además, la capa MAC sólo recibe mensajes desde la interfaz de la capa física, ya que no se modela un nivel de aplicación por encima. Por lo tanto, hay partes del código que sólo se accederán desde los nodos y otras que sólo las activará el AP. Sin embargo, cada esclavo es un módulo independiente y las capas MAC no se comunican entre sí.

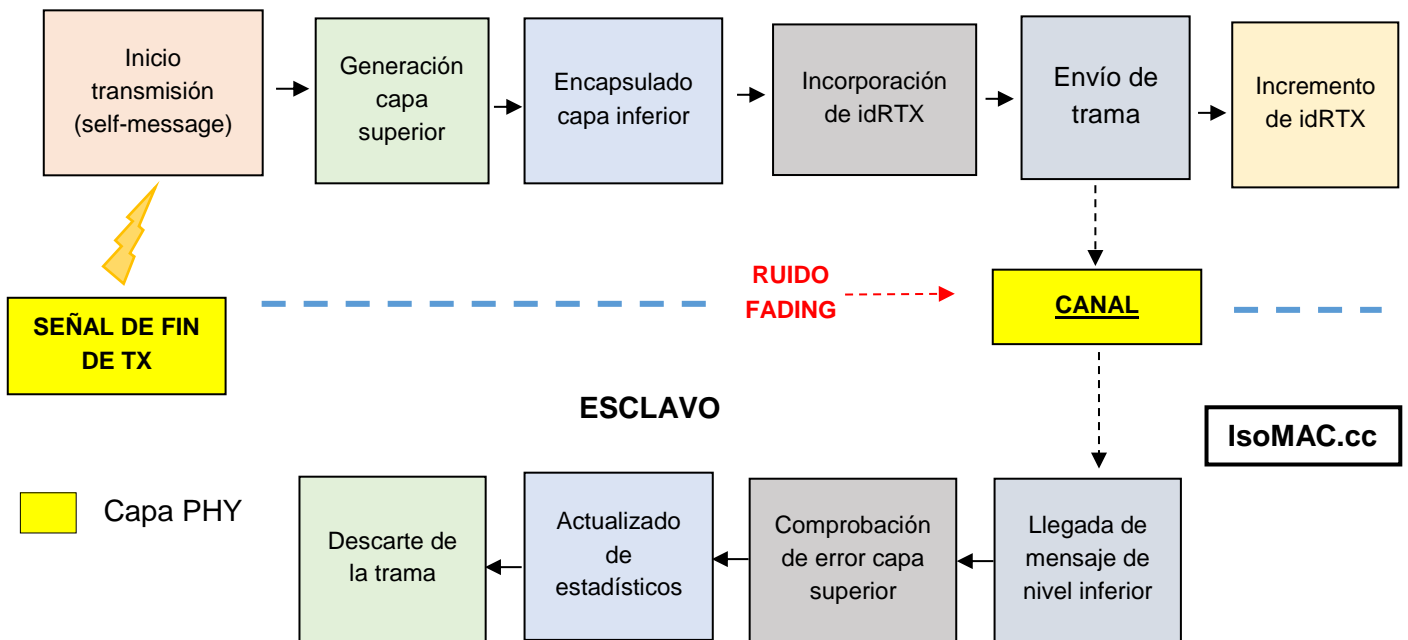


Figura 24. Esquema de funcionamiento modelo simple

En *IsoMAC.cc*, el archivo que define la MAC del programa, se sigue la cronología de la figura superior. El inicio de la transmisión se hace mediante un temporizador que genera un automensaje, un mensaje enviado al mismo módulo en el que se trabaja. Desde el AP se entra en *IsoMAC.cc*, en cada transmisión, marcando los mensajes con un identificador de retransmisión y pasándolos al nivel físico. La PHY modela la transmisión aérea del mensaje. Desde los esclavos se entra entonces a *IsoMAC.cc*, donde se comprueba el error en la capa superior de LDM. Se tiene que llevar una numeración que controle la cantidad de paquetes que han llegado correctos a la primera, segunda o tercera transmisión, con el fin de calcular los PER sucesivos:

$$PER = \frac{n^{\circ} \text{ paquetes erróneos}}{n^{\circ} \text{ paquetes totales}}$$

9.3.2) Cadena completa

En el escenario de la cadena completa, que involucra también el envío de paquetes por parte de los nodos, es importante resaltar dos aspectos. En primer lugar, ahora es necesario establecer direcciones MAC concretas en los paquetes para identificar los participantes en la comunicación. En segundo lugar, cabe recordar que todos los paquetes se envían en formato LDM, incluso las confirmaciones que producen los nodos. Por tanto, el AP deberá también comprobar el error en las capas NOMA de los paquetes ACK/NACK.

Los elementos de este código conforman otro archivo diferente a *IsoMAC.cc*, y se deberá elegir entre uno u otro funcionamiento de la capa MAC para la red. Las funciones más relevantes que se diseñan son:

- *handleLowerMsg*: es la función común para recepción de los mensajes provenientes del nivel físico. Distingue si se accede a la función como un esclavo o el AP, comprobando las direcciones MAC de destino y de origen del mensaje.
 - Por ejemplo, si el mensaje tiene como destino la dirección del AP y origen la de un esclavo, sólo se accederá a esta parte del código desde AP como receptor.
 - En ella, si se recibe un ACK correcto, se reinicializa el contador de retransmisiones y se pasa a comunicarse con el siguiente esclavo, por orden creciente de 0 a 20. Si se recibe un paquete NACK o cualquier otro paquete erróneo, se descarta el mismo y se pasa a retransmitirlo hasta un máximo de 2 veces.
 - Si se accede como el esclavo destinatario del paquete, se envía ACK o NACK dependiendo de si la trama ha sido errónea o no
- *receiveSignal*: es la función “oyente” o *listener* de las señales a las que el módulo se “suscribe”, es decir que quiere poder recibir. Es necesaria para recibir una señal en cualquier módulo. En este caso sirve para cambiar el estado del canal cuando acaba una transmisión. Permite que el módulo esté listo para recibir y quede escuchando mensajes.

Otra particularidad del modelo de comunicaciones es que no se simulan paquetes erróneos por pérdida de paquetes (temporizadores agotados). Todos los mensajes llegan a los nodos receptores y se decide si tratarlos como erróneos comparándolos con los datos de las curvas PER/SNR. La SNR recibida será proporcional a la distancia entre los dispositivos que se comunican.

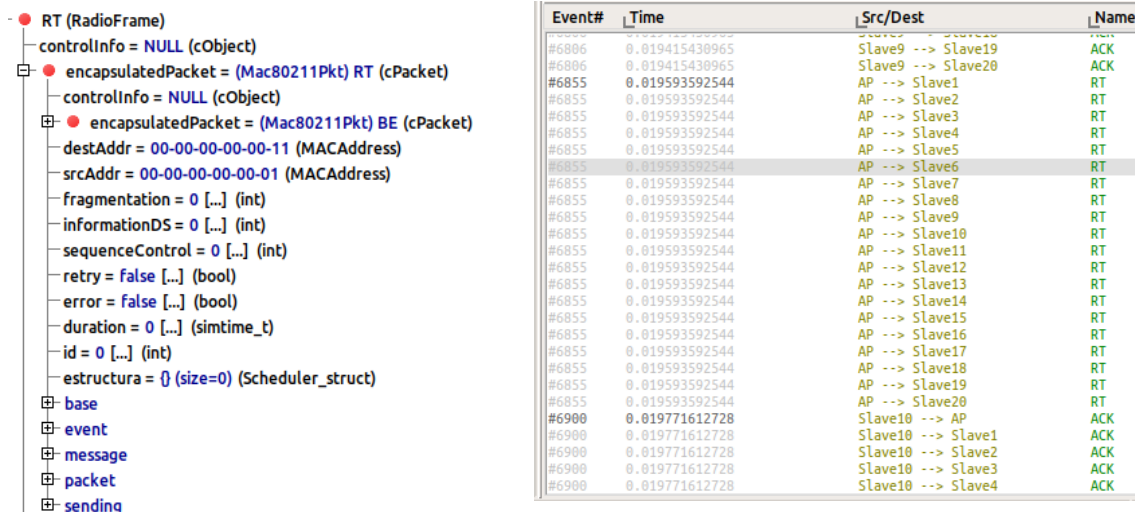


Figura 25. Detalle de la información de paquete y ejemplo de comunicación

9.4) Escenario de simulación y resultados obtenidos

9.4.1) Cadena completa

Para este caso no se evaluaron resultados de simulación. Esto se debió a falta de recursos temporales, debido a la carga computacional que requiere cada una de las simulaciones (7-12 horas por ejecución). Era inabordable desde el punto de vista de la planificación del proyecto, teniendo en cuenta la carga del modelo simple. Sin embargo, se verificó su funcionamiento y se realizaron pruebas, comprobando su posible uso futuro. El código de la implementación puede consultarse en el Anexo 1.

9.4.2) Modelo simple

En este apartado se exponen las configuraciones de simulación y los resultados obtenidos. En OMNet++, los resultados pueden ser de tipo escalar o vectorial, y se almacenan en archivo .sca o .vec respectivamente.

En este caso, se hace uso de los dos tipos. Se necesita un vector que, para cada esclavo, recoja el instante en el que cada paquete llega correctamente. Es decir, se guarda en un vector el instante de la primera llegada correcta de cada paquete que se envía a cada esclavo. Por lo tanto, existen 20 vectores con un número de datos en cada uno igual al número de paquetes diferentes que envía el AP (en total se envían 3 veces más de paquetes, las copias del original). Con ello, se consigue medir la latencia en la transmisión de un paquete

$$latencia = instante\ RX - instante\ TX$$

Como resultados escalares, se implementarán 4 contadores para cada esclavo, encargados de llevar la cuenta de los paquetes para los que la primera transmisión correcta se ha producido en 1er, 2º o 3er envío. Es decir, se quiere conocer cuántos

paquetes han necesitado de 1, 2 o 3 transmisiones para que lleguen sin error al destino. Además, existe un contador de paquetes para los que se han producido 3 errores seguidos, lo que implica que ese paquete no va poder recuperarse en recepción. Estos escalares permitirán conocer los diferentes PER que se consiguen mediante las retransmisiones sucesivas, ya que el PER también se define como:

$$PER = 1 - \frac{n^{\circ} \text{ paquetes correctos}}{n^{\circ} \text{ paquetes totales}}$$

Así pues, se plantea la realización de un barrido de los parámetros que puedan alterar los niveles de PER. Desde el archivo omnetpp.ini, que define las variables de la simulación, se pueden modificar las condiciones iniciales de la red.

Tamaño de celda de red	20 x 20 – 100 x100	
Potencia del transmisor	10 mW – 100 mW	
Tipo de canal	AWGN - INDUSTRIAL	
Semillas	5	
Casos NOMA PHY	CASE C	CASE D
	UL Bitrate = 6 Mbps LL Bitrate = 36 Mbps Injection Level = 4 dB	UL Bitrate = 6 Mbps LL Bitrate = 48 Mbps Injection Level = 1.5 dB
Tiempo de simulación total	100 s	
Modelo de Fading	Rayleigh	
Potencia de ruido	-90 dBm	

Tabla 10. Parámetros de simulación del modelo simple

En la tabla superior se detallan los parámetros que se modifican en cada ejecución de las simulaciones. De la combinación de todos los parámetros variables entre sí se obtiene la suma total de ejecuciones. En total:

$$2 \text{ tamaño celda} \times 2 \text{ TXPower} \times 2 \text{ canales} \times 5 \text{ semillas} \times 2 \text{ casos PHY} = 80 \text{ ejecuciones}$$

La elección de los parámetros se ajusta al hecho de que los resultados puedan ser referenciados a los obtenidos en [3], puesto que este trabajo se plantea como una aportación a este estudio. Por ello, los casos de NOMA son 2 de los usados en ese trabajo, elegidos porque son los que presentan una más alta disparidad entre servicios, donde NOMA es más ventajoso de utilizar. Los 2 tipos de canales AWGN (Additive White Gaussian Noise) e INDUSTRIAL permitirán comparar el rendimiento bajos 2 entornos de propagación. El número de semillas permite reposicionar los nodos y confirmar la coherencia de resultados. El área simulada será de 20x20 metros o 100x100. Se elige 100 segundos de simulación por ejecución, a un ritmo medio de 0.03 segundos de simulación por segundo natural.

En OMNet++, existe la posibilidad de automatizar las ejecuciones de los parámetros variables, es decir que el programa enumera los diferentes *runs* que se crean de la combinación de variables. Además, permite la paralelización de ejecuciones, dividiendo la carga de trabajo en diferentes procesadores. Todo esto permite agilizar las simulaciones considerablemente, aunque el proceso es largo: por ejemplo, hay parámetros “*hardcodeados*” que deben cambiarse cada vez (tamaño celda, casos NOMA).

Los resultados se resumen a continuación:

- **TAMAÑO CELDA 20x20**

Tx Power = 10 mW						
CASO C						
	Enviado Total	Rx 1 ^a	Rx 2 ^a	Rx 3 ^a	Recibido Total	Pérdida
NOMA-AWGN	185185	185073,8	111,11	0,09	185185	0 %
NOMA-IND	185185	185054,9	131,92	0,15	185185	0 %
CASO D						
NOMA-AWGN	185185	185003,52	181,2	0,28	185185	0 %
NOMA-IND	185185	184981,57	202,97	0,46	185185	0 %
Tx Power = 100 mW						
CASO C						
NOMA-AWGN	185185	185173,29	11,71	0	185 185	0 %
NOMA-IND	185185	185172,09	12,91	0	185 185	0 %
CASO D						
NOMA-AWGN	185185	185166,76	18,24	0	185185	0 %
NOMA-IND	185185	185164,56	20,44	0	185185	0 %

Tabla 11. Resultados brutos de recepción de paquetes 20x20

Tx Power = 10 mW						
	CASO C			CASO D		
	PER 1	PER 2	PER 3	PER 1	PER 2	PER 3
NOMA-AWGN	6,005E-04	4,860E-07	0	9,800E-04	1,512E-06	0
NOMA-IND	7,132E-04	8,100E-07	0	1,109E-03	2,484E-06	0
Tx Power = 100 mW						
	CASO C			CASO D		
	PER 1	PER 2	PER 3	PER 1	PER 2	PER 3
NOMA-AWGN	6,323E-05	0	0	9,850E-05	0	0
NOMA-IND	6,971E-05	0	0	1,104E-04	0	0

Tabla 12. Resultados PER 20x20

- **TAMAÑO CELDA 100x100**

Tx Power = 10 mW						
CASO C						
	Enviado Total	Rx 1 ^a	Rx 2 ^a	Rx 3 ^a	Recibido Total	Pérdida
NOMA-AWGN	185185	179155,1	5750,8	348,3	185149,1	0,019 %
NOMA-IND	185185	178023,9	6671,6	494,4	185124,4	0,033 %
CASO D						
NOMA-AWGN	185185	175456,2	8945,8	847	185184,4	0,0003%
NOMA-IND	185185	174426,5	9675,8	1023,6	18518,2	0,0004%
Tx Power = 100 mW						
CASO C						
NOMA-AWGN	185185	184569,8	624,3	6,1	185045,7	0,075 %
NOMA-IND	185185	184445,2	743,9	8,5	185013,7	0,092 %
CASO D						
NOMA-AWGN	185185	184171,1	1012,8	13,7	185184	0,0006 %
NOMA-IND	185185	184053,5	1133,3	16,1	185183,9	0,0006 %

Tabla 13. Resultados brutos de recepción de paquetes 100x100

Tx Power = 10 mW						
CASO C			CASO D			
	PER 1	PER 2	PER 3	PER 1	PER 2	PER 3
NOMA-AWGN	3,296E-02	2,026E-03	1,938E-04	5,318E-02	5,183E-03	7,520E-04
NOMA-IND	3,867E-02	2,921E-03	3,273E-04	5,810E-02	6,348E-03	9,253E-04
Tx Power = 100 mW						
CASO C			CASO D			
	PER 1	PER 2	PER 3	PER 1	PER 2	PER 3
NOMA-AWGN	3,393E-03	3,290E-05	4,320E-06	5,523E-03	7,532E-05	6,120E-06
NOMA-IND	4,045E-03	4,640E-05	3,240E-06	6,186E-03	8,887E-05	6,642E-06

Tabla 14. Resultados PER 100x100

Para comparar los resultados con los obtenidos en [3], utilizamos el escenario 100x100 y 10 mW, que es el que se usaba en ese trabajo sin las retransmisiones incorporadas.

	SIN RTX		CON RTX	
	CASO C	CASO D	CASO C	CASO D
NOMA-AWGN	3,4E-02	5,4E-02	1,938E-04	7,520E-04
NOMA-IND	4E-02	6E-02	3,273E-04	9,253E-04

Tabla 15. Comparativa PER

En cuanto a los resultados de latencia, cabe recalcar que solamente pueden darse 4 valores para cada comunicación:

- el paquete llega correctamente desde el 1^{er} envío, la latencia es 180 μ s
- llega correctamente a partir del el 2^o envío, la latencia es 360 μ s
- llega correctamente sólo en el 3^{er} envío, la latencia es 540 μ s
- no se recibe correctamente en 3 envíos sucesivos, el paquete se pierde, latencia infinita

Es importante añadir que el tiempo de propagación existe, y se pudo comprobar a partir de los resultados vectoriales que era despreciable (hablamos de recintos de 100x100 metros como máximo) y no se dan grandes distancias ni multitrayectos. De media el tiempo de propagación, simulado por la capa PHY del modelo, nunca superaba los 0,2 μ s en el caso 100x100.

Tx Power = 10 mW		
	CASO C	CASO D
Latencia Media (μ s)		
NOMA-AWGN	186,33	190,4
NOMA-IND	187,45	191,34
Tx Power = 100 mW		
	CASO C	CASO D
NOMA-AWGN	180,63	181,02
NOMA-IND	180,75	181,13

Tabla 16. Latencia media 100x100

La tabla 16 recoge la latencia media de un paquete sin error en el caso 100x100 metros. En el caso 20x20 la variación de la latencia con respecto a los 180 μ s de duración del slot es despreciable (pocos paquetes necesitan de más de 1 envío) con lo que no se incluye en el análisis. Además, aún en el caso 100x100 la variación es pequeña: tan sólo en el caso de 10 mW es de más de 5 μ s.

Cabe recordar que el criterio que se sigue para marcar como erróneo un paquete es que la capa superior sea correcta, con independencia del error en la capa inferior. Es decir, estos resultados se centran en el flujo RT. Este apunte es importante: la capa superior tiene menos exigencias de SNR al ser más robusta. Los resultados serían por lo tanto peores en cuanto a número de errores si se exige que lleguen correctamente las 2 capas (en este caso estaríamos centrándonos en el tráfico BE, el que impone el criterio de SNR más elevado).

Con los resultados obtenidos en las simulaciones realizadas pueden establecerse las siguientes conclusiones:

- La incorporación de un sistema de retransmisión de paquetes es viable dentro del modelo NOMA 802.11n para comunicaciones industriales, bien sea mediante comunicación bidireccional o por redundancia.
- Los parámetros que más influyen a la pérdida de paquetes son el tamaño del área (5 veces más grande, 7 dB) y la potencia del transmisor (10 dB).
- El canal industrial produce más errores que el canal gaussiano, debido a que tiene requisitos de umbral SNR más altos, pero la tendencia con respecto al canal gaussiano se respeta (mismo orden de magnitud). Esto se debe a que NOMA ofrece una mejora en la robustez en los servicios RT en detrimento de los BE que sufren más errores.
- Se consigue que no haya errores en el caso 20x20, todos los paquetes se reciben con las 2 potencias de transmisión. Cada retransmisión mejora en un 1 orden de magnitud de media el PER del sistema.
- Se consigue mejorar la fiabilidad del sistema en 2 órdenes de magnitud respecto al modelo de referencia, llegando al límite de los PER que se exige en comunicaciones industriales 10^{-4} , a pesar de que no es suficiente para una comunicación RT.
- La elección de retransmitir cada paquete sin esperar confirmaciones por parte de los nodos es la que va a proporcionar una mejor latencia media (más baja), siempre y cuando se den muchos errores. En nuestro caso, aunque no es la mejor opción, los peores resultados de latencia se desvían 10 μ s de la duración del slot. Aun así, están lejos los de los requisitos marcados para comunicaciones industriales (hasta 100 μ s)

10) PLANIFICACIÓN

En este apartado se detallan los integrantes del proyecto, las fases del mismo y su duración. La planificación representa una etapa fundamental a la hora de definir objetivos, fechas límite y material necesario. Es imprescindible realizar un desglose de hitos y entregables además de elaborar un diagrama de Gantt con el que situar el proyecto en su extensión temporal.

10.1) Equipo de trabajo

Nombre	Responsabilidad	Perfil
Pablo Angueira	Director del proyecto	Ingeniero senior
Eneko Iradier	Codirector del proyecto	Ingeniero senior
Mario Albisu	Realizador del proyecto	Ingeniero junior

Tabla 17. Equipo de trabajo

10.2) Fases del trabajo

PT0. Coordinación, gestión y supervisión del proyecto
El director y codirector supervisan el trabajo realizado. Comienzo: 01/11/2018 Fin: 18/02/2019 Recursos humanos: Dos ingenieros senior (40 horas) Recursos técnicos: Un ordenador con la suite Microsoft Office (10 horas) y el software OMNet ++ instalado (5 horas)

Tabla 18. PT0. Coordinación, gestión y supervisión del proyecto.

PT1. Planificación del proyecto
El director y codirector describen las pautas a seguir para la realización del proyecto. PT1.1. Definición de los objetivos y alcance. Definir los objetivos del proyecto dejando claro cuál será su finalidad. PT1.2. Definición de las fases del proyecto y fechas límite. Definir el plan de trabajo para lograr los objetivos marcados. Comienzo: 01/11/2018 Fin: 14/11/2018 Recursos humanos: Dos ingenieros senior (10 horas) y un ingeniero junior (10 horas) Recursos técnicos: Un ordenador con la suite Microsoft Office (10 horas)

Tabla 19. PT1. Planificación del proyecto.

PT2. Estudio del estado del arte
<p>Estudio y análisis de los conocimientos teóricos requeridos para la realización del proyecto.</p> <p>PT2.1. Requisitos de las redes industriales. Principales requisitos que garanticen las exigencias del sector industrial en las IWSN.</p> <p>PT2.2. Tecnología NOMA/SHARP. Estudio de las propuestas NOMA 802.11n y SHARP, atendiendo a sus contribuciones y resultados en la capa MAC</p> <p>PT2.3. Estándar 802.11g/n. Estudio de la capa física inalámbrica 802.11, funcionamiento, formato de tramas y prestaciones.</p> <p>Comienzo: 15/11/2018 Fin: 06/12/2018 Recursos humanos: Un ingeniero junior (60 horas) Recursos técnicos: Un ordenador con la suite Microsoft Office (45 horas)</p>

Tabla 20. PT2. Estudio del estado del arte

PT3. Aprendizaje del programa OMNeT++
<p>Profundización en el conocimiento del programa para su uso en la elaboración de la propuesta.</p> <p>PT3.1. Estudio del entorno de programación y del lenguaje C++. Familiarización con el programa, comandos básicos, uso de señales y mensajes.</p> <p>PT3.2. Estudio del modelo de referencia NOMA 802.11n en OMNeT++. Análisis del programa de simulaciones NOMA con 802.11n, características de la red modelada, funcionamiento de la capa MAC.</p> <p>Comienzo: 10/12/2018 Fin: 21/12/2018 Recursos humanos: Un ingeniero junior (10 horas) Recursos técnicos: Un ordenador con el software OMNeT++ (10 horas)</p>

Tabla 21. PT3. Profundización en el programa OMNeT++

PT4. Desarrollo de la aplicación
<p>Este paquete de trabajo constituye la mayor carga de trabajo en el proyecto. Engloba los pasos concernientes a la implementación del código.</p> <p>PT4.1. Elaboración y selección de propuestas MAC. Estudio de métodos de retransmisión posibles en redes inalámbricas industriales y selección de una candidata.</p> <p>PT4.2. Integración del módulo de retransmisiones en el modelo de capa MAC de referencia sobre NOMA 802.11n.</p> <p>Desarrollo e implementación del nuevo modelo sobre la base de NOMA y 802.11n. Definición de una nueva capa de acceso al medio. Nuevos indicadores de resultados.</p> <p>Comienzo: 24/12/2018 Fin: 13/01/2019 Recursos humanos: Dos ingenieros senior (30 horas) y un ingeniero junior (150 horas) Recursos técnicos: Un ordenador con el software OMNet++ (115 horas)</p>

Tabla 22. PT4. Desarrollo de la aplicación

PT5. Simulación y obtención de resultados
<p>Realización de las diferentes simulaciones para la obtención de resultados de PER y latencia, trabajando sobre barridos de parámetros en base a jerarquías.</p> <p>Comienzo: 14/01/2019 Fin: 23/01/2019 Recursos humanos: Dos ingenieros senior (10 horas) y un ingeniero junior (20 horas) Recursos técnicos: Un ordenador con el software OMNet++ (15 horas)</p>

Tabla 23. PT5. Simulación y obtención de PER y latencia

PT6. Documentación.
<p>Elaboración de la memoria del proyecto, sintetizando los resultados de las contribuciones realizadas. Redacción y revisiones finales.</p> <p>Comienzo: 18/01/2019 Fin: 18/02/2019 Recursos humanos: Un ingeniero junior (50 horas) Recursos técnicos: Un ordenador con la suite Microsoft Office (50 horas)</p>

Tabla 24. PT6. Redacción de informe

Entregables		
E1	Plazos del proyecto	14/11/2018
E2	Bibliografía referente al estado del arte	06/12/2018
E3	Código de la solución propuesta	13/01/2019
E4	Memoria final	18/02/2019

Tabla 25. Entregables

Hitos		
H1	Planificación finalizada	14/11/2018
H2	Estado del arte completado	07/12/2018
H3	Simulaciones finalizadas	23/01/2019
H4	Finalización del proyecto	18/02/2019

Tabla 26. Hitos

10.3) Diagrama de Gantt

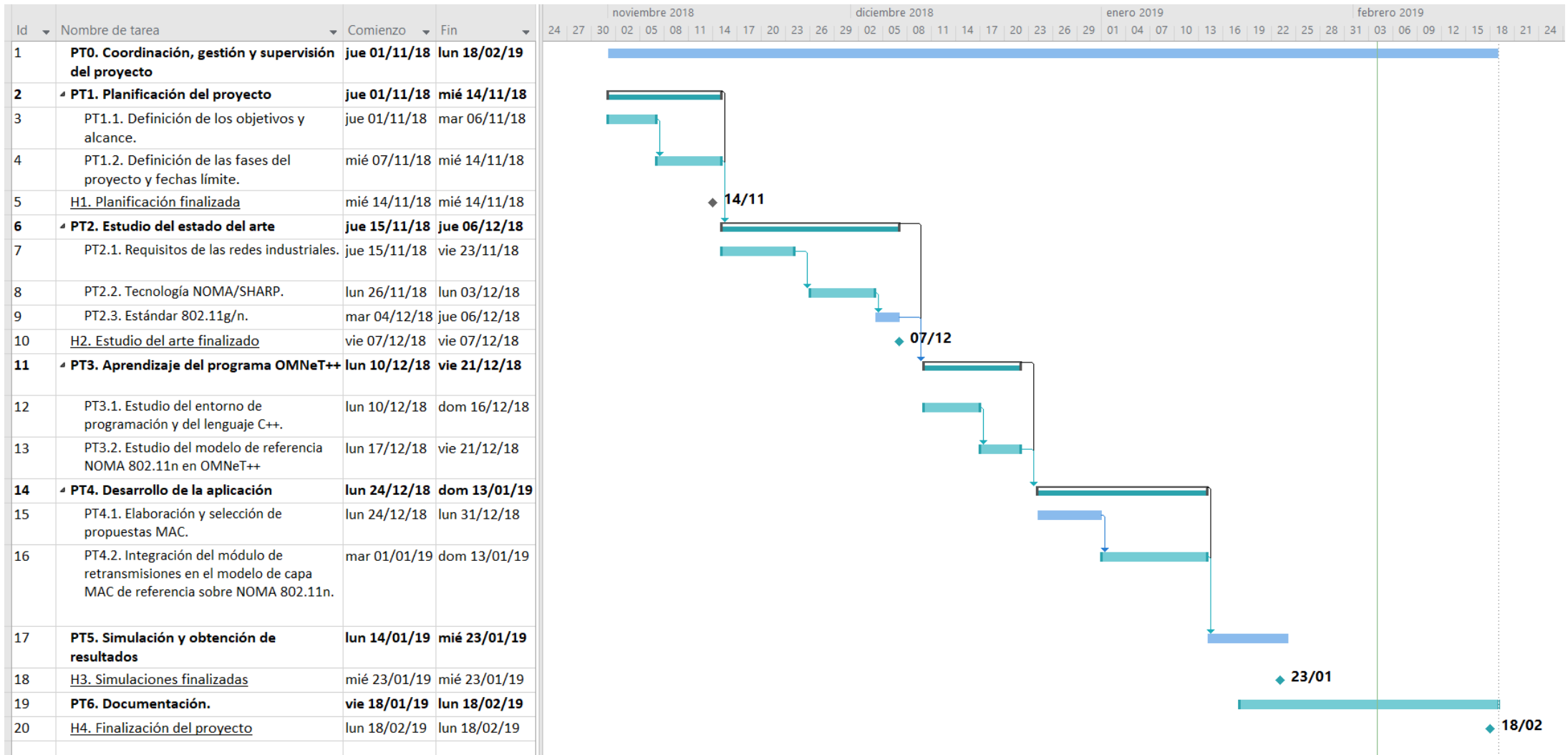


Figura 26. Diagrama de Gantt

11) PRESUPUESTO Y COSTES

Este apartado se dedica al desglose y detalle de los costes asociados a la realización del proyecto. Además, se incluyen los gastos acarreados en las amortizaciones de los equipos y en las horas de trabajo utilizadas.

11.1) Horas internas

Se detallan las horas dedicadas al proyecto por cada uno de los integrantes. Los costes indirectos (agua, luz, internet...) vienen incluidos en el coste directo del personal.

Puesto	Coste (€/h)	Número de horas (h)	Coste total (€)
Director del proyecto	70	20	1 400
Codirector del proyecto	50	60	3 000
Realizador del proyecto	25	300	7 500
TOTAL			11 900 €

Tabla 27. Horas internas.

11.2) Amortizaciones

El apartado de amortizaciones describe las depreciaciones en el valor de los equipos utilizados. Su elaboración permite saber el uso que se ha dado al material y el coste que ha supuesto para el proyecto.

De media, la vida útil de un ordenador es de 3 años, es decir 36 meses. Teniendo en cuenta que se utiliza a lo largo de 300 días, unas 8 horas al día, la vida útil asciende a 2400 horas por año, 7200 horas en 3 años. Si además el coste del equipo fue de 1000 € y para este proyecto se dedican 300 horas de uso del ordenador, podemos calcular el coste por hora ($\frac{1000}{7200} = 0,1389 \text{ €/h}$) y por consiguiente el coste total. De igual forma procedemos con las licencias, teniendo éstas una vigencia (vida útil) de 1 año.

Recurso	Coste inicial (€)	Vida útil (h)	Tiempo de uso (h)	Coste total (€)
Ordenador	1 000	7 200	300	41,67
Licencia Microsoft Office	100	1 500	105	7
Windows 10 SO	150	1 500	105	7
TOTAL				55,67

Tabla 28. Amortizaciones

11.3) Gastos totales

Finalmente, se combinan todos los gastos para obtener el presupuesto total utilizado. El resultado proporcionará la cifra a la que asciende la realización del proyecto.

Concepto	Subtotal (€)
Horas internas	11 900
Amortizaciones	55,67
TOTAL	11 955,67

Tabla 29. Presupuesto completo

El coste final del proyecto *análisis de las retransmisiones en sistemas wireless-NOMA para aplicaciones industriales* asciende a un total de once mil novecientos cincuenta y cinco con sesenta y siete euros.

En Bilbao, a 18 de febrero de 2019.

Fdo. Mario Albisu González

12) ANÁLISIS DE RIESGOS

En la realización de cualquier proyecto existe la posibilidad de aparición de riesgos que pueden comprometer en mayor o menor medida el desarrollo y los resultados del mismo. Por ello, es de vital importancia identificar los posibles riesgos, atendiendo a su impacto y aparición. Además, se clasificarán en una matriz probabilidad-impacto y se establecerán las medidas de contingencia que permitan minimizar o evitar sus efectos perjudiciales.

12.1) Identificación y evaluación de los riesgos

En esta sección se enumeran los riesgos identificados. Cada uno de ellos se definen por las variables I y P en una escala de 1 (bajo) a 3 (alto) que describen el impacto (cómo de perjudicial serían los efectos) y la probabilidad de que se den respectivamente.

- **Simulaciones poco representativas o escasas (R1):**

Antes de realizar las simulaciones, se enumeran todas las variables que puedan afectar al resultado de PER y latencia. A partir de esto, se procede a realizar un barrido de parámetros, empezando por las variables que más peso tengan en la variación de los resultados. Todas las combinaciones de variables se enumeran para sacar los *runs* o ejecuciones de simulación. Si no se añaden todas las variables, o el número de semillas (variable de inicio de aleatorización) es escaso, los resultados pueden no ser fiables y no abarcar todos los escenarios requeridos. Podría clasificarse como un riesgo de impacto medio (I=2) y probabilidad media (P=2).

- **Bajo rendimiento del modelo de retransmisiones (R2):**

Si bien se elaboran unos cálculos previos de las prestaciones de la solución, es posible que a raíz de las simulaciones se den unos resultados peores a los esperados. Esto se traduciría en que no se satisfarían los requisitos de las comunicaciones industriales, y que su aplicabilidad se vería reducida. A su vez, quedaría expuesta la capacidad de las retransmisiones para ofrecer mejor fiabilidad al sistema NOMA con la propuesta elaborada. Podría clasificarse como impacto alto (I=3) y probabilidad media (P=2).

- **Mala implementación del código (R3):**

Un mal planteamiento conceptual del mecanismo de retransmisiones o una mala implementación del código puede ocasionar demoras en la realización de simulaciones y obtención de resultados. Definir los actores participantes en el sistema es un ejercicio imprescindible además de entender los módulos que intervienen. Podría clasificarse como impacto alto (I=3) y probabilidad baja (P=1).

En la siguiente figura, se representan los riesgos descritos en función de su probabilidad de aparición y su impacto en el desarrollo del proyecto. Habrá que estar especialmente preparados para afrontar los riesgos que aparecen en la zona roja de la tabla, ya que tienen un mayor impacto en el proyecto.

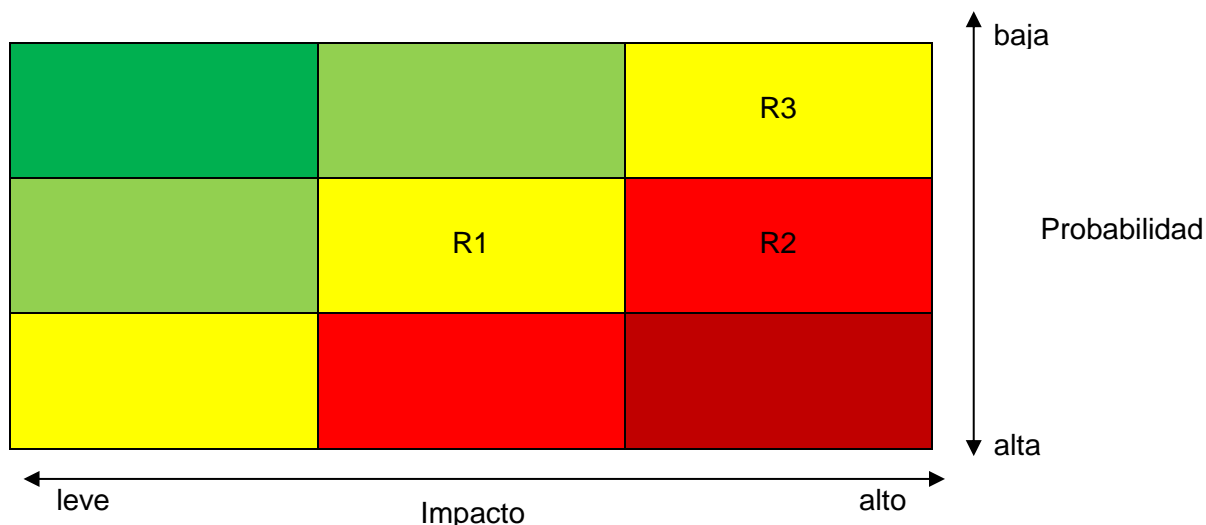


Figura 27. Matriz probabilidad-impacto

12.2) Planes de contingencia

Los planes de contingencia establecen las medidas a tomar para contrarrestar o evitar los efectos que se generan cuando se producen los riesgos previamente definidos. Permiten obtener vías alternativas para actuar si se produjeran obstáculos en el desarrollo del trabajo.

- **Simulaciones poco representativas o escasas (R1):**

En este caso, es imprescindible definir, a partir de la literatura asociada a las comunicaciones industriales, los casos de operación más representativos de las IWSN. Además, se necesita comprobar en las simulaciones que las ejecuciones correspondientes a los mismos parámetros, pero de diferentes semillas tengan resultados similares. Si no fuese así, sería necesario ampliar el número de semillas con el que se trabaja. Así se garantiza la coherencia de los resultados y su validez.

- **Bajo rendimiento del modelo de retransmisiones (R2):**

Este riesgo, si se da, será difícil de evitar una vez que se hayan obtenidos los resultados. Sin embargo, es posible reducir su impacto. Para ello, será imprescindible elaborar un acercamiento teórico sobre el que referenciar los resultados. Además, en el modelo OMNet++, es importante configurar a nivel MAC los parámetros que garanticen, según la literatura, unos resultados suficientes. El debate sobre la elección de las propuestas MAC de retransmisiones será por ello de enorme importancia para el rendimiento del sistema

- **Mala implementación del código (R3):**

Para prevenir este riesgo se debe hacer un estudio exhaustivo del lenguaje C++ y de los módulos de los que dispone OMNet++ (inet). Se deben manejar herramientas tanto gráficas como en ventana de comandos y se debe realizar un diseño previo de las funciones, variables de entrada y salida, y marcas de depuración de errores necesarias. Además, si bien no se trabaja sobre la capa PHY que es heredada (802.11n), es fundamental conocer los procedimientos de modelado de transmisión aérea y de detección de tramas erróneas.

13) CONCLUSIONES Y VISIÓN DE FUTURO

En este proyecto se ha presentado la inclusión de un sistema de retransmisiones de paquetes dentro de la propuesta de NOMA sobre 802.11. La finalidad última es dotar de más fiabilidad al sistema NOMA, con el fin de que cumpla las exigencias del sector industrial y con ello esté más cerca de ser validado. En esta sección se presentan las conclusiones principales del trabajo.

Se ha elaborado en primer lugar un estudio del marco en el que se encuadra el proyecto de referencia, la tecnología NOMA y los estándares 802.11n/g. Esto ha permitido comprender el funcionamiento del sistema que se pretende perfeccionar, atendiendo a sus capas MAC y PHY. Además, se ha podido conocer el mecanismo de retransmisiones que se sigue en otra propuesta de comunicaciones industriales inalámbrica, SHARP. Asimismo, se han interiorizado los requisitos de la industria para comunicaciones industriales, que el sistema deberá cumplir.

Se ha realizado un sondeo de posibles implementaciones MAC de retransmisiones, donde se ha decidido la realización de 2 modelos: un modelo simple, basado en redundancia de paquetes y una cadena completa, que incluye confirmaciones de paquetes. El hecho de trabajar bajo Wireless HP marca las pautas a la hora de añadir retransmisiones al modelo.

Se han descrito los modelos, haciendo especial hincapié en el funcionamiento del software, OMNet ++, y en la arquitectura de red del trabajo de referencia. A su vez, se han presentado los módulos a añadir y sus principales funciones, así como sus interacciones con el trabajo previo.

Se han expuesto los resultados, no sin antes definir las condiciones de simulación. En ellos, se ha comprobado la eficacia de las retransmisiones para mejorar la fiabilidad. Centrando el análisis sobre el servicio RT, el que más se beneficia de la tecnología LDM, se ha estudiado el modelo simple dentro del 802.11n basado en NOMA. En todos los escenarios propuestos se ha mejorado la fiabilidad en un orden de magnitud como mínimo con cada retransmisión, y, incluso, para el caso 20x20, se ha eliminado la aparición de errores con 3 envíos. En cuanto a la latencia, si bien es cierto que ha empeorado respecto al trabajo de referencia sin retransmisiones, el compromiso que ha de hacerse parece ventajoso: una ganancia de hasta 2 órdenes de magnitud en la tasa de errores de paquetes a costa de un incremento de 10 μ s como máximo en la latencia. El beneficio de NOMA de añadir determinismo a los 2 servicios permite que el análisis de la latencia sea predecible.

Desde una visión de futuro, existen varios aspectos en las que profundizar. En primer lugar, la simulación del modelo de cadena completa sería interesante, con el fin de compararlo al modelo simple. Igualmente, una nueva vía de estudio podría centrarse en el consumo de energía, un aspecto crítico en las redes industriales, que aproveche las

características de NOMA. Finalmente, existe el aspecto de la movilidad de equipos en el entorno industrial, y más utilizando tecnología inalámbrica. Este es un tema que no se aborda ya que en cada ejecución la posición de las estaciones no varía.

REFERENCIAS

- [1] Heung Mook Kim, Jae-Young Lee, Pablo Angueira, and Jon Montalban, *Layered-Division-Multiplexing: Theory and Practice*. IEEE transactions on broadcasting, vol. 62, no. 1, march 2016.
- [2] Stephan Ludwig, Michael Karrenbauer, Amina Fellan, Hans D. Schotten., *A 5G Architecture for The Factory of the Future*. 2018, ISBN 978-1-5386-7108-5
- [3] Montalban, Jon, Iradier, Eneko, Iñaki, and Angueira, Pablo. *NOMA-based 802.11n for Industrial Automation*. 2018.
- [4] Steven Dietrich, Gunther May, Oliver Wetter, Holger Heeren, *Performance Indicators and Use Case Analysis for Wireless Networks in Factory Automation*. Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limmasol, Cyprus, 12–15 Sept 2017, IEEE; DOI: 10.1109/ETFA.2017. 8247605.
- [5] José Francisco Montserrat del Río. *Sistemas de Telecomunicación Privados, WLAN*. Escola Politècnica de Gandía, 2015, https://poliformat.upv.es/access/content/group/OCW_6511_2010/Unidad%20ha%20IEEE/3.1.WLAN.pdf
- [6] Liang Zhang, Yiyang Wu, Bo Rong. *Layered Division Multiplexing (LDM) Summary*. Communications Research Centre Canada (CRC) 2010. <http://www.transmitter.com/RF251/LDM.pdf>
- [7] Pablo Angueira, Cristina Regueiro, Jon Montalban, *Layered Division Multiplexing: A technique to make flexible use of the broadcast spectrum*. Dpt Communications Engineering University of the Basque Country, DVB TM Geneva, Oct 2014. <http://www.ldm-tech.com/wpcontent/uploads/2015/04/LDM-presentation.pdf>
- [8] Yuya Saito, Yoshihisa Kishiyama, Kenichi Higuchi. *Non-orthogonal Multiple Access (NOMA) for Future Radio Access*. Tokyo University of Science, February 26, 2016, ISBN 978-1-4673-6337-2/13
- [9] Philipp Schulz, Maximilian Matthé, Henrik Klessig, Meryem Simsek. *Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture*. IEEE Communications Magazine, February 2017, ISBN 0163-6804/17
- [10] Essam Sourour, *Non-Orthogonal Multiple Access as a Candidate for 5G*. Prince Sattam Bin Abdulaziz University, Faculty of Engineering, Wadi

Aldawaser,2016,<https://engw.psau.edu.sa/sites/default/files/field/attached/Seminar%20PresentationProf%20Dr%20Essam%20Sourour%20on%20NOMA.pdf>

- [11] Óscar Seijo, Zaloa Fernández, Iñaki Val, Jesús A. López-Fernández, *SHARP: Synchronous and Hybrid Architecture for Real-Time Performance Overview*, Ikerlan, 2018
- [12] Óscar Seijo, Zaloa Fernández, Iñaki Val, Jesús A. López-Fernández, *SHARP: A Novel Hybrid Architecture for Industrial Wireless Sensor and Actuator Networks*, 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), 13-15 June 2018, Imperia, IEEE, ISBN 978-1-5386-1066-4
- [13] Tektronix MDO400. *Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements*. Tektronix Solutions, 37W-29447-0 2014, https://www.cnrood.com/en/media/solutions/WiFi_Overview_of_the_802.11_Physical_Layer.pdf
- [14] Michele Luvisotto, Zhibo Pang, Dacfeý Dzung, *Ultra High Performance Wireless Control for Critical Applications: Challenges and Directions*. IEEE transactions on industrial informatics, vol. 13, no. 3, june 2017, Paper no. TII-16-0603, ISBN 1551-3203
- [15] Wei, Y. H., Leng, Q., Han, S., Mok, A. K., Zhang, W., & Tomizuka, M. (2013, December). *RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications*. Real-Time Systems Symposium (RTSS), 2013 IEEE 34th. IEEE.
- [16] Tramarin, F., Vitturi, S., Luvisotto, M., & Zanella, A. (2016). *On the use of IEEE 802.11n for industrial communications*. IEEE Transactions on Industrial Informatics
- [17] Axel Bottcher, Christoph Meier-Hedde, *Static and dynamic retransmission control for slotted aloha in transmission channels with erasure and capture*, International Journal of Satellite Communications, Octobre 1993
- [18] Donghong Cai, Zhiguo Ding, Pingzhi Fan, *On the Performance of NOMA with Hybrid ARQ*, IEEE Transactions on Vehicular Technology, Volume: 67 , Issue: 10 , Oct. 2018, IEEE, ISSN : 1939-9359
- [19] Petcharat Suriyachai, Utz Roedig, Andrew Scott, *A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks*, IEEE Communications Surveys & Tutorials Volume: 14 , Issue: 2 , Second Quarter 2012, IEEE, ISSN : 1553-877X

ANEXO 1: CÓDIGO OMNET++

Este anexo incluye el código desarrollado en OMNet++ que permite realizar las simulaciones indicadas a lo largo de esta memoria: los archivos IsoMAC_UL.cc y IsoMAC.cc.

1.1) Cadena Completa

```
#include <IsoMAC_UL.h>
#include "inet/linklayer/common/Ieee802Ctrl.h"

namespace inet {
namespace physicallayer{

Define_Module(IsoMAC_UL);

simsignal_t IsoMAC_UL::packetSentToUpperSignal =
registerSignal("packetSentToUpper");
simsignal_t IsoMAC_UL::packetReceivedFromUpperSignal =
registerSignal("packetReceivedFromUpper");
simsignal_t IsoMAC_UL::packetFromUpperDroppedSignal =
registerSignal("packetFromUpperDropped");

simsignal_t IsoMAC_UL::packetSentToLowerSignal =
registerSignal("packetSentToLower");
simsignal_t IsoMAC_UL::packetReceivedFromLowerSignal =
registerSignal("packetReceivedFromLower");
simsignal_t IsoMAC_UL::packetFromLowerDroppedSignal =
registerSignal("packetFromLowerDropped");

void IsoMAC_UL::finish()
{
    recordScalar("numSlots", numSlots);
    recordScalar("packetSent", packetSent);
    recordScalar("numRcvRT", emergRcv);
    recordScalar("numRcvBE", videoRcv);
    recordScalar("numDroppedChannelBusy", numDroppedChannelBusy);
    recordScalar("numDroppedBitError", numDroppedBitError);
    recordScalar("numCorrectRtx0", correctos0);
    recordScalar("numCorrectRtx1", correctos1);
    recordScalar("numCorrectRtx2", correctos2);
}

void IsoMAC_UL::initialize(int stage)
{
    if (stage == INITSTAGE_LOCAL)
    {
        EV << "Initializing stage 0\n";

        upperLayerIn = findGate("upperLayerIn");
        upperLayerOut = findGate("upperLayerOut");
        lowerLayerIn = findGate("lowerLayerIn");
        lowerLayerOut = findGate("lowerLayerOut");

        fsc = 1;
        numFrames = 0;
        bAP = true;
        tipoPaquete = 0;

        initializeMACAddress();

        srcAddr = MACAddress(par("address").stringValue());
        firstSlotOffset = par("firstSlotOffset").doubleValue();
    }
}
```

```

slotSize = par("slotSize").doubleValue();
queueLength = par("queueLength").longValue();
ULLength = par("ULLength").longValue();
LLLength = par("LLLength").longValue();

// Statistics initialization.
numSlots = 0;
packetSent = 0;
emergRcv = 0;
videoRcv = 0;
numDroppedBitError = 0;
numDroppedChannelBusy = 0;
idRtx=0;
correctos0=0;
correctos1=0;
correctos2=0;
needRtx0=false;
needRtx1=false;
chFree = true;
slot5Addr = MACAddress("00:00:00:00:00:00");
slaveAddress=MACAddress("00:00:00:00:00:02");
idAddr=2;
recibidoCorrecto.setName("CorrectReceptionTime");

numScheduledSlots = 6; //Cambiar cuando el scheduler deje de ser fijo

cModule *radioModule = gate("lowerLayerOut")->getPathEndGate()-
>getOwnerModule();
//radioModule->subscribe(IRadio::radioModeChangedSignal, this);
radioModule->subscribe(IRadio::receptionStateChangedSignal, this);
radioModule->subscribe(IRadio::transmissionStateChangedSignal, this);
radioModule->subscribe(IRadio::receivedSignalPartChangedSignal, this);
radio = check_and_cast<IRadio *>(radioModule);
transmissionState = IRadio::TRANSMISSION_STATE_UNDEFINED;
GeneratedScheduler = Scheduler::createSchedule(CreateScheduler);

// Schedule the first slot timer.
slotTimer = new cMessage("slot-timeout", SLOT_TIMEOUT);
txTimer = new cMessage("tx-timeout", TX_TIMEOUT);

if(srcAddr == MACAddress("00:00:00:00:00:01"))
    scheduleAt(simTime() + firstSlotOffset, slotTimer);

}
else if (stage == INITSTAGE_LINK_LAYER)
{
    radio->setRadioMode(IRadio::RADIO_MODE_RECEIVER);
}
}

void IsoMAC_UL::initializeMACAddress()
{
    const char *addrstr = par("address");

    if (!strcmp(addrstr, "auto"))
    {
        // assign automatic address
        address = MACAddress::generateAutoAddress();

        // change module parameter from "auto" to concrete address
        par("address").setStringValue(address.str().c_str());
    }
    else
        address.setAddress(addrstr);
}

void IsoMAC_UL::handleMessage(cMessage *msg)

```



```

{
    if(msg->isSelfMessage()) //Automensaje
        handleSelfMsg(msg);

    else if(msg->getArrivalGateId()==lowerLayerIn) // Recibir
    {
        emit(packetReceivedFromLowerSignal, msg);
        handleLowerMsg(msg);
    }
    else
        throw cRuntimeError("Message '%s' received on unexpected gate '%s'",
msg->getName(), msg->getArrivalGate()->getFullName());
}

void IsoMAC_UL::handleSelfMsg(cMessage *msg)
{
    EV << simTime() << " handleSelfMsg " << msg->getFullName() << "\n";
    switch (msg->getKind())
    {
        case SLOT_TIMEOUT:
            assert(msg == slotTimer);
            performCurrentSlotTask();
            break;
        default:
            error("Unknown timer type");
            break;
    }
}

void IsoMAC_UL::incredAddr() {

    idAddr++;

    if (idAddr == 3) {
        slaveAddress.setAddress("00:00:00:00:00:03");
    }
    if (idAddr == 4) {
        slaveAddress.setAddress("00:00:00:00:00:04");
    }
    if (idAddr == 5) {
        slaveAddress.setAddress("00:00:00:00:00:05");
    }
    if (idAddr == 6) {
        slaveAddress.setAddress("00:00:00:00:00:06");
    }
    if (idAddr == 7) {
        slaveAddress.setAddress("00:00:00:00:00:07");
    }
    if (idAddr == 8) {
        slaveAddress.setAddress("00:00:00:00:00:08");
    }
    if (idAddr == 9) {
        slaveAddress.setAddress("00:00:00:00:00:09");
    }
    if (idAddr == 10) {
        slaveAddress.setAddress("00:00:00:00:00:10");
    }
    if (idAddr == 11) {

```

```

        slaveAddress.setAddress("00:00:00:00:00:11");
    }
    if (idAddr == 12) {
        slaveAddress.setAddress("00:00:00:00:00:12");
    }
    if (idAddr == 13) {
        slaveAddress.setAddress("00:00:00:00:00:13");
    }
    if (idAddr == 14) {
        slaveAddress.setAddress("00:00:00:00:00:14");
    }
    if (idAddr == 15) {
        slaveAddress.setAddress("00:00:00:00:00:15");
    }
    if (idAddr == 16) {
        slaveAddress.setAddress("00:00:00:00:00:16");
    }
    if (idAddr == 17) {
        slaveAddress.setAddress("00:00:00:00:00:17");
    }
    if (idAddr == 18) {
        slaveAddress.setAddress("00:00:00:00:00:18");
    }
    if (idAddr == 19) {
        slaveAddress.setAddress("00:00:00:00:00:19");
    }
    if (idAddr == 20) {
        slaveAddress.setAddress("00:00:00:00:00:20");
    }
    if (idAddr == 21) {
        slaveAddress.setAddress("00:00:00:00:00:21");
    }
    if(idAddr==22){
        idAddr=2;
        slaveAddress.setAddress("00:00:00:00:00:02");
    }
}

void IsoMAC_UL:: txLDM(char *sUL, char *sLL)
{
    Mac80211Pkt *macFrame = new Mac80211Pkt(sUL);
    Mac80211Pkt *macFrame2 = new Mac80211Pkt(sLL);

    macFrame->setByteLength(ULLength);
    macFrame->setSrcAddr(srcAddr);
    macFrame->setDestAddr(slaveAddress);

    macFrame2->setByteLength(LLLength);
    macFrame2->setSrcAddr(srcAddr);
    macFrame2->setDestAddr(slaveAddress);

    macFrame->encapsulate(macFrame2);

    if (transmissionState != IRadio::TRANSMISSION_STATE_TRANSMITTING) {

```

```

        macFrame->setInformationDS(idRtx);
        EV << "Retransmission number: " << idRtx << "\n";
        startTransmittingLDM(macFrame);

    } else {
        delete macFrame;
        //error("Packet cannot be sent, because MAC is busy");
    }

}

void IsoMAC_UL::startTransmittingLDM(cPacket *msg)
{
    // send
    if(radio->getReceptionState()==IRadio::RECEPTION_STATE_RECEIVING){
//Sensar el canal
        numDroppedChannelBusy++;
        EV << "Channel is busy, package will be retransmitted in the next
listening window\n";
    }
    else{
        EV << "Starting transmission of " << msg << endl;

        radio->setRadioMode(IRadio::RADIO_MODE_TRANSMITTER);

        if (msg->isPacket()){
            emit(packetSentToLowerSignal, msg);
            send(msg, lowerLayerOut);
            packetSent++;
        }
    }
}

void IsoMAC_UL::receiveSignal(cComponent *source, simsignal_t signalID, long
value DETAILS_ARG)
{
    Enter_Method_Silent();
    if (signalID == IRadio::transmissionStateChangedSignal) {
        IRadio::TransmissionState newRadioTransmissionState =
(IRadio::TransmissionState)value;
        if (transmissionState == IRadio::TRANSMISSION_STATE_TRANSMITTING &&
newRadioTransmissionState == IRadio::TRANSMISSION_STATE_IDLE){
            radio->setRadioMode(IRadio::RADIO_MODE_RECEIVER);
        }
        transmissionState = newRadioTransmissionState;
    }
}

void IsoMAC_UL::txUL(char *name)
{
    Mac80211Pkt *macFrame = new Mac80211Pkt(name);
    Mac80211Pkt *macFrame2 = new Mac80211Pkt("BE");

    macFrame->setByteLength(ULLength);
    macFrame->setSrcAddr(srcAddr);
    macFrame->setDestAddr(MACAddress("00:00:00:00:00:01"));

    macFrame2->setByteLength(LLLength);
    macFrame2->setSrcAddr(srcAddr);
    macFrame2->setDestAddr(MACAddress("00:00:00:00:00:01"));

    macFrame->encapsulate(macFrame2);
}

```

```

        if (transmissionState != IRadio::TRANSMISSION_STATE_TRANSMITTING) {
            startTransmittingUL(macFrame);
        } else {
            delete macFrame;
            //error("Packet cannot be sent, because MAC is busy");
        }
    }
}

void IsoMAC_UL::startTransmittingUL(cPacket *msg)
{
    // send
    if(radio->getReceptionState()==IRadio::RECEPTION_STATE_RECEIVING){
//Sensar el canal
        numDroppedChannelBusy++;
        EV << "Channel is busy, package will be retransmitted in the next
listening window\n";
    }
    else{
        EV << "Starting transmission of " << msg << endl;

        radio->setRadioMode(IRadio::RADIO_MODE_TRANSMITTER);

        if (msg->isPacket()){
            emit(packetSentToLowerSignal, msg);
            send(msg, lowerLayerOut);
        }
    }
}

void IsoMAC_UL::handleLowerMsg(cMessage *msg) {
    Mac80211Pkt *macPkt = static_cast<Mac80211Pkt*>(msg);
    // Check radio state

    if(macPkt->getSrcAddr()==slaveAddress && macPkt->getDestAddr()==srcAddr){
        if (radio->getRadioMode() != IRadio::RADIO_MODE_RECEIVER) {
            EV << "handleLowerMsg frame " << macPkt<< " deleted, strange
race condition\n";

            delete macPkt;
            return;
        }
        packetReceived++;

        if (macPkt->getError() || strcmp(macPkt->getName(), "NACK")==0) {

            if(idRtx!=2){
                EV << "ERROR IN UPPER LAYER, WHOLE PACKAGE DROPPED\n";

                idRtx++;
                performCurrentSlotTask();

                numDroppedBitError++;
                numDroppedBitError++;
                delete macPkt;
                return;
            }

            else{
                idRtx=0;
                incremAddr();
                performCurrentSlotTask();
            }
        }
    }
}

```

```

        else {
            EV << "UPPER LAYER SUCCESSFULLY RECEIVED\n";
            EV << "handleLowerMsg frame " << macPkt << " received from "<<
macPkt->getSrcAddr() << " address to "
            << macPkt->getDestAddr() << "\n";

            idRtx=0;
            incremAddr();
            performCurrentSlotTask();

            emergRcv++;
            if (static_cast<Mac80211Pkt*>(macPkt->getEncapsulatedPacket())-
>getError()) {

                EV << "ERROR IN LOWER LAYER, PACKAGE DROPPED\n";

                numDroppedBitError++;
            } else {
                EV << "LOWER LAYER SUCCESSFULLY RECEIVED\n";
                EV << "handleLowerMsg frame "<<
static_cast<Mac80211Pkt*>(macPkt->getEncapsulatedPacket())<< " received from "
<< macPkt->getSrcAddr()
                << " address to "<< static_cast<Mac80211Pkt*>(macPkt-
>getEncapsulatedPacket())->getDestAddr()<< "\n";

                videoRcv++;
            }
        }

        if ((macPkt->getSrcAddr()==MACAddress("00:00:00:00:00:01")) && (macPkt-
>getDestAddr()==srcAddr)){

            if (radio->getRadioMode() != IRadio::RADIO_MODE_RECEIVER) {
                EV << "handleLowerMsg frame " << macPkt<< " deleted, strange race
condition\n";

                delete macPkt;
                return;
            }
            packetReceived++;

            if (macPkt->getError()) {
                EV << "ERROR IN UPPER LAYER, WHOLE PACKAGE DROPPED\n";

                char name1[30];
                sprintf(name1, "NACK");
                txUL(name1);

                numDroppedBitError++;
                numDroppedBitError++;
                delete macPkt;
                return;
            }

            else {
                EV << "UPPER LAYER SUCCESSFULLY RECEIVED\n";
                EV << "handleLowerMsg frame " << macPkt << " received from "<< macPkt-
>getSrcAddr() << " address to "
                << macPkt->getDestAddr() << "\n";

                char name1[30];
                sprintf(name1, "ACK");
                txUL(name1);

                emergRcv++;
            }
        }
    }
}

```

```

        if (static_cast<Mac80211Pkt*>(macPkt->getEncapsulatedPacket())-
>getError()) {

            EV << "ERROR IN LOWER LAYER, PACKAGE DROPPED\n";

            numDroppedBitError++;
        } else {
            EV << "LOWER LAYER SUCCESSFULLY RECEIVED\n";
            EV << "handleLowerMsg frame "<< static_cast<Mac80211Pkt*>(macPkt-
>getEncapsulatedPacket())<< " received from " << macPkt->getSrcAddr()
            << " address to "<< static_cast<Mac80211Pkt*>(macPkt-
>getEncapsulatedPacket())->getDestAddr()<< "\n";

            videoRcv++;
        }
    }
}

else{
    EV << "PACKAGE DROPPED, NOT FOR ME " << "ADDRESS " <<
srcAddr;

    delete macPkt;
    return;
}
// End of the reception
}

void IsoMAC_UL::performCurrentSlotTask()
{
    // Moves to next slot.
    char name1[30],name2[30];
    numSlots++;

    EV << "Entering a slot assigned to me.\n";

    sprintf(name1, "RT");
    sprintf(name2, "BE");
    txLDM(name1,name2);
}
}
}

```

1.2) Modelo Simple

```

#include <IsoMAC.h>
#include "inet/linklayer/common/Ieee802Ctrl.h"

namespace inet {
namespace physicallayer{

Define_Module(IsoMAC);

simsignal_t IsoMAC::packetSentToUpperSignal =
registerSignal("packetSentToUpper");
simsignal_t IsoMAC::packetReceivedFromUpperSignal =
registerSignal("packetReceivedFromUpper");
simsignal_t IsoMAC::packetFromUpperDroppedSignal =
registerSignal("packetFromUpperDropped");

```

```

simsignal_t IsoMAC::packetSentToLowerSignal =
registerSignal("packetSentToLower");
simsignal_t IsoMAC::packetReceivedFromLowerSignal =
registerSignal("packetReceivedFromLower");
simsignal_t IsoMAC::packetFromLowerDroppedSignal =
registerSignal("packetFromLowerDropped");

void IsoMAC::finish()
{
    recordScalar("numSlots", numSlots);
    recordScalar("packetSent", packetSent);
    recordScalar("numRcvRT", emergRcv);
    recordScalar("numRcvBE", videoRcv);
    recordScalar("numDroppedChannelBusy", numDroppedChannelBusy);
    recordScalar("numDroppedBitError", numDroppedBitError);
    recordScalar("numCorrectRtx0", correctos0);
    recordScalar("numCorrectRtx1", correctos1);
    recordScalar("numCorrectRtx2", correctos2);
    recordScalar("error3", error3);
}

void IsoMAC::initialize(int stage)
{
    if (stage == INITSTAGE_LOCAL)
    {
        EV << "Initializing stage 0\n";

        upperLayerIn = findGate("upperLayerIn");
        upperLayerOut = findGate("upperLayerOut");
        lowerLayerIn = findGate("lowerLayerIn");
        lowerLayerOut = findGate("lowerLayerOut");

        fsc = 1;
        numFrames = 0;
        bAP = true;
        tipoPaquete = 0;

        initializeMACAddress();

        srcAddr = MACAddress(par("address").stringValue());
        firstSlotOffset = par("firstSlotOffset").doubleValue();
        slotSize = par("slotSize").doubleValue();
        queueLength = par("queueLength").longValue();
        ULLength = par("ULLength").longValue();
        LLLength = par("LLLlength").longValue();

        // Statistics initialization.
        numSlots = 0;
        packetSent = 0;
        emergRcv = 0;
        videoRcv = 0;
        numDroppedBitError = 0;
        numDroppedChannelBusy = 0;
        idRtx=0;
        correctos0=0;
        correctos1=0;
        correctos2=0;
        error3=0;
        needRtx0=false;
        needRtx1=false;
        chFree = true;
        slot5Addr = MACAddress("00:00:00:00:00:00");
        recibidoCorrecto.setName("CorrectReceptionTime");

        numScheduledSlots = 6; //Cambiar cuando el scheduler deje de ser hijo

        cModule *radioModule = gate("lowerLayerOut")->getPathEndGate()-
>getOwnerModule();

```

```

//radioModule->subscribe(IRadio::radioModeChangedSignal, this);
radioModule->subscribe(IRadio::receptionStateChangedSignal, this);
radioModule->subscribe(IRadio::transmissionStateChangedSignal, this);
radioModule->subscribe(IRadio::receivedSignalPartChangedSignal, this);
radio = check_and_cast<IRadio *>(radioModule);
transmissionState = IRadio::TRANSMISSION_STATE_UNDEFINED;
GeneratedScheduler = Scheduler::createSchedule(CreateScheduler);

// Schedule the first slot timer.
slotTimer = new cMessage("slot-timeout", SLOT_TIMEOUT);
txTimer = new cMessage("tx-timeout", TX_TIMEOUT);

if(srcAddr == MACAddress("00:00:00:00:00:01"))
    scheduleAt(simTime() + firstSlotOffset, slotTimer);
}
else if (stage == INITSTAGE_LINK_LAYER)
{
    radio->setRadioMode(IRadio::RADIO_MODE_RECEIVER);
}
}

void IsoMAC::initializeMACAddress()
{
    const char *addrstr = par("address");

    if (!strcmp(addrstr, "auto"))
    {
        // assign automatic address
        address = MACAddress::generateAutoAddress();

        // change module parameter from "auto" to concrete address
        par("address").setStringValue(address.str().c_str());
    }
    else
        address.setAddress(addrstr);
}

void IsoMAC::handleMessage(cMessage *msg)
{
    if(msg->isSelfMessage()) //Automensaje
        handleSelfMsg(msg);
    else if(msg->getArrivalGateId()==upperLayerIn) // Enviar
    {
        emit(packetReceivedFromUpperSignal, msg);
        handleUpperMsg(msg);
    }
    else if(msg->getArrivalGateId()==lowerLayerIn) // Recibir
    {
        emit(packetReceivedFromLowerSignal, msg);
        handleLowerMsg(msg);
    }
    else
        throw cRuntimeError("Message '%s' received on unexpected gate '%s'",
msg->getName(), msg->getArrivalGate()->getFullName());
}

void IsoMAC::handleSelfMsg(cMessage *msg)
{
    EV << simTime() << " handleSelfMsg " << msg->getFullName() << "\n";
    switch (msg->getKind())
    {
        {
        case SLOT_TIMEOUT:
            assert(msg == slotTimer);
            performCurrentSlotTask();
            break;
        default:
            error("Unknown timer type");
        }
    }
}

```



```

        break;
    }
}

void IsoMAC:: txLDM(char *sUL, char *sLL)
{
    Mac80211Pkt *macFrame = new Mac80211Pkt(sUL);
    Mac80211Pkt *macFrame2 = new Mac80211Pkt(sLL);

    macFrame->setByteLength(ULLength);
    macFrame->setSrcAddr(srcAddr);
    macFrame->setDestAddr(MACAddress::BROADCAST_ADDRESS);

    macFrame2->setByteLength(LLLength);
    macFrame2->setSrcAddr(srcAddr);
    macFrame2->setDestAddr(MACAddress::BROADCAST_ADDRESS);

    macFrame->encapsulate(macFrame2);

    if (transmissionState != IRadio::TRANSMISSION_STATE_TRANSMITTING) {
        if(idRtx!=3){
            macFrame->setInformationDS(idRtx);
            EV << "Retransmission number: " << idRtx << "\n";
            startTransmitting(macFrame);

        }
        else{
            idRtx=0;
            macFrame->setInformationDS(idRtx);
            EV << "Retransmission number: " << idRtx << "\n";
            startTransmitting(macFrame);

        }
    } else {
        delete macFrame;
        //error("Packet cannot be sent, because MAC is busy");
    }
}

void IsoMAC::startTransmitting(cPacket *msg)
{
    // send
    if(radio->getReceptionState()==IRadio::RECEPTION_STATE_RECEIVING){
//Sensar el canal
        numDroppedChannelBusy++;
        EV << "Channel is busy, package will be retransmitted in the next
listening window\n";
    }
    else{
        EV << "Starting transmission of " << msg << endl;

        radio->setRadioMode(IRadio::RADIO_MODE_TRANSMITTER);

        if (msg->isPacket()){
            emit(packetSentToLowerSignal, msg);
            send(msg, lowerLayerOut);
            idRtx++;
            packetSent++;

        }
    }
}
}

```

```

void IsoMAC::receiveSignal(cComponent *source, simsignal_t signalID, long
value DETAILS_ARG)
{
    Enter_Method_Silent();
    if (signalID == IRadio::transmissionStateChangedSignal) {
        IRadio::TransmissionState newRadioTransmissionState =
(IRadio::TransmissionState)value;
        if (transmissionState == IRadio::TRANSMISSION_STATE_TRANSMITTING &&
newRadioTransmissionState == IRadio::TRANSMISSION_STATE_IDLE){
            radio->setRadioMode(IRadio::RADIO_MODE_RECEIVER);
            scheduleAt(simTime() + 0.000002, slotTimer);
        }
        transmissionState = newRadioTransmissionState;
    }
}

void IsoMAC::handleLowerMsg(cMessage *msg) {
    Mac80211Pkt *macPkt = static_cast<Mac80211Pkt*>(msg);
    // Check radio state

    if (radio->getRadioMode() != IRadio::RADIO_MODE_RECEIVER) {
        EV << "handleLowerMsg frame " << macPkt<< " deleted, strange race
condition\n";
        if ((macPkt->getInformationDS()) == 0) {
            needRtx0 = true;
        }
        if ((macPkt->getInformationDS()) == 1) {
            needRtx1 = true;
        } if ((macPkt->getInformationDS()) == 2 && needRtx0 == true &&
needRtx1 == true){

            recibidoCorrecto.record(900);
            error3++;

        }
        delete macPkt;
        return;
    }

    packetReceived++;

    if (macPkt->getError()) {
        EV << "ERROR IN UPPER LAYER, WHOLE PACKAGE DROPPED\n";
        if ((macPkt->getInformationDS()) == 0) {
            needRtx0 = true;
        }
        if ((macPkt->getInformationDS()) == 1) {
            needRtx1 = true;
        } if ((macPkt->getInformationDS()) == 2 && needRtx0 == true &&
needRtx1 == true){

            recibidoCorrecto.record(900);
            error3++;

        }
        numDroppedBitError++;
        numDroppedBitError++;
        delete macPkt;
        return;
    }

    else {
        EV << "UPPER LAYER SUCCESSFULLY RECEIVED\n";
        EV << "handleLowerMsg frame " << macPkt << " received from "<< macPkt-
>getSrcAddr() << " address to "
        << macPkt->getDestAddr() << "\n";

        if ((macPkt->getInformationDS()) == 0) {
            recibidoCorrecto.record(simTime());

```

```

        needRtx0 = false;
        correctos0++;
    }

    if ((macPkt->getInformationDS()) == 1 && needRtx0 == true) {
        recibidoCorrecto.record(simTime());
        needRtx1 = false;
        correctos1++;
    }

    if ((macPkt->getInformationDS()) == 2 && needRtx0 == true &&
needRtx1==true) {
        recibidoCorrecto.record(simTime());
        correctos2++;
    }

    emergRcv++;
    if (static_cast<Mac80211Pkt*>(macPkt->getEncapsulatedPacket())-
>getError()) {
        EV << "ERROR IN LOWER LAYER, PACKAGE DROPPED\n";

        numDroppedBitError++;
    } else {
        EV << "LOWER LAYER SUCCESSFULLY RECEIVED\n";
        EV << "handleLowerMsg frame "<< static_cast<Mac80211Pkt*>(macPkt-
>getEncapsulatedPacket())<< " received from " << macPkt->getSrcAddr()
        << " address to "<< static_cast<Mac80211Pkt*>(macPkt-
>getEncapsulatedPacket())->getDestAddr()<< "\n";

        videoRcv++;
    }
}

if((macPkt->getInformationDS())==2){
    needRtx0 = false;
    needRtx1 = false;
}

// End of the reception
}

void IsoMAC::performCurrentSlotTask()
{
    // Moves to next slot.
    char name1[30],name2[30];
    numSlots++;

    EV << "Entering a slot assigned to me.\n";

    sprintf(name1, "RT");
    sprintf(name2, "BE");
    txLDM(name1,name2);
}

void IsoMAC::handleUpperMsg(cMessage *msg) {
    EV << "IsoMAC::handleUpperMsg " << msg->getName() << "\n";

    cPacket* upperLayerPkt = static_cast<cPacket*>(msg);

    // Checks bit length for fragmentation.
    if (upperLayerPkt->getBitLength() > 18496)
    {
        error("Packet from higher layer (%s)%s is too long for 802.11b, %d
bytes (fragmentation is not supported yet)",

```

```

        upperLayerPkt->getClassName(), upperLayerPkt->getName(),
upperLayerPkt->getByteLength());
    }

    // Encapsulates the upper layer packet into a MAC packet and sets
    // the corresponding fields.

    Mac80211Pkt *macPkt =
static_cast<Mac80211Pkt*>(encapsulate(upperLayerPkt));
    macPkt->setTimestamp(); // For calculating the channel access delay.
    macPkt->setRetry(false);
    EV << "packet " << upperLayerPkt << " received from higher layer, dest="
<< macPkt->getDestAddr() << ", encapsulated\n";

    bool queueFull = false;

    if ((unsigned)DlQueue.getLength() == queueLength) {
        queueFull = true;
    }
    else {
        DlQueue.insert(macPkt);
    }

    // Check if we could not add the packet because the queue was full.
    if (queueFull) {
        msg->setName("MAC ERROR");
        //msg->setKind(PACKET_DROPPED);
        //sendControlUp(msg);
        EV << "packet " << msg << " received from higher layer but MAC queue
is full, signalling error\n";
    }
}

Mac80211Pkt *IsoMAC::encapsulate(cPacket* msg)
{
    Mac80211Pkt *pkt = new Mac80211Pkt("DATA");

    pkt->setDestAddr(MACAddress::BROADCAST_ADDRESS);
    pkt->setSrcAddr(srcAddr);
    pkt->setRetry(false); // This is not a retry.
    pkt->setSequenceControl(fsc++); // Add a unique fsc to it.

    if(fsc <= 0)
        fsc = 1;

    return pkt;
}

void IsoMAC::sendSensor() // No se simula el UL
{
    Mac80211Pkt *macFrame = new Mac80211Pkt("Actual State");

    macFrame->setByteLength(ULLength);
    macFrame->setDestAddr(MACAddress("00:00:00:00:00:01"));
    macFrame->setSrcAddr(srcAddr);

    if(transmissionState != IRadio::TRANSMISSION_STATE_TRANSMITTING)
        startTransmitting(macFrame);
    else
        delete macFrame;
//    error("Packet cannot be sent, because MAC is busy");
}
}
}

```




BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO