



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño y evaluación de redes definidas por software para la orquestación dinámica de calidad de servicio en redes industriales de sensores inalámbricos.

Federico Orozco Santos

Director:

Dr. Víctor Miguel Sempere Payá

Dr. Javier Lidiano Silvestre Blanes

Universitat Politècnica de València

Departamento de Comunicaciones

Valencia

Septiembre 2024

Diseño y evaluación de redes definidas por software para la orquestación dinámica de calidad de servicio en redes industriales de sensores inalámbricos.

Federico Orozco Santos

Director:

Dr. Víctor Miguel Sempere Payá
Dr. Javier Lidiano Silvestre Blanes

Universitat Politècnica de València
Departamento de Comunicaciones
Valencia
Septiembre 2024

A Jhovana y Fede, mi equipo de apoyo incondicional en cada paso de este camino.

A mi familia, y especialmente a mi hermana Pao, por su apoyo constante y amor incondicional a lo largo de todos los desafíos y momentos importantes de mi vida.

A Víctor y Javier, mis directores de tesis, cuyo compromiso, dedicación y experiencia han sido fundamentales para el avance y culminación de este trabajo.

Resumen

Las redes industriales de sensores inalámbricos (IWSN - *Industrial Wireless Sensor Networks*) tienen una creciente aceptación en los entornos productivos debido a su facilidad de despliegue, bajos costes y eficiencia energética. Sin embargo, la complejidad y precisión que demandan estos entornos requieren que las IWSN implementen mecanismos de calidad de servicio (QoS) que les permitan operar con un alto determinismo.

Por esta razón, el estándar IEEE 802.15.4e ha incorporado protocolos de acceso al medio que permiten tener un mayor control sobre las comunicaciones. Estos protocolos permiten asignar de forma predecible y cíclica instantes de tiempo donde los dispositivos de una red pueden transmitir información sin ningún tipo de interferencia o contienda con otros dispositivos de la red. Uno de ellos es el protocolo salto de canal con ranuras de tiempo (TSCH - *Time Slotted Channel Hopping*) el cual planifica las transmisiones entre los dispositivos en una matriz de dos dimensiones, tiempo y frecuencia. Esta planificación de las transmisiones permite que los dispositivos tengan instantes de tiempo exclusivos para la transmisión, reduciendo así la interferencia y aumentando la fiabilidad de las transmisiones. Sin embargo, este estándar no especifica como asignar los recursos temporales en la planificación TSCH, dando lugar a múltiples soluciones de planificación.

Estos enfoques de planificación adquieren una gran relevancia en el actual entorno de la Industria 4.0, donde se busca aumentar de forma masiva la digitalización la interacción y la integración con los diferentes procesos industriales. Por lo tanto, hay una gran diversidad de flujos de información sobre la misma red. Además, al ser flujos de diferentes procesos, también tienen diferentes criticidades, donde deben garantizarse parámetros de QoS como latencia, throughput, pérdida de paquetes y deadline. Esta creciente demanda de QoS y la diversidad de flujos de tráfico requieren un control estricto y dinámico que no puede lograrse con planificadores convencionales. Por esto, la evolución hacia nuevos paradigmas como las redes definidas por software (SDN - *Software Defined Networks*) se está convirtiendo en una interesante alternativa para abordar la complejidad de los retos actuales en las IWSN. Estas aceptan un aumento del tráfico de señalización a cambio de ventajas adicionales, tales como: una reducción de la complejidad de los procesos en los nodos, una visión global de todos los elementos de la red y una alta capacidad de reconfiguración.

En el desarrollo de esta tesis, se aprovecha el alto nivel de detalle y control que ofrecen las SDN para abordar los complejos desafíos que enfrenta actualmente la adopción de las IWSN

en el sector industrial, como la movilidad, la escalabilidad y la garantía de parámetros de QoS. Estos desafíos presentan una complejidad significativa al utilizar los protocolos convencionales; por ello, es necesario implementar un cambio disruptivo que permita superar estas limitaciones de manera sencilla y eficiente. En consecuencia, esta tesis se centra en la investigación y aplicación de mecanismos que permitan adaptar las IWSN al paradigma SDN, así como el desarrollo de protocolos de enrutamiento y planificación TSCH dinámicos que aseguren un alto grado de determinismo y una alta capacidad de adaptación a los cambios en la topología. De esta forma, se garantizan requisitos de QoS independientes para cada tipo de flujo de manera dinámica y con una baja complejidad, incluso en condiciones de movilidad y alta densidad de nodos.

Palabras clave: Escalabilidad, IWSN, Movilidad, QoS, SDN, TSCH.

Resum

Les xarxes industrials de sensors sense fils (IWSN - *Industrial Wireless Sensor Networks*) tenen una creixent acceptació en els entorns productius a causa de la seua facilitat de desplegament, baixos costos i eficiència energètica. No obstant això, la complexitat i precisió que demanden aquests entorns requereixen que les IWSN implementen mecanismes de qualitat de servici (QoS) que els permeten operar amb un alt determinisme.

Per aquesta raó, l'estàndard IEEE 802.15.4e ha incorporat protocols d'accés al medi que permeten tindre un major control sobre les comunicacions. Aquest protocols permeten assignar de manera predictable i cíclica instants de temps on els dispositius d'una xarxa poden transmetre informació sense cap mena d'interferència o contesa amb altres dispositius de la xarxa. Un d'ells és el protocol salte de canal amb espais de temps (TSCH - *Time Slotted Channel Hopping*) el qual planifica les transmissions entre els dispositius en una matriu de dos dimensions, temps i freqüència. Aquesta planificació de les transmissions permet que els dispositius tinguen instants de temps exclusius per a la transmissió, reduint així la interferència i augmentant la fiabilitat de les transmissions. No obstant això, aquest estàndard no especifica com assignar els recursos temporals en la planificació TSCH, donant lloc a múltiples solucions de planificació.

Aquest enfocaments de planificació adquireixen una gran rellevància en l'actual entorn de la Indústria 4.0, on es busca augmentar de manera massiva la digitalització, la interacció i la integració amb els diferents processos industrials. Per tant, hi ha una gran diversitat de fluxos d'informació sobre la mateixa xarxa. A més, en ser fluxos de diferents processos, també tenen diferents criticitats, on han de garantir-se paràmetres de QoS com a latència, throughput, pèrdua de paquets i deadline. Aquesta creixent demanda de QoS i la diversitat de fluxos de trànsit requereixen un control estricte i dinàmic que no pot aconseguir-se amb planificadors convencionals. Per això, l'evolució cap a nous paradigmes com les xarxes definides per programari (SDN - *Software Defined Networks*) s'està convertint en una interessant alternativa per a abordar la complexitat dels reptes actuals en les IWSN. Aquestes accepten un augment del trànsit de senyalització a canvi d'avantatges addicionals, com ara: una reducció de la complexitat dels processos en els nodes, una visió global de tots els elements de la xarxa i una alta capacitat de reconfiguració.

En el desenvolupament d'aquesta tesi, s'aprofita l'alt nivell de detall i control que ofereixen les SDN per a abordar els complexos reptes que enfronta actualment l'adopció de les IWSN

en el sector industrial, com la mobilitat, l'escalabilitat i la garantia de paràmetres de QoS. Aquests reptes presenten una complexitat significativa en utilitzar els protocols convencionals; per això, és necessari implementar un canvi transgressor que permeti superar aquestes limitacions de manera senzilla i eficient. En conseqüència, aquesta tesi se centra en la investigació i aplicació de mecanismes que permeten adaptar les IWSN al paradigma SDN, així com el desenvolupament de protocols d'encaminament i planificació TSCH dinàmics que assegurin un alt grau de determinisme i una alta capacitat d'adaptació als canvis en la topologia. D'aquesta manera, es garanteixen requisits de QoS independents per a cada tipus de flux de manera dinàmica i amb una baixa complexitat, fins i tot en condicions de mobilitat i alta densitat de nodes.

Paraules clau: Escalabilitat, IWSN, Mobilitat, QoS, SDN, TSCH.

Abstract

Industrial Wireless Sensor Networks (IWSNs) are becoming increasingly popular in production environments due to their ease of deployment, low cost and energy efficiency. However, the complexity and accuracy demanded by these environments require IWSNs to implement quality of service (QoS) mechanisms that allow them to operate with high determinism.

For this reason, the IEEE 802.15.4e standard has incorporated medium access protocols that allow for greater control over communications. These protocols enable predictable and cyclic allocation of time slots where devices in a network can transmit information without any interference or contention with other devices in the network. One such protocol is the Time Slotted Channel Hopping (TSCH) protocol, which schedules transmissions between devices using a two-dimensional matrix of time and frequency. This scheduling of transmissions allows devices to have unique time slots for transmission, thereby reducing interference and increasing the reliability of transmissions. However, this standard does not specify how to allocate time resources in TSCH scheduling, leading to multiple scheduling solutions such as centralized scheduling, distributed scheduling, and hybrid scheduling, each with its own advantages and disadvantages.

These planning approaches become highly relevant in today's Industry 4.0 environment, where digitisation, interaction and integration with different industrial processes are being massively increased. Therefore, there is a great diversity of information flows over the same network. Moreover, as they are flows of different processes, they also have different criticalities, where QoS parameters such as latency, throughput, packet loss, and deadline must be guaranteed. This increasing QoS demand and the diversity of traffic flows require strict and dynamic control that cannot be achieved with conventional schedulers. Therefore, the evolution towards new paradigms such as Software Defined Networks (SDN) is becoming an interesting alternative to address the complexity of today's challenges in IWSNs. SDNs, in the context of IWSNs, refer to the use of software to manage and control the network, allowing for a more efficient and dynamic allocation of resources. They accept an increase in signalling traffic in exchange for additional benefits, such as: a reduction of the complexity of processes at the nodes, a global view of all network elements and a high reconfiguration capacity.

In the development of this thesis, the high level of detail and control offered by SDNs is

leveraged to address the complex challenges currently facing the adoption of IWSNs in the industrial sector, such as mobility, scalability and QoS parameter guarantees. These challenges present significant complexity when using conventional protocols; therefore, a disruptive change needs to be implemented to overcome these limitations in a simple and efficient manner. Consequently, this thesis focuses on the investigation and implementation of specific mechanisms, such as dynamic resource allocation and traffic prioritization, to adapt IWSNs to the SDN paradigm. Additionally, new protocols, including dynamic TSCH routing and scheduling protocols, are being developed to ensure a high degree of determinism and a high adaptability to topology changes. In this way, independent QoS requirements for each flow type are guaranteed dynamically and with low complexity, even under conditions of mobility and high node density.

Keywords: IWSN, Mobility, QoS, SDN, Scalability, TSCH

Lista de Figuras

1-1. Arquitectura jerárquica Edge, Cloud y Fog	2
2-1. Tecnologías de comunicación inalámbrica	14
2-2. Topologías típicas en WPAN	15
2-3. Comparación de la pila de protocolos en diferentes tecnologías de WPAN . .	17
2-4. Encabezados y fragmentación 6LoWPAN	19
2-5. Supertrama o <i>Slotframe</i> del protocolo DSME	27
2-6. Macrotrama o <i>Slotframe</i> del protocolo TSCH	28
2-7. Redes definidas por software en: (a) Planos, (b) Capas y (c) Arquitectura. .	32
3-1. (a) Pila de protocolos SDN-WISE, (b) Paquetes de beacon, (c) Paquetes de reporte	41
3-2. Estructura del paquete OpenPath original. (a) ejemplo genérico, (b) ejemplo con path de 3 nodos y 1 Regla de flujo	43
3-3. Arquitectura general SDN WISE-TSCH	45
3-4. Flujo entre los nodos 10 y 1. (a) Dijkstra simple basado en el RSSI, (b) Dijkstra acumulado con balanceo de carga	50
3-5. Panificación de múltiples flujos con diferentes deadlines	55
3-6. Estructura genérica del paquete OpenPathTSCH	57
3-7. Topología de 10 nodos, planificación TSCH de un path con 5 nods y NR=2 .	57
3-8. Estructura del paquete OpenPathTSCH con la planificación del path con 5 nodos y NR=2	58
3-9. Tiempo de convergencia SDN con sincronización TSCH	63
3-10. Tiempo medio de sincronización por salto en una topología lineal [70].	64
3-11. Tráfico de control generado y uso de la radio en los protocolos RPL y SDN- WISE TSCH	65
3-12. Tiempo de respuesta y cambio de planificación	65
3-13. (a) PIAT para todos los flujos, (b) Total de paquetes enviados por flujo . . .	66
3-14. Distribución de los flujos sobre los nodos (a) SDN-WISE TSCH (b) única ruta para todos los nodos, (c) network lifetime para ambos casos.	67
3-15. PIAT y PDR con cambio de ruta después del fallo del nodo 8	68

3-16. Comparación entre SDN-WISE TSCH y planificación distribuida	69
3-17. Efecto de múltiples transmisores simultáneos en la topología de prueba	70
3-18. Elementos del testbed para topología de 10 nodos	71
3-19. Tolerancia a fallos de SDN WISE-TSCH en tesbed	72
4-1. Distribución del plano de control para los diferentes tipos de planificadores TSCH (a) Autónomos (b) Distribuidos (c) Centralizados (SDN) (d) Centrali- zados	77
4-2. Topología con 10 nodos fijos.	84
4-3. E2E delay para distintos periodos de transmisión.	87
4-4. Radio duty cycle para cada planificador en función del periodo de paquete.	88
4-5. PIAT para cada planificador con diferentes periodos de paquetes.	89
4-6. Deadline Satisfaction ratio para cada planificador.	90
4-7. PIAT para cada uno de los planificadores con múltiples flujos de datos.	91
4-8. Testbed usando OpenMote-B.	92
4-9. E2E delay y PIAT en testbed con SDN WISE-TSCH y RPL + Orchestra.	93
4-10. DSR obtenido en el testbed para cada nodo Tx.	94
5-1. Topología y slotframe usado en FTS-SDN	100
5-2. Efecto del coeficiente de ocupación en la cantidad de tráfico de control.	104
5-3. Topología de referencia y slotframe con MMF-SDN	106
5-4. Número de estados de transmisión	107
5-5. Número de estados de recepción	108
5-6. Consumo de energía por nodo en el escenario 1.	110
5-7. Número de paquetes de control enviados durante 1 hora de simulación	111
5-8. (a) E2E delay con $v = 1m/s$ para FTS-SDN y MMF-SDN, (b) Consumo medio de energía en el escenario 1.	112
5-9. Comparación de MMF-SDN y FTS-SDN a distintas velocidades de los nodos móviles.	113
5-10. PDR con diferentes velocidades del nodo móvil.	114
5-11. Topología con 43 nodos, 3 nodos móviles.	115
5-12. Total de paquetes de control en función del número de nodos para $T = 10$ s, con diferentes periodos de beacon (B_p) y periodos de reporte (R_p).	116
5-13. Número de slots compartidos a utilizar en función del tráfico de control.	117
5-14. E2E delay en topología con 43 nodos	119
5-15. Escenario en un entorno industrial con 25 nodos fijos y 2 móviles.	119
5-16. E2E delay para el movimiento horizontal y vertical del escenario 3.	120
5-17. Comparación entre MMF-SDN y FTS-SDN Escenario 3.	121
5-18. Estabilidad de MMF-SDN en el escenario 3.	122
6-1. SD-IWSN sink convencional con una única interfaz de radio	130

6-2. SD-IWSN sink virtual con 2 interfaces radio usando 4 canales.	131
6-3. Número de interfaces de radio en función del número de nodos y de la frecuencia de paquetes con $n_{L1} = 8$	133
6-4. Número de interfaces de radio en función del número de nodos y de la frecuencia de paquetes con $n_{L1} = 24$	134
6-5. Múltiples SD-IWSN separadas por frecuencia y orquestadas por el mismo controlador SDN	135
6-6. Topología de malla cuadrada con sink en el centro	135
6-7. Aumento de los saltos al sink en topología de malla cuadrada en función del número de nodos	136
6-8. Tamaño del OpenPathTSCCH en función del número de nodos y repeticiones dentro del Slotframe	137
6-9. Promedio de paquetes recibidos/transmitidos por cada nodo en función de la distancia al sink	139
6-10. Total paquete de datos en simulación con sink convencional	140
6-11. Ocupación del slotframe en función de la cantidad de nodos	140
6-12. Promedio de paquetes recibidos/transmitidos por los nodos en función de la distancia al sink	141
6-13. Paquetes de datos totales en simulaciones con sink virtual compuesto de 2 interfaces radio	142
6-14. Testbed: Dos SD-IWSN con topología de malla cuadrada, sinks virtuales con dos interfaces radio	144
6-15. PDR y Utilización del Slotframe en el testbed con diferentes frecuencias de paquetes	146

Lista de Tablas

1-1. Correspondencia entre los objetivos y contribuciones con los capítulos.	11
2-1. Comparación de protocolos inalámbricos basados en 802. 15.4	16
2-2. Comparación de las diferentes versiones de Bluetooth	23
2-3. Comparativa de Protocolos de Control de Acceso al Medio en 802.15.4e	30
3-1. Clasificación de las prioridades	46
3-2. Parámetros utilizados en la simulación	60
3-3. Flujos de tráfico de datos	61
3-4. Modos de operación y consumo de energía para el OpenMote-B	62
4-1. Resumen de las características de cada uno de los planificadores TSCH.	83
4-2. Parámetros de simulación.	85
4-3. Resumen de características de los distintos planificadores TSCH.	92
5-1. Descripción de los parámetros utilizados	102
5-2. Parámetros usados en el escenario 1	109
5-3. Parámetros utilizados en el escenario 2	115
5-4. Parámetros utilizados en el escenario 3	120
6-1. Comparativa de estudios sobre escalabilidad en IWSN	127
6-2. Resumen de las notaciones utilizadas en este capítulo	129
6-3. Parámetros utilizados en la simulación	138
6-4. Uso de CPU y RAM en el controlador SDN ONOS con múltiples IWSN	144
6-5. Parámetros utilizados en el testbed	145

Lista de acrónimos

Acrónimo	Definición
AI	Artificial Intelligence
ASN	Absolute Sequence Number
ACK	Acknowledgement
API	Application Programming Interface
ALICE	Autonomous Link-based Cell scheduling
AMUS	Adaptative Multi-hop Scheduling
CPS	Cyber-Physical Systems
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
DAG	Directed Acyclic Graph
DeTAS	Decentralized Traffic Aware Scheduling
DSME	Deterministic and Synchronous Multi-Channel Extension
DSR	Deadline Satisfaction Ratio
EB	Enhanced Beacon
FDMA	Frequency Division Multiple Access
IETF	Internet Engineering Task Force
IIoT	Industrial Internet of Things
IoT	Internet of Things
IWSN	Industrial Wireless Sensor Network
LAN	Local Area Network
LLDN	Low Latency Deterministic Networks
M2M	Machine to Machine
MAC	Media Access Control
MANET	Mobile Ad-hoc Network
ML	Machine Learning
MMF	Mobile Multicast Forwarding
MQTT	Message Queuing Telemetry Transport
MSF	Minimal Scheduling Function
NFV	Network Function Virtualization
NFVO	Network Function Virtualization Orchestrator

Acrónimo	Definición
NFVM	Network Function Virtualization Manager
PCE	Path Computation Element
PDR	Packet Delivery Rate
PIAT	Packet Inter-Arrival Time
ONF	Open Networking Foundation
QoS	Quality of Service
RPL	Routing Protocol for Low-Power and Lossy Networks
RSSI	Received Signal Strength Indication
SDN	Software Defined Networking
SDR	Software-Defined Radio
TASA	Traffic Aware Scheduling Algorithm
TDMA	Time Division Multiple Access
TSCH	Time-Slotted Channel Hopping
TSN	Time-Sensitive Networking
URLLC	Ultra Reliable Low Latency Communications
VIM	Virtualized Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
WAN	Wide Area Network
WSAN	Wireless Sensor and Actuator Network
WSN	Wireless Sensor Network

Contenido

Resumen	VII
Resumen	IX
Abstract	XI
Lista de figuras	XIII
Lista de tablas	XV
Lista de acrónimos	XVII
1. Introducción	1
1.1. Antecedentes y motivación	1
1.2. Objetivos	5
1.3. Publicaciones	7
1.3.1. Artículos en revistas científicas	7
1.3.2. Proyectos de investigación	8
1.4. Contribución	9
1.5. Estructura de la tesis	10
2. Estado del arte	13
2.1. WPAN	14
2.1.1. IEEE 802.15.4	16
2.1.2. Bluetooth	22
2.2. IoT en la industria	22
2.2.1. Calidad de servicio (QoS)	23
2.2.2. Redes de sensores inalámbricos industriales (IWSN)	25
2.2.3. Control de Acceso al medio en redes 802.15.4e	26
2.2.4. Planificadores TSCH	29
2.3. Redes definidas por software	31

2.4. SDN en WSN	33
2.5. Movilidad en IWSN	35
3. Integración de SDN en IWSN	39
3.1. Introducción	39
3.2. Arquitectura	40
3.2.1. SDN-WISE	40
3.2.2. Arquitectura propuesta: SDN WISE-TSCH	44
3.3. Descriptor de tráfico y QoS	46
3.4. Algoritmo de enrutamiento	47
3.5. Planificador TSCH en SDN WISE-TSCH	51
3.6. Modificación SDN WISE	55
3.7. Evaluación del rendimiento	60
3.7.1. Escenarios y métricas	60
3.8. Análisis de resultados	62
3.8.1. Simulación	62
3.8.2. Testbed	71
3.9. Conclusiones	72
4. Planificación de acceso al medio determinista	75
4.1. Introducción	75
4.2. Planificadores TSCH	76
4.2.1. Autónomos	77
4.2.2. Distribuidos	78
4.2.3. Centralizados	79
4.3. Escenarios y métricas	83
4.4. Análisis de resultados	86
4.4.1. Retardo extremo a extremo y tasa de pérdida de paquetes	86
4.4.2. Consumo de energía	88
4.4.3. Tiempo entre paquetes y tasa de satisfacción de deadline	89
4.4.4. Pruebas Multi-flujo	90
4.4.5. Testbed	92
4.4.6. Retardo extremo a extremo y tiempo entre paquetes	93
4.4.7. Tasa de satisfacción de deadline	94
4.4.8. Consumo de energía	95
4.5. Conclusiones	95
5. Gestión dinámica de recursos en IWSN con nodos móviles	97
5.1. Introducción	97
5.2. Movilidad en IWSN	98
5.2.1. Mobile Multicast Forwarding SDN (MMF-SDN)	100

5.2.2. Caracterización y análisis del tráfico de control	102
5.2.3. Asignación de la planificación TSCH	104
5.3. Evaluación del rendimiento	109
5.3.1. Escenario 1	109
5.3.2. Escenario 2	115
5.3.3. Escenario 3	118
5.4. Conclusiones	122
6. Escalabilidad en SD-IWSN	125
6.1. Introducción	125
6.1.1. Análisis de escalabilidad en WSN	126
6.1.2. Análisis del tráfico de control	128
6.1.3. Propuesta de escalabilidad en SD-IWSN	130
6.1.4. Topología	135
6.1.5. Configuración de rutas	136
6.2. Evaluación del rendimiento	138
6.2.1. Sink convencional	138
6.2.2. Sink virtual	141
6.2.3. Múltiples SD-IWSN con sink virtual	143
6.2.4. Tesbed	144
6.3. Conclusiones	146
7. Conclusiones y trabajos futuros	149
Bibliografía	152

Capítulo 1

Introducción

1.1. Antecedentes y motivación

En el contexto actual, caracterizado por un entorno global cada vez más interconectado, la tecnología desempeña un papel esencial en la transformación y evolución de diversos ámbitos de nuestra sociedad, generando una creciente necesidad de mejorar la eficiencia, garantizar la seguridad y optimizar la utilización de recursos en las redes de comunicaciones y las infraestructuras de red. Para abordar estos desafíos y aprovechar eficazmente las oportunidades emergentes, se han desarrollado diversos conceptos y tecnologías como, por ejemplo, *Edge*, *Cloud* y *Fog computing*, que han cobrado relevancia como soluciones clave que permiten una distribución efectiva y adaptativa de los recursos y servicios computacionales en diferentes niveles de la infraestructura de red. El *Edge computing* se refiere a la capacidad de procesar datos y tomar decisiones en tiempo real cerca de donde se generan los datos, lo que permite una mayor eficiencia y menores latencias. Por otro lado, el *Cloud computing* se basa en la utilización de recursos de computación y almacenamiento en centros de datos remotos, proporcionando alta disponibilidad y acceso a servicios a gran escala. En cambio, el *Fog* actúa como un puente entre el *Edge* y el *Cloud*, proporcionando capacidades de computación y almacenamiento intermedias, y permitiendo un mejor equilibrio entre la descentralización y la centralización de los recursos como se muestra en la figura **1-1**.

Esta arquitectura jerarquizada ha simplificado la incorporación de tecnologías avanzadas como la Inteligencia Artificial (IA) y el *BigData*, las cuales suelen necesitar un amplio volumen de recursos computacionales para procesar y analizar grandes cantidades de datos. La sinergia entre *Edge*, *Cloud* y *Fog computing* con IA facilita la implementación de soluciones de análisis de alto nivel y aprendizaje automático en diferentes capas de la red, lo cual se traduce en una mayor adaptabilidad y eficiencia en el procesamiento de la información. Para mejorar la gestión y control de esta infraestructura de red en constante evolución, el concepto SDN, ha surgido como una propuesta que brinda mayor flexibilidad y escalabilidad al desplegar

y administrar recursos y servicios, facilitando la adaptación a las exigencias variables del entorno. La arquitectura SDN introduce una separación entre el plano de control y el plano de datos, lo que facilita la implementación de políticas de red y la reconfiguración dinámica de la misma.

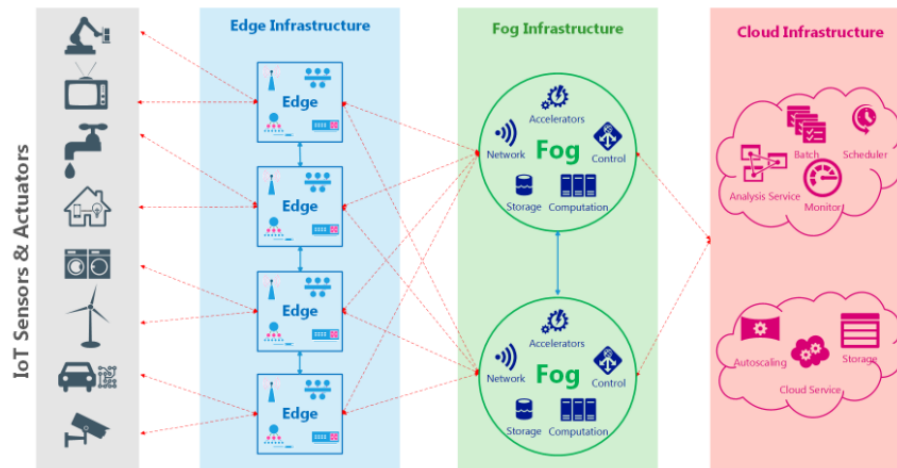


Figura 1-1: Arquitectura jerárquica Edge, Cloud y Fog [1]

Esta integración de tecnologías ha reforzado la capacidad para enfrentar desafíos y aprovechar oportunidades en entornos en constante cambio. Sin embargo, la base y efectividad de estos procesos complejos radica en la digitalización, elemento esencial que actúa como impulsor en la implementación y evolución de soluciones de IA y sistemas de comunicación avanzados. Por ello, la integración del internet de las cosas (IoT – *Internet of Things*) en el ámbito de la automatización industrial ha dado lugar a la transformación digital conocida como Industria 4.0 [2], donde, gracias al IoT es posible recopilar y procesar información en el *Edge*, donde se genera la información [3]. La popularidad y demanda actual de dispositivos conectados ha sido el enlace perfecto para obtener más información de los entornos industriales, con un costo relativamente bajo, siendo fácilmente sustituibles y poco intrusivos. Sin embargo, los requisitos temporales, la fiabilidad y la seguridad son más estrictos en entornos industriales que en el IoT. Por eso, dentro de IoT, existe una rama especializada para el uso industrial que consta de dispositivos y protocolos más robustos, que se adaptan a las necesidades de la industria, denominada Industrial-IoT (IIoT).

Una parte importante del IIoT [4] la constituyen las IWSN. Estas redes se encuentran conformadas por una gran cantidad de nodos especializados en la adquisición y transmisión de datos. Por lo general, son dispositivos de bajo costo con capacidades limitadas, debido a su tamaño reducido y al funcionamiento con baterías, lo que lleva a limitaciones en cuanto a procesamiento, memoria y consumo energético [5]. Estos dispositivos han tenido éxito en la industria y han logrado impulsar una revolución en la captura de información. No obstante, los procesos actuales de convergencia e industrialización requieren que los dispositivos in-

teractúen de manera directa con el entorno. Por esto, su funcionamiento se ha expandido a las redes industriales de sensores y actuadores inalámbricos (IWSAN - *Industrial Wireless Sensor and Actuator Networks*), agregando actuadores que les permiten realizar acciones en el plano físico de acuerdo con los datos recogidos.

Esta interacción requiere un mayor control sobre la infraestructura de red, ya que se debe garantizar un alto grado de fiabilidad, flexibilidad y bidireccionalidad en las comunicaciones, para así lograr que los flujos de información sean completamente deterministas y garantizados. Por esta razón, tanto las IWSN como las IWSAN deben evolucionar para converger hacia los nuevos paradigmas de red como por ejemplo SDN [6], que les permitan solucionar problemas intrínsecos de las IWSN e IWSAN como son, la garantía de calidad de servicio, la dificultad de administración y la capacidad de reconfiguración [7], [8]. Los protocolos en una red SDN no tienen una direccionalidad, es decir, son igual de eficientes administrando tráfico de *upstream* y *downstream* en cualquier punto de la red. A diferencia de las redes IWSN tradicionales, cuyos protocolos y operación están diseñados para enviar tráfico en *upstream* es decir, desde las hojas al nodo sink, las IWSAN utilizan protocolos diferentes, optimizados para lograr la bidireccionalidad. Por ello, en esta tesis el término IWSN es equivalente a IWSAN cuando la gestión se realiza por SDN.

Las IWSN se han ido perfeccionando con protocolos como el RPL (*Routing Protocol for Low-Power and Lossy Networks*) y TSCH [9], que permiten planificar las transmisiones para evitar colisiones e interferencias [10]. Sin embargo, las decisiones de enrutamiento y planificación TSCH tomadas de forma autónoma por cada nodo afectan el determinismo porque la previsibilidad del comportamiento de la red se ve limitada por la naturaleza distribuida y dinámica de estas decisiones. Asimismo, la flexibilidad se ve limitada debido a que no es posible adaptar de manera dinámica la red en respuesta a cambios en el tráfico o condiciones de red de manera eficaz. No obstante, el uso de un enfoque centralizado con un conocimiento total de la red permite optimizar el enrutamiento, la planificación TSCH, y reducir la carga de procesamiento en los nodos al hacer un menor uso de protocolos locales. Aun así, el inconveniente principal de los enfoques centralizados es que implican agregar un flujo de tráfico adicional para la información de control. Esto reduce el rendimiento de la red como se ha demostrado con TASA [11] y AMUS [12]. El principal problema de estos trabajos es que no representan un cambio significativo en la pila de protocolos y operación de los nodos. Por eso se hace necesario un enfoque más disruptivo como las SDN, donde se divide la red en dos planos diferentes, de control y datos, y los nodos operan según las instrucciones de un agente centralizado que tiene un conocimiento global de la red denominado controlador SDN. Esto simplifica la administración de la red, debido a que las aplicaciones se desarrollan orientadas al controlador, el cual abstrae las configuraciones de los dispositivos y en general el funcionamiento a bajo nivel de la red. Esta abstracción permite que las aplicaciones modifiquen el comportamiento de la red, a través de requisitos de alto nivel enviados al controlador. Sin embargo, la aplicación de SDN en entornos IWSN no es directa porque SDN

fue diseñado para los equipos en los centros de datos, que cuentan con múltiples enlaces simultáneos y anchos de banda muy elevados, a diferencia de los equipos de las IWSN que son muy limitados en ancho de banda y en el número de enlaces simultáneos. Por esto, esta tesis se enfoca en un cambio más disruptivo, donde se reemplaza la mayor parte de la pila o *stack* de protocolos convencionales para generar alternativas que permitan acercar las SDN a las redes de sensores.

El uso de SDN en IWSN amplía las posibilidades de control sobre la red y permite mejorar su rendimiento a nivel industrial, ya que ofrece una gestión centralizada con una visión global de los recursos disponibles, donde se pueden crear arquitecturas flexibles y dinámicas. Esto permite aplicar políticas de QoS entre diferentes capas, como el enrutamiento y la planificación de acceso al medio, donde los algoritmos consideran parámetros críticos para la QoS, como el retardo, su variación o *jitter*, el ancho de banda y la pérdida de paquetes. Esta integración y orquestación entre las diferentes capas de red y la gestión centralizada permite aplicar enfoques como el de *slicing*, el cual permite dividir una red física en múltiples segmentos lógicos, cada uno con sus propias características y garantías de QoS. Esto permite que múltiples tipos de tráfico, con diferentes requisitos de rendimiento y fiabilidad, coexistan en la misma red.

La centralización de la información de control en las redes IWSN brinda oportunidades para optimizar diversos parámetros de red, como la selección de canal basada en niveles de señal históricos y el aprendizaje de cambios en la topología, especialmente cuando se trata de elementos en movimiento. En entornos industriales, es común encontrar elementos móviles como vehículos guiados autónomos (AGV) que requieren las mismas garantías de QoS que los nodos fijos. El desafío principal radica en que estos elementos móviles salen constantemente del área de cobertura de los nodos fijos, y los protocolos que utilizan para determinar la pérdida de conexión se basan en temporizadores que exigen la pérdida de paquetes para iniciar el proceso de reconexión. Además, el proceso de reconexión con los protocolos tradicionales implica el intercambio de mensajes hasta que la topología de la red se estabilice, lo que deja sin conexión al nodo móvil. Este proceso se repite cada vez que el nodo móvil se aleja del área de cobertura. Por ello, se han realizado diversos estudios que buscan estimar el desplazamiento del nodo y acelerar los temporizadores para ofrecer una rápida reconexión. No obstante, estos algoritmos son complejos, se ejecutan localmente en los nodos y no llegan a soluciones óptimas.

Otra característica clave de las redes IWSN es su capacidad de desplegarse de forma masiva, que ha sido estudiada en profundidad, demostrando la alta escalabilidad que se logra al utilizar el protocolo de acceso al medio CSMA, sin ofrecer garantías de QoS. Al reemplazar este protocolo de acceso al medio por uno determinista orientado a la QoS, como TSCH, la escalabilidad de la red dependerá de diversos factores como la densidad de nodos, los patrones de tráfico y la planificación de recursos, ya que las transmisiones se realizan en instantes acotados de tiempo en los que solo uno de los nodos puede transmitir. Las planificaciones

autónomas y distribuidas, no están diseñadas para considerar estos parámetros, por lo tanto, con estas se reduce el ancho de banda y condiciona la cantidad máxima de nodos que se pueden desplegar.

Sin embargo, al integrar un enfoque centralizado, como el proporcionado por las redes SDN, es posible mejorar la eficiencia en la reconexión de nodos móviles y la escalabilidad de la red en general. El controlador SDN, al tener una visión global de la red, puede adaptar dinámicamente las rutas y la asignación de recursos, mejorando así el rendimiento en presencia de elementos móviles. Además, con el uso de aplicaciones específicas en el controlador SDN es posible adaptarse a los cambios en la topología de la red de manera más eficiente que con algoritmos locales ejecutados en cada nodo. Al distribuir los recursos de manera óptima y coordinar las transmisiones de manera eficiente, es posible mantener las garantías de QoS al tiempo que se aborda el desafío de la escalabilidad en entornos industriales con altos requisitos de rendimiento y fiabilidad.

1.2. Objetivos

Las investigaciones realizadas para el desarrollo de esta tesis han abordado los desafíos que enfrentan las IWSN en entornos industriales, con el objetivo de garantizar su adaptación y eficiencia en un panorama tecnológico en constante evolución. Estos entornos presentan un creciente aumento del tráfico de red, producto de la convergencia de tecnologías, lo cual demanda una gestión más sencilla y eficiente de los flujos de información. A lo largo de estas investigaciones, se ha identificado que el estado del arte actual en las IWSN presenta dificultades para satisfacer las necesidades emergentes en aspectos cruciales como la calidad de servicio, la movilidad de los nodos y la escalabilidad. Además, se ha observado que los intentos de abordar estos desafíos han dado como resultado una creciente complejidad de los algoritmos que se ejecutan de forma local en los nodos, lo cual afecta el rendimiento y la eficiencia energética, sin lograr resultados óptimos.

Por ello, esta tesis se estructura en torno a los siguientes objetivos que buscan no solo integrar los paradigmas de nueva generación, como SDN y *Slicing*, en las IWSN, sino también complementar estas tecnologías con protocolos de enrutamiento y algoritmos de planificación optimizados. Esto permitirá mejorar la asignación de recursos y el control de acceso al medio, para así aumentar la fiabilidad, flexibilidad y escalabilidad de las IWSN.

- Investigar y desarrollar una arquitectura integrada que combine SDN e IWSN para mejorar la eficiencia y la gestión de recursos en este tipo de redes.
- Desarrollar algoritmos y protocolos adaptativos que optimicen el rendimiento de la red y aseguren el cumplimiento de los parámetros de calidad de servicio. Esto incluye abordar la asignación de recursos, la planificación de rutas y la gestión de prioridades.

- Evaluar y proponer soluciones para abordar los desafíos de la movilidad de nodos y los cambios dinámicos en la topología de la red, manteniendo la estabilidad de la red y la calidad de servicio en entornos IWSN.
- Investigar estrategias para aumentar la escalabilidad y densidad de nodos en redes IWSN TSCH, considerando la capacidad de la red para mantener una alta calidad de servicio mientras se adapta a un mayor número de dispositivos y demandas de tráfico.

1.3. Publicaciones

1.3.1. Artículos en revistas científicas

Clave: Revista indexada JCR	Factor de Impacto: 3.847	Cuartil: Q2
Título: Enhancing SDN WISE with Slicing Over TSCH		
Autores: Orozco-Santos, F.; Sempere-Payá, V.; Albero-Albero, T.; Silvestre-Blanes, J.		
Revista: Sensors		
Editorial: MDPI	Volumen: 21	Páginas:
		Año: 2021
DOI: https://doi.org/10.3390/s21041075		

Clave: Revista indexada JCR	Factor de Impacto: 3.476	Cuartil: Q2
Título: Multicast Scheduling in SDN WISE to Support Mobile Nodes in Industrial Wireless Sensor Networks		
Autores: Orozco-Santos, F.; Sempere-Payá, V.; Albero-Albero, T.; Silvestre-Blanes, J.		
Revista: IEEE Access		
Editorial: IEEE Access	Volumen: 9	Páginas: 141651-141666
		Año: 2021
DOI: https://doi.org/10.1109/ACCESS.2021.3120917		

Clave: Revista indexada JCR	Factor de Impacto: 2.838	Cuartil: Q2
Título: TSCH Multiflow Scheduling with QoS Guarantees: A Comparison of SDN with Common Schedulers		
Autores: Orozco-Santos, F.; Sempere-Payá, V.; Albero-Albero, T.; Silvestre-Blanes, J.		
Revista: Applied Sciences		
Editorial: MDPI	Volumen: 12	Páginas:
		Año: 2022
DOI: https://doi.org/10.3390/app12010119		

Clave: Revista indexada JCR	Factor de Impacto:	Cuartil: Q2
Título: Scalability Enhancement on Software Defined Industrial Wireless Sensor Networks over TSCH		
Autores: Orozco-Santos, F.; Sempere-Payá, V.; Silvestre-Blanes, J.; Vera-Perez, J.		
Revista: IEEE Access		
Editorial: IEEE Access	Volumen: 9	Páginas: 141651-141666
		Año: 2022
DOI: https://doi.org/10.1109/ACCESS.2021.3120917		

1.3.2. Proyectos de investigación

Título del proyecto: HYPERFACTORY -Arquitecturas, herramientas y servicios para la creación de factorías hiperconectadas
Entidad titular: Instituto Tecnológico de Informática
Entidad financiadora: IVACE (IMDEEA/2020/68; IMDEEA/2021/87)
Fecha de inicio: 01/07/2020
Fecha de fin: 30/06/2022

Título del proyecto: DAIS - Distributed Artificial Intelligent System
Entidad financiadora: European Commision; IVACE (101007273; IMAMCN/2021/1)
Fecha de inicio: 01/05/2021
Fecha de fin: 30/04/2024

1.4. Contribución

Las contribuciones realizadas en esta tesis se han centrado en la integración de paradigmas de nueva generación, como SDN y Slicing, en las redes IWSN. Además, complementar estas tecnologías con mejoras en protocolos de enrutamiento y algoritmos de planificación, para optimizar la asignación de recursos en la capa MAC y abordar los desafíos actuales de los entornos industriales.

A continuación, se detallan estas contribuciones:

- Arquitectura de red IWSN y SDN determinista y eficiente para la gestión dinámica de dispositivos, que permite un control preciso y amplio sobre el comportamiento de la red de forma totalmente centralizada.
- Arquitectura de protocolos optimizada para permitir una asignación de recursos TSCH totalmente programable desde una entidad centralizada, considerando los aspectos del tráfico de control para simplificar la propagación de las configuraciones.
- Protocolo de enrutamiento dinámico y multiruta para balancear el tráfico de la red de acuerdo con los requisitos de calidad de servicio y los niveles de congestión previstos en la red.
- Algoritmo de planificación de recursos TSCH dinámico y multicapa para optimizar la asignación de recursos y controlar los instantes de transmisión que permiten garantizar los requisitos de QoS.
- Sistema de etiquetado y gestión de prioridades dinámico, que permite organizar los flujos de tráfico e implementar funcionalidades de QoS utilizando Slicing en redes SW-IWSN con TSCH.
- Aplicación de gestión de recursos desde el plano de aplicación, que permite extender a través de un enfoque multicast las garantías de calidad de servicio a los nodos en movimiento en redes SD-IWSN con TSCH.
- Técnicas para aumentar la cantidad y densidad de nodos que pueden pertenecer a una red SD-IWSN, donde se analizan las limitaciones de una red TSCH, el tráfico de control de SDN y se proponen mecanismos de multiplexación para conservar los estrictos requisitos de QoS.

1.5. Estructura de la tesis

La tesis ha sido estructurada de acuerdo a los desarrollos realizados y los aspectos que motivaron a realizar cada uno de ellos. Estos están dirigidos a dotar las IWSN de un funcionamiento más flexible, dinámico y optimizado, el cual es posible conseguir a través de una gestión centralizada de la red y una pila de protocolos que permita modificar su comportamiento de forma dinámica. El contenido de los capítulos se distribuye de la siguiente forma:

- En el Capítulo 2 se realiza una revisión bibliográfica sobre las IWSN y su adaptación a nivel industrial con los diferentes mecanismos de acceso al medio. Además, se profundiza en la integración con SDN y los retos que afrontan este tipo de redes. Finalmente, se selecciona la arquitectura Software-Defined Wireless Sensor network (SDN-WISE) y el protocolo TSCH como el enfoque principal.
- En el Capítulo 3 se presenta la arquitectura SDN WISE y sus diferentes componentes. Además, se realizan modificaciones a esta arquitectura para conseguir un comportamiento determinista y fácilmente reconfigurable. Estas modificaciones incluyen el cambio de la capa MAC de CSMA a TSCH, el protocolo de enrutamiento, el protocolo de planificación que recibe como entradas las rutas obtenidas por el protocolo de enrutamiento y parámetros de calidad de servicio. Finalmente, el método para encapsular esta información y desplegarla sobre los nodos de la red.
- En el Capítulo 4 se realiza un análisis comparativo entre las diferentes propuestas de planificación de recursos TSCH, incluyendo la propuesta de planificación basada en SDN presentada y desarrollada en el Capítulo 3. La comparativa incluye diferentes escenarios y configuraciones para cada uno de los métodos y se analiza el rendimiento obtenido en aspectos de calidad de servicio.
- En el Capítulo 5 se propone una solución a los problemas de rendimiento que surgen en las IWSN con nodos móviles, denominada Mobile Multicast Forwarding SDN (MMF-SDN). La cual explota el uso de SDN con la planificación centralizada para sincronizar las celdas de recepción de múltiples nodos.
- En el Capítulo 6 se estudian las limitaciones de escalabilidad que tienen las IWSN con SDN. Para abordar este reto, se propone el uso de virtual sinks, que permiten agrupar lógicamente nodos físicos para aumentar el ancho de banda del nodo sink.
- En el Capítulo 7 se presentan las conclusiones finales teniendo en cuenta cada uno de los desarrollos realizados y el estado del arte actual. Además, se plantean posibles trabajos futuros que se basan en la arquitectura SD-IWSN.

La tabla 1-1 muestra la relación entre los objetivos, las contribuciones, los capítulos y las publicaciones realizadas durante el desarrollo de esta tesis.

Objetivos	Contribuciones	Capítulo	Publicación
Investigar y desarrollar una arquitectura integrada que combine SDN e IWSN para mejorar la eficiencia y la gestión de recursos en este tipo de redes.	Arquitectura de red IWSN y SDN determinista y eficiente para la gestión dinámica de dispositivos, que permite un control preciso y amplio sobre el comportamiento de la red de forma totalmente centralizada.	CAP 3	Enhancing SDN WISE with Slicing Over TSCH
	Arquitectura de protocolos optimizada para permitir una asignación de recursos TSCH totalmente programable desde una entidad centralizada, considerando los aspectos del tráfico de control para la propagación óptima de las configuraciones.		
Desarrollar algoritmos y protocolos adaptativos que optimicen el rendimiento de la red y aseguren el cumplimiento de los requisitos de calidad de servicio. Esto incluye abordar la asignación de recursos, la planificación de rutas y la gestión de prioridades.	Protocolo de enrutamiento dinámico y multiruta para balancear el tráfico de la red de acuerdo con los requisitos de calidad de servicio y los niveles de congestión previstos en la red.	CAP 4	TSCH Multiflow Scheduling with QoS Guarantees: A Comparison of SDN with Common Schedulers
	Algoritmo de planificación de recursos TSCH dinámico y multicapa para optimizar la asignación de recursos y controlar los instantes de transmisión que permiten garantizar los requisitos de QoS.		
	Sistema de etiquetado y gestión de prioridades dinámico, que permite organizar los flujos de tráfico e implementar funcionalidades de QoS utilizando Slicing en redes SW-IWSN con TSCH		
Evaluar y proponer soluciones para abordar los desafíos de la movilidad de nodos y los cambios dinámicos en la topología de la red, manteniendo la estabilidad y la calidad de servicio en entornos IWSN.	Aplicación de gestión de recursos desde el plano de aplicación, que permite extender a través de un enfoque multicast las garantías de calidad de servicio a los nodos en movimiento en redes SD-IWSN con TSCH.	CAP 5	Multicast Scheduling in SDN WISE to Support Mobile Nodes in Industrial Wireless Sensor Network
Investigar estrategias para aumentar la escalabilidad y densidad de nodos en redes IWSN TSCH, considerando la capacidad de la red para mantener una alta calidad de servicio mientras se adapta a un mayor número de dispositivos y demandas de tráfico.	Técnicas para aumentar la cantidad y densidad de nodos que pueden pertenecer a una red SD-IWSN, donde se analizan las limitaciones de una red TSCH, el tráfico de control de SDN y se proponen mecanismos de multiplexación para conservar los estrictos requisitos de QoS.	CAP 6	Scalability Enhancement on Software Defined Industrial Wireless Sensor Networks over TSCH

Tabla 1-1: Correspondencia entre los objetivos y contribuciones con los capítulos.

Capítulo 2

Estado del arte

El concepto del IoT ha experimentado un crecimiento exponencial debido a su facilidad de integración en una amplia variedad de entornos centrándose en transformar objetos cotidianos en dispositivos interconectados. Esta rápida expansión y adopción masiva refleja la versatilidad y el potencial del IoT para adaptarse a diferentes contextos y satisfacer diversas necesidades, es el caso de e-Health [13] o Smartcity [14]. Incluso, la expansión del IoT está facilitando de manera significativa la transición hacia la Industria 4.0, que se basa en la capacidad de recolectar y procesar información de su entorno, para la ejecución de acciones, que pueden ser locales o remotas. De esta manera, se fomenta un ecosistema industrial inteligente y adaptable, donde la optimización en la toma de decisiones se logra gracias al procesamiento de información. Sin embargo, cada uno de estos entornos tiene requisitos de operación diferentes, lo que da lugar a un ecosistema IoT compuesto por una gran diversidad de dispositivos y protocolos específicos para satisfacer los requisitos de cada aplicación. Aunque el IoT contempla redes cableadas e inalámbricas, el empleo de tecnologías inalámbricas es el más extendido debido a su flexibilidad, ya que permite adaptarse de forma rápida y eficiente a la dinámica del entorno. Además, la implementación de soluciones inalámbricas disminuye significativamente los costos de infraestructura, así como los tiempos de instalación y mantenimiento en comparación con la red cableada. La diversidad de protocolos inalámbricos usados en IoT se puede clasificar en función de dos parámetros principales: el rango de cobertura y la velocidad de transmisión como se muestra en la figura **2-1**.

Tomando en cuenta estos dos parámetros, se pueden segmentar los tipos de redes en diferentes categorías. Por ejemplo, las redes de baja potencia (LP - *Low Power*) son aquellas que buscan minimizar el consumo de energía, normalmente a expensas de reducir el rendimiento como por ejemplo una menor tasa de transferencia. Además, según el rango de cobertura, podemos diferenciar redes que abarcan desde decenas de metros, como las redes inalámbricas de área personal (WPAN - *Wireless Personal Area Networks*), hasta redes que alcanzan decenas de kilómetros, como las redes inalámbricas de área amplia (WWAN - *Wireless Wide Area*

Networks). Esta clasificación tiene un impacto directo en la implementación y eficiencia de las soluciones IoT, permitiendo ajustar la tecnología a los requisitos específicos de cada aplicación y entorno.

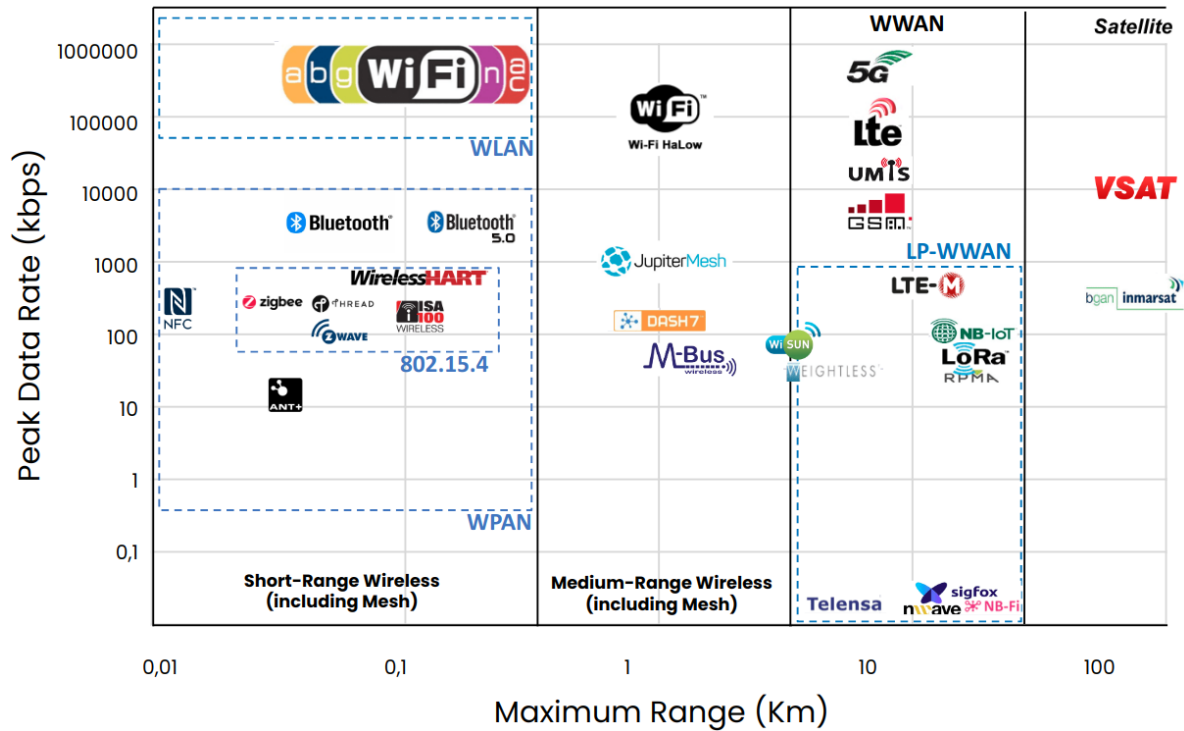


Figura 2-1: Tecnologías de comunicación inalámbrica [15]

2.1. WPAN

Las Redes de Área Personal Inalámbrica (WPAN - Wireless Personal Area Network) están diseñadas para permitir la interacción entre dispositivos a distancias cortas, generalmente menos de 100 metros para una comunicación directa de un solo salto entre dispositivos. Sin embargo, mediante topologías más complejas que reenvían el tráfico de un nodo a otro, es posible extender el alcance efectivo de estas redes más allá de la limitación de los 100 metros, ofreciendo flexibilidad en su implementación y conectividad. Entre este tipo de redes se encuentran tecnologías como Bluetooth, Zigbee, 6LowPAN, WirelessHart, ISA100.1, Thread, NFC y Ultra-Wideband (UWB), cada una con características propias que les permiten adaptarse a escenarios específicos.

Estas redes se componen de dos o más dispositivos que pueden ser dispositivos de funcionalidad completa (FFD - *Full Function Device*) o dispositivos de funcionalidad reducida (RFD - *Reduced Function Device*). Los FFD pueden operar como coordinador PAN que se encarga de configurar parámetros de transmisión, a diferencia de los RFD que tienen un

funcionamiento limitado y que únicamente pueden establecer un enlace con un FFD [16]. Al desplegar múltiples dispositivos se conforman diferentes topologías o estructuras que determinan la organización y la interconexión de los nodos de la red. La topología adoptada tiene implicaciones significativas en el rendimiento de la red, su escalabilidad, la gestión de la energía y la tolerancia a los fallos, los tipos más comunes de topologías pueden observarse en la figura 2-2.

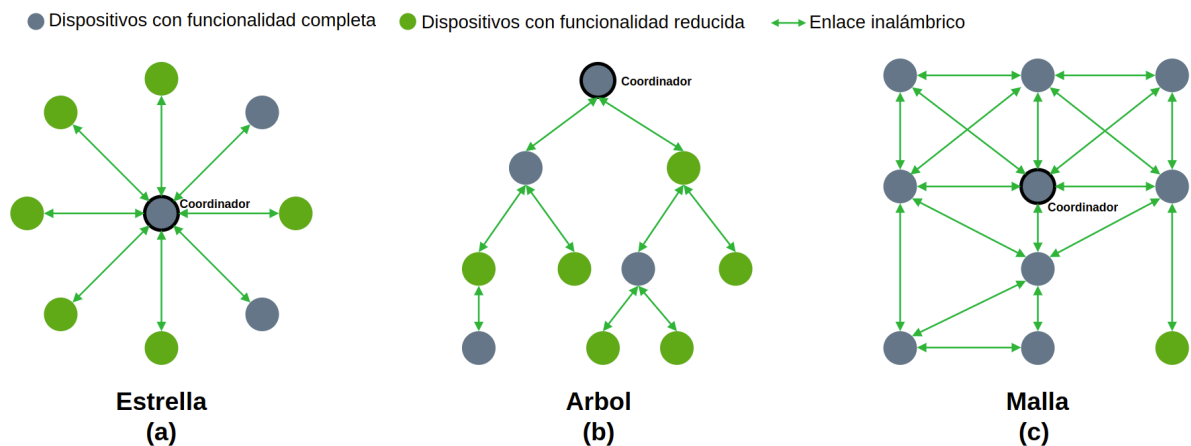


Figura 2-2: Topologías típicas en WPAN

La topología en estrella, es uno de los modelos estructurales más comunes en las WPAN, opera con un nodo central o coordinador que maneja las comunicaciones entre los demás nodos de la red como se muestra en la figura 2-2(a). Se utiliza ampliamente en tecnologías como Bluetooth y Zigbee, donde la simplicidad y la eficacia en la gestión de la congestión de la red son críticas. Sin embargo, esta topología presenta la limitación de que si el nodo coordinador falla, la operatividad de toda la red se ve comprometida [17].

Otra estructura es la topología de árbol, una variante de la topología de malla o red mallada, donde se establecen restricciones en las conexiones para formar una jerarquía. Esta estructura es útil en aplicaciones que requieren control jerárquico, como los sistemas de sensores IoT y corresponde a la mostrada en la figura 2-2(b). Sin embargo, su mayor limitación es la dependencia del nodo raíz o padre, si este falla, interrumpe la comunicación de los nodos inferiores.

Por último, la topología de malla, donde cada nodo puede comunicarse con cualquier otro nodo dentro de su alcance como en la figura 2-2(c). En esta estructura, la robustez es su principal ventaja, si un nodo falla, los demás nodos pueden redirigirse a través de rutas alternativas. Esta topología es esencial en tecnologías como Zigbee, 6LowPAN y Bluetooth, que buscan garantizar la comunicación continua en entornos de IoT. Sin embargo, la implementación y gestión de las topologías de malla tiene una mayor complejidad, ya que requiere algoritmos de enrutamiento eficientes capaces de adaptar dinámicamente las rutas de red.

Es posible orquestar parámetros de la red para crear una topología híbrida, que combine dos de las dos topologías anteriores, como podría ser la topología en estrella-malla, donde se conforman agrupaciones o clusteres de red que operan en estrella. Aquí, varios coordinadores en estrella están interconectados en una topología de malla, lo que permite una mayor flexibilidad y robustez en comparación con la topología en estrella pura. Aunque más compleja, esta topología permite una mejor gestión de la congestión y una mayor escalabilidad [17].

2.1.1. IEEE 802.15.4

El estándar IEEE 802.15.4 [15] define las características y control de acceso al medio para redes inalámbricas de área personal con baja tasa de datos y baja potencia (LP-WPAN - *Low Power Wireless Personal Area Network*). En este estándar podemos agrupar tecnologías como Zigbee, 6LoWPAN, ISA100.1, WirelessHart y Thread, ya que comparten la capa física y de control de acceso al medio como puede observarse en la figura 2-3. Estos protocolos mantienen una velocidad de transferencia de datos relativamente baja, hasta 250 kbps, permitiéndoles minimizar el consumo de energía y operar con baterías durante largos periodos de tiempo. Las características principales y diferencias entre estas tecnologías se observan en la tabla 2-1.

	ZigBee	6LoWPAN	ISA100.11a	WirelessHART
Topología	Estrella, Malla y Árbol	Estrella y malla	Estrella y malla	Malla
Seguridad	AES 128-bit	Varía (IPSec, etc.)	Varía (AES, etc.)	AES 128-bit
Tipo de datos	Periódico	Periódico y aperiódico	Periódico y aperiódico	Periódico y aperiódico
Soporte IPv6	No	Sí	Sí	No
Enrutamiento	Proactivo	Reactivo	Proactivo	Proactivo
Fiabilidad	Alta	Alta	Muy alta	Muy alta
Densidad de nodos	Alta	Alta	Alta	Alta
Retardo de red	Medio	Medio	Bajo	Bajo
Interoperabilidad	Baja	Alta	Medio	Medio
Escalabilidad	Alta	Alta	Alta	Alta
Tiempo de vida de la batería	Alta	Media	Alta	Alta

Tabla 2-1: Comparación de protocolos inalámbricos basados en 802.15.4

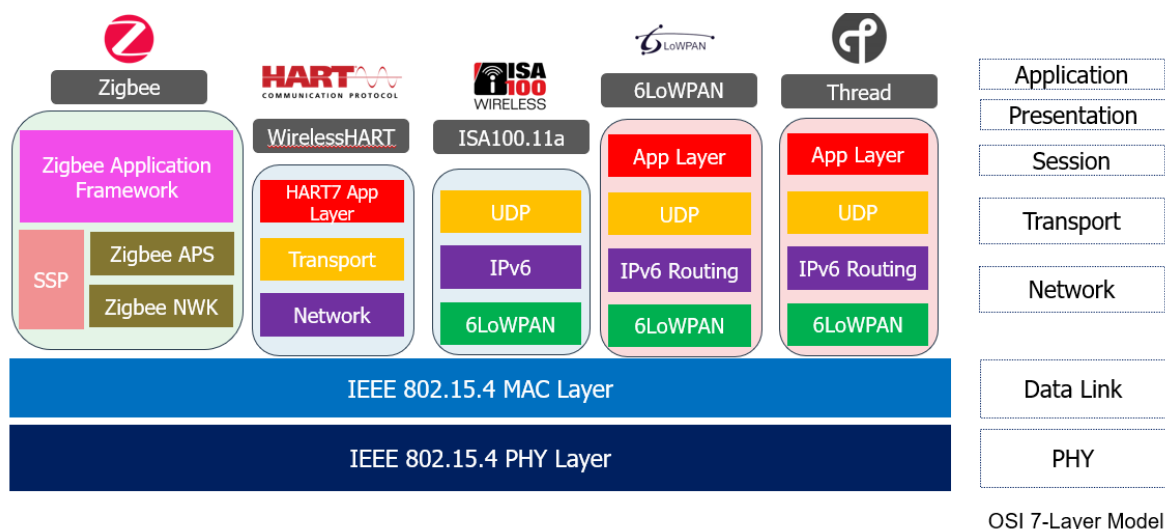


Figura 2-3: Comparación de la pila de protocolos en diferentes tecnologías de WPAN

Zigbee

El estándar Zigbee está orientado a redes de bajo consumo, baja tasa de transmisión y bajo coste, su pila de protocolos se desarrolla sobre la capa física y el control de acceso al medio definido en IEEE 802.15.4. Zigbee está enfocado en formar redes autosuficientes que se gestionan y reconfiguran de forma completamente distribuida y autónoma [18], soportando redes malladas, estrella y árbol. Para la gestión de la red tiene definidos tres tipos de nodos: Coordinador Zigbee, enrutador Zigbee y nodo final.

El coordinador tiene las funciones de iniciar y mantener la comunicación con los demás nodos de la red, además de configurar los parámetros de operación. Para redes con topología en estrella los nodos finales se comunican directamente con el coordinador Zigbee y es posible tener una red con hasta 255 dispositivos. A diferencia de las redes con topología mallada o en árbol, donde los enrutadores Zigbee envían información de control y datos, usando un protocolo de enrutamiento jerárquico que divide la red en subredes. En estas subredes cada nodo puede retransmitir datos, extendiendo el alcance de la red y proporcionando redundancia en caso de fallo de un nodo, por ello en este tipo de redes, es posible agregar hasta 65.000 dispositivos [18].

La transmisión de información se realiza con un esquema de acceso múltiple por detección de portadora con evasión de colisiones (CSMA-CA - *Carrier Sense Multiple Access with Collision Avoidance*), donde los nodos finales inician la transmisión cuando tienen información disponible, espera la confirmación de recepción (ACK) y en caso de no recibirla realiza una retransmisión. La pila de protocolos puede observarse en la figura 2-3, donde se tiene sobre la capa física y MAC 802.15.4 la capa de red (NWK) y de aplicación (APL). Dentro de la capa de red, se tienen los diferentes subsistemas que realizan la elección del coordinador y

los enrutadores Zigbee.

6LoWPAN

El estándar IPv6 sobre redes inalámbricas personales de baja potencia (6LoWPAN - *IPv6 over Low-Power Wireless Personal Area Networks*), fue diseñado para mejorar la interoperabilidad y la integración de dispositivos LP-WPAN al proporcionar soporte nativo para direccionamiento IPv6 en su pila de protocolos. El uso de direccionamiento IP permite una mayor cantidad de dispositivos en la red y posibilita una integración más fluida con otro tipo de redes basadas en IP.

Para ello, implementa mecanismos que permiten comprimir el encabezado de IPv6, que normalmente requiere de 48 bytes en UDP y 60 en TCP a 6 y 8 bytes respectivamente. Estos mecanismos de compresión permiten adaptar el protocolo IPv6 al tamaño máximo de paquete definido en el estándar 802.15.4 que es de 127 Bytes. Además, especifica la encapsulación, fragmentación y el encabezado para la red mallada.

Estos pueden verse en la figura 2-4 [19], donde se muestran las funciones específicas y técnicas que rigen el estándar 6LoWPAN. La encapsulación es la distribución para que los datagramas IPv6 sean comprimidos en una estructura que se acomoda al protocolo 802.15.4. Este proceso aprovecha el campo de carga útil de los paquetes 802.15.4 para transportar datagramas IPv6, respetando el tamaño de la unidad máxima de transmisión impuesto por el estándar 802.15.4.

Adicionalmente, el proceso de fragmentación es la estrategia que 6LoWPAN utiliza para gestionar datagramas IPv6 que exceden el tamaño máximo del paquete 802.15.4. Este proceso involucra la división del datagrama original en múltiples fragmentos, cada uno con su propio encabezado de fragmentación, para ser transmitidos de manera individual a través de la red 802.15.4.

Por último, el encabezado de red mallada es una estructura especial definida en 6LoWPAN para gestionar el enrutamiento en una topología de red mallada. Este encabezado proporciona la información necesaria para la retransmisión de paquetes entre nodos, permitiendo que los paquetes puedan seguir diferentes caminos a través de la red para llegar a su destino.

Thread

Thread es un protocolo basado en estándares abiertos enfocado en la seguridad de la red, utiliza las capas de 6LoWPAN. Por ello, la capa física y de acceso al medio son las que se encuentran en IEEE 802.15.4, además, tiene soporte para el direccionamiento IPv6. La diferencia con otras tecnologías se encuentra en los protocolos de enrutamiento, transporte y aplicación, sobre los que se añade una capa de seguridad y puesta en marcha.

Las redes Thread, forman topologías malladas, por lo tanto, los mensajes pueden tomar

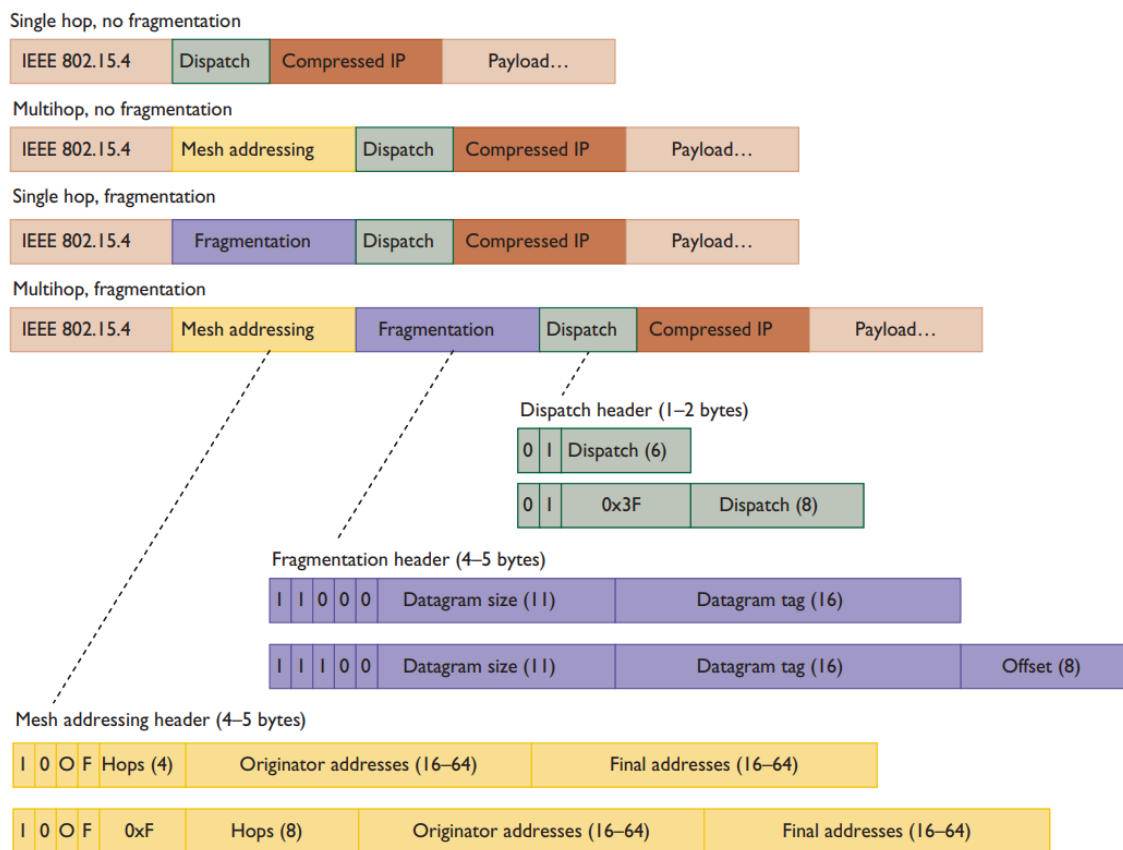


Figura 2-4: Encabezados y fragmentación 6LoWPAN

múltiples rutas a través de la red para llegar a su destino, mejorando así la robustez de la red y su capacidad de recuperación ante fallos. Además, utiliza el protocolo de enrutamiento RPL en modo no almacenante (*non-storing mode*), lo que permite un enrutamiento escalable y eficiente en redes de IoT, especialmente en situaciones donde los nodos pueden tener recursos limitados. La formación de esta topología es dinámica y autónoma, con dispositivos capaces de auto-configurarse y autenticarse, simplificando la incorporación de nuevos dispositivos a la red. Cuando un dispositivo se enciende por primera vez, puede elegir automáticamente convertirse en coordinador si no hay ninguna red existente a la que unirse. Este coordinador es responsable de gestionar aspectos como la asignación de direcciones, la gestión de enrutamiento y el mantenimiento de la conectividad de la red.

Estas redes están compuestas por cuatro tipos de dispositivos: enrutador de frontera (BR - *Border Router*) que actúa como una pasarela entre la red Thread y otro tipo de redes IP; enrutadores (*Routers*), que son dispositivos interconectados con múltiples enlaces, los cuales mantienen la radio encendida de forma permanente; dispositivos finales (SED - *Sleepy End Devices*) que están enlazados con un único enrutador padre, son los únicos que pueden operar en modo de bajo consumo; y finalmente, dispositivos finales con capacidad de enruta-

miento (REED - *Router Eligible End Device*), los cuales modifican su comportamiento entre enrutador y dispositivo final según las necesidades de la red [20].

El enfoque principal de Thread es ofrecer características sólidas de seguridad en la capa de red, incluyendo el cifrado y la autenticación. Para garantizar la seguridad de las comunicaciones utiliza el algoritmo de cifrado AES (*Advanced Encryption Standard*) con una longitud de clave de 128 bits. También implementa la comisión o puesta en marcha de dispositivos a través de J-PAKE (*Password Authenticated Key Exchange by Juggling*) o la asignación de una PSKc (*Pre-Shared Key for Commissioning*), permitiendo la autenticación segura de los dispositivos en la red [21].

WirelessHart

WirelessHART es una tecnología de comunicación diseñada para uso en procesos industriales, está basado en el protocolo de comunicación HART (*Highway Addressable Remote Transducer*). Este protocolo utiliza una topología de red mallada sobre la que se proporciona un cifrado robusto y administración de claves, asegurando la integridad y la confidencialidad de los datos.

El control de acceso al medio en WirelessHART está basado en un esquema de división de tiempo conocido como acceso múltiple por división de tiempo (TDMA - *Time Division Multiple Access*). Este esquema asigna a cada dispositivo en la red una o más ranuras de tiempo específicas durante las cuales puede transmitir o recibir datos. En el TDMA, cada ciclo de comunicación se denomina supertrama, donde cada supertrama se divide en un número fijo de ranuras de tiempo. Este enfoque minimiza la probabilidad de colisiones de paquetes que se producen cuando dos dispositivos intentan transmitir simultáneamente, ya que cada dispositivo conoce su ventana de transmisión asignada.

WirelessHART también incorpora un algoritmo de salto de canal. En cada ranura de tiempo, los dispositivos pueden cambiar a un canal diferente según un patrón predefinido. El salto de canal ayuda a mitigar la interferencia y a mejorar la robustez de la red, ya que, si un canal en particular experimenta interferencias, los dispositivos pueden comunicarse a través de otros canales. Esto es particularmente útil en entornos industriales, donde los obstáculos y las fuentes de interferencia pueden variar con el tiempo [22].

La combinación de TDMA con el salto de canal consigue la multiplexación de las comunicaciones, permitiendo que diferentes dispositivos transmitan o reciban datos simultáneamente en diferentes canales sin interferirse. Para ello, incluye un protocolo de gestión de red encargado de la formación, mantenimiento y optimización de la red. Este protocolo gestiona tareas como la selección de ruta, la sincronización de la red y la gestión de la energía. Por ejemplo, el enrutamiento se gestiona mediante un algoritmo basado en grafos. Cada dispositivo mantiene un grafo de vecinos que describe las conexiones con los dispositivos en el rango de cobertura. Estos grafos se utilizan para calcular las rutas más eficientes para el

tráfico de red. Los dispositivos también pueden cambiar dinámicamente su grafo de vecinos en respuesta a cambios en la red, como la adición o eliminación de dispositivos, o la variación en la calidad del enlace.

Debido a su enfoque industrial, la seguridad es un aspecto crítico en las redes WirelessHART, incorporando características de seguridad desde la capa física hasta la capa de aplicación. Todos los mensajes son cifrados utilizando AES con una clave de 128 bits. Además, el protocolo autentica a los dispositivos antes de su incorporación a la red. La gestión de claves se lleva a cabo mediante un gestor de seguridad de red, que es responsable de generar y distribuir las claves de cifrado, las cuales pueden actualizarse dinámicamente. Para asegurar la autenticidad y la integridad de los datos, cada mensaje incluye un código de autenticación de mensajes (MAC - *Message Authentication Code*) basado en la clave y el contenido del mensaje [23].

ISA100.1

Es un estándar industrial desarrollado por la sociedad internacional de automatización (ISA - *International Society of Automation*) el cual proporciona una red inalámbrica segura y robusta para transmisiones de datos industriales. Como WirelessHART, se enfoca en el entorno industrial, priorizando la seguridad y fiabilidad. También utiliza TDMA y saltos de frecuencia para optimizar determinismo y fiabilidad en las transmisiones.

Sin embargo, en términos de configuración de red, ISA 100.11a es más flexible. Esto se debe a que adopta un enfoque de asignación de roles de enrutamiento basado en las necesidades o características locales de los dispositivos, como el nivel de energía. Por lo tanto, no requiere que todos los dispositivos de campo sean enrutadores, lo que ayuda a prolongar la vida útil de los dispositivos con batería.

Además, la norma ISA 100.11a ofrece una mayor flexibilidad en la configuración del timeslot, que puede adaptarse a las necesidades de la red y las condiciones del entorno. A diferencia de WirelessHART, que usa un tiempo de ranura fijo de 10 ms, ISA 100.11a permite configurar dinámicamente los timeslots en un rango de 8 a 14 ms [10].

Otra diferencia clave entre ISA 100.11a y WirelessHART es el uso de 6LowPAN, lo que le permite formar un backbone basado en IPv6 y segmentar subredes inalámbricas. Esto mejora la compatibilidad con otras redes y dispositivos, facilitando la integración de la red inalámbrica industrial con otros componentes de la infraestructura. El enrutamiento en ISA 100.11a utiliza diferentes enfoques dependiendo del contexto. En una subred inalámbrica, el enrutamiento se realiza en la capa de enlace de datos utilizando una dirección de subred corta. En el backbone de la red, el enrutamiento se realiza en la capa de red utilizando IPv6.

2.1.2. Bluetooth

Bluetooth es una tecnología de red que ha impactado significativamente la comunicación inalámbrica de corto alcance, siendo el protocolo WPAN más reconocido. Con la llegada de Bluetooth 5.0, se ha mejorado la velocidad de transferencia de datos hasta 2 Mbps y el alcance de la red. Además, se ha extendido con *Bluetooth Low Energy* (BLE) para ofrecer conectividad en el IoT, donde proporciona una alta eficiencia energética. Está basado en la técnica de espectro disperso por salto de frecuencia (FHSS - *Frequency-Hopping Spread Spectrum*), y opera en la banda de 2,4 GHz. Esta banda se divide en 79 canales de 1 MHz cada uno y los dispositivos que utilizan Bluetooth cambian de canal 1600 veces por segundo siguiendo un patrón pseudoaleatorio para evitar la interferencia co-canal y la intermodulación [24].

Para el control de acceso al medio utiliza un esquema de división de tiempo dúplex (TDD, - Time-Division Duplex), donde un único canal de frecuencia se divide en intervalos de tiempo para permitir la transmisión y recepción alternativa de paquetes de datos entre el dispositivo maestro y el esclavo. El esquema TDMA también se utiliza para permitir la comunicación entre un maestro y varios esclavos en una red de topología en estrella denominada piconet. Un maestro puede controlar hasta siete esclavos activos en un piconet y varios piconets pueden entrelazarse para formar una scatternet [24].

En términos de seguridad, Bluetooth implementa una serie de medidas, incluyendo la autenticación y el cifrado. La autenticación se basa en un secreto compartido (la clave de enlace) y utiliza el algoritmo de autenticación de clave de enlace E0. El cifrado se realiza mediante el algoritmo de flujo de cifrado E0, que es un cifrado de flujo basado en LFSR (*Linear Feedback Shift Register*). A partir de la versión 2.1, Bluetooth también incorporó el protocolo de emparejamiento seguro y sencillo (SSP - *Secure Simple Pairing*) para simplificar el emparejamiento de dispositivos y mejorar la seguridad. En la tabla **2-2** se presenta una comparativa de las diferentes versiones de Bluetooth.

2.2. IoT en la industria

El papel del IoT en la industria es indiscutible, ha emergido como una herramienta clave que permite una visión detallada y un entendimiento más profundo de los procesos de producción a través de la recopilación de datos eficiente y de bajo costo. Sin embargo, uno de los aspectos más destacados del IoT en la industria es su naturaleza mínimamente invasiva: estos dispositivos pueden integrarse en los procesos existentes sin interrumpirlos o alterarlos, una característica crucial en estos entornos, donde cualquier cambio podría tener un impacto significativo en el rendimiento y los costos de producción. Además, posibilita la conexión de procesos industriales aislados que no pueden intercambiar información, lo cual facilita una visión global del sistema y aumentar las variables a los procesos productivos para hacerlos

	1.0 - 1.0B	2.0 + EDR	3.0 + HS	4.0	5.0	5.2
Velocidad de datos	1 Mbps	3 Mbps	24 Mbps	1 Mbps (LE)	2 (LE)	2 Mbps (LE)
Alcance (m)	10	30	100	60	200	400
Máx. dispositivos	8	8	8	8	8	8
Modo Dual	No	No	Sí	Sí	Sí	Sí
Cifrado	No	Sí	Sí	Sí	Sí	Sí
Tecnología base	FHSS	FHSS	FHSS	FHSS	FHSS	FHSS
Potencia (mW)	1	2.5	10	2.5	2.5	2.5

Tabla 2-2: Comparación de las diferentes versiones de Bluetooth

más eficientes, integrados y flexibles [25].

A pesar de sus ventajas, los dispositivos IoT convencionales no satisfacen las demandas específicas de la industria, especialmente en términos de fiabilidad, seguridad y tolerancia a fallos. Para cubrir estas necesidades, ha surgido una rama especializada dentro del IoT: el Industrial IoT (IIoT), el cual comprende dispositivos y protocolos más robustos, diseñados específicamente para adaptarse a las rigurosas condiciones y cumplir con los requisitos del entorno industrial. El IIoT se distingue del IoT convencional por su énfasis en la interconexión máquina a máquina (M2M - *Machine to Machine*). Esta orientación mejora la comunicación entre las máquinas, lo que puede dar como resultado un aumento de la productividad y la eficiencia [3].

2.2.1. Calidad de servicio (QoS)

La convergencia de la información y la necesidad de que esta esté disponible para otros servicios ha causado un aumento significativo del tráfico de red, y en la diversidad de requisitos de los flujos. Este incremento en el tráfico puede suponer congestión, retrasos en el envío e incluso pérdida de paquetes. Estos son parámetros que deterioran la calidad del servicio. Aunque la solución común en los entornos industriales es aislar los procesos críticos para eliminar todas las fuentes de interferencia, el paso hacia una Industria más conectada y dinámica implica que esto ya no es viable.

Por ello, otra alternativa menos restrictiva es gestionar los flujos de información mediante políticas de calidad de servicio, las cuales garantizan los requisitos para cada tipo de tráfico

de extremo a extremo. Estas políticas mejoran la utilización de los recursos de la red y permiten que el uso de recursos esté priorizado para los diferentes tipos de tráfico. Por ejemplo, el tráfico de voz, video y aplicaciones de datos importantes puede ser procesado con prioridad.

Parámetros de QoS

Por ello, durante el desarrollo de esta tesis se considera a la calidad de servicio como un conjunto de parámetros que permiten evaluar el rendimiento de la red. Por ello, es esencial conocer los factores que la afectan: ancho de banda, throughput, latencia, jitter y tasa de pérdida de paquetes.

- **Ancho de Banda:** Es la cantidad máxima de datos que pueden transmitirse entre dos puntos en un período de tiempo específico. Expresado en bits por segundo (bit/s), es un indicador clave de la capacidad de transmisión y de las limitaciones de un canal de comunicación.
- **Throughput:** Es la cantidad real de datos transmitidos a través de un canal de comunicación. Es generalmente menor que el ancho de banda debido a que se ve afectado por los parámetros de los protocolos y otras limitaciones de la red, como la latencia y la pérdida de paquetes.
- **Latencia:** Es el tiempo requerido para que un paquete viaje desde el emisor hasta el receptor, incluyendo el tiempo de transmisión y procesamiento.
- **Jitter:** Describe la variación en el tiempo de transmisión de paquetes en la misma conexión. Es crítico en transmisiones en tiempo real como voz y video, ya que puede causar interrupciones.
- **Tasa de Pérdida de Paquetes:** Es la proporción de paquetes perdidos en comparación con el total transmitido.

Además, se han considerado parámetros adicionales que permiten evaluar la integración entre los diversos componentes. Aunque estos no afectan de forma directa la calidad del servicio, simplifican la administración de la red, permitiendo que esta se adapte con mayor facilidad a los requisitos específicos.

- **Flexibilidad:** La capacidad de un sistema para adaptarse a diferentes condiciones, necesidades y requisitos operativos sin una degradación significativa del rendimiento. Esto es esencial para mantener la eficiencia y la calidad del servicio en entornos dinámicos y cambiantes.
- **Capacidad de reconfiguración:** Complejidad de la red o sus dispositivos para modificar su configuración y comportamiento sin la necesidad de una intervención física o interrupción del servicio.

- **Determinismo:** Es la capacidad de la red para operar de manera predecible y consistente, garantizando que las transmisiones se realicen en un orden específico y dentro de tiempos definidos.

Modelos de Servicio QoS

Para mejorar la calidad del servicio, se han definido varios modelos QoS [26]:

- **Best-Effort:** Es el modelo por defecto en Internet, donde no se garantiza el rendimiento en términos de retraso y fiabilidad, adecuado para servicios con bajos requisitos de calidad.
- **Integrated Service (*IntServ*):** Utiliza un protocolo de señalización para reservar recursos específicos en la red antes de la transmisión de datos. Cada flujo de datos mantiene una reserva de recursos en la red para garantizar el rendimiento.
- **Differentiated Service (*DiffServ*):** Clasifica los paquetes y les asigna diferentes prioridades para que así cada paquete reciba un tratamiento individualizado salto a salto. No requiere señalización previa y es el modelo más utilizado actualmente, proporcionando servicios diferenciados basados en parámetros QoS en los paquetes.

Mecanismos en el Modelo DiffServ

Este modelo incluye los siguientes mecanismos:

- **Clasificación y Marcado de Tráfico:** Divide los paquetes en diferentes clases o asigna prioridades. Se implementa mediante clasificadores de tráfico y mapeo de prioridades.
- **Funciones policía:** Controlan la tasa de tráfico y el tamaño de las ráfagas dentro de un límite de ancho de banda.
- **Control de la congestión:** El control de congestión utiliza colas para ordenar el reenvío de paquetes, y evitar la congestión. Además, monitoriza el uso de recursos para mitigar la sobrecarga.

2.2.2. Redes de sensores inalámbricos industriales (IWSN)

Las IWSN, constituyen uno de los principales recursos en el IIoT [27], estas redes están compuestas por múltiples nodos orientados específicamente a la recolección, procesamiento y envío de información. Son dispositivos de baja potencia y bajo costo cuyo objetivo general es reducir el tiempo de despliegue y el coste en comparación a lo que se necesitaría para las redes cableadas manteniendo un nivel adecuado de calidad de servicio [28]. Debido a su tamaño y operación generalmente con baterías, son dispositivos que tienen grandes limitaciones de procesamiento y memoria [5]. Aunque esto, también les permite ser muy flexibles, fáciles de

reemplazar y adaptar a ambientes industriales. Las IWSN se han ido refinando con diversos protocolos que buscan una mejor adaptación a las necesidades industriales, inicialmente el estándar proponía como protocolo de control de acceso al medio el protocolo de acceso múltiple por detección de portadora con evasión de colisiones (*CSMA-CA* - Carrier Sense Multiple Access with Collision Avoidance). Sin embargo, para mejorar esta adaptación el estándar fue actualizado al IEEE 802.15.4e, donde se incorporan tres nuevos protocolos orientados al uso industrial: *Low Latency Deterministic Network* (LLDN), *Deterministic and Synchronous Multi-channel Extension* (DSME) and *Time Slotted Channel Hopping* (TSCH). Cada uno de estos mecanismos fue definido para satisfacer determinadas necesidades de diferentes casos de uso, como reducir el consumo de energía, reducir la latencia a través de redes en estrella, aumentar la fiabilidad de las transmisiones o lograr un mayor determinismo en la red al eliminar gran parte de la aleatoriedad que presenta el CSMA-CA.

2.2.3. Control de Acceso al medio en redes 802.15.4e

LLDN

Este protocolo, de naturaleza determinista para topologías en estrella, ofrece un entorno de transmisión de datos altamente predecible y organizado, permitiendo a las WSN adaptarse a aplicaciones críticas en tiempo y con requisitos exigentes de latencia, se caracteriza por su operación libre de contienda, es decir, los nodos no compiten por el acceso al canal de comunicación. Debido a que cada nodo tiene una ventana temporal específica para la transmisión de sus datos, lo cual garantiza la entrega oportuna de información y evita la colisión de tramas. Al ser un protocolo basado en TDMA, contiene una estructura que se denomina supertrama, en LLDN se divide en dos segmentos distintos: el Periodo Libre de Contienda (CFP - *Contention-Free Period*) y el Periodo Basado en Contienda (CBP - *Contention-Based Period*). El CFP se emplea para la transmisión de datos que requieren una entrega inmediata, con requisitos temporales altos, mientras que el CBP se utiliza para los datos que admiten un retardo más prolongado.

Además, el CFP se subdivide en dos subtramas: la Trama de Señalización (BF) y la Trama de Datos (DF). La BF tiene la función de sincronizar los nodos dentro de la red, además de distribuir a la planificación correspondiente al CFP. Por otro lado, la DF se destina a la transmisión de datos durante el CFP.

DSME

Este protocolo combina las capacidades del TDMA con las ventajas de acceso múltiple por división de frecuencias (FDMA - *Frequency Division Multiple Access*) para proporcionar un enlace de comunicación robusto y eficiente. El funcionamiento del DSME se basa en la organización de las comunicaciones en una estructura de multi-supertramas, las cuales están compuestas por múltiples supertramas como en la figura 2-5. Cada supertrama está com-

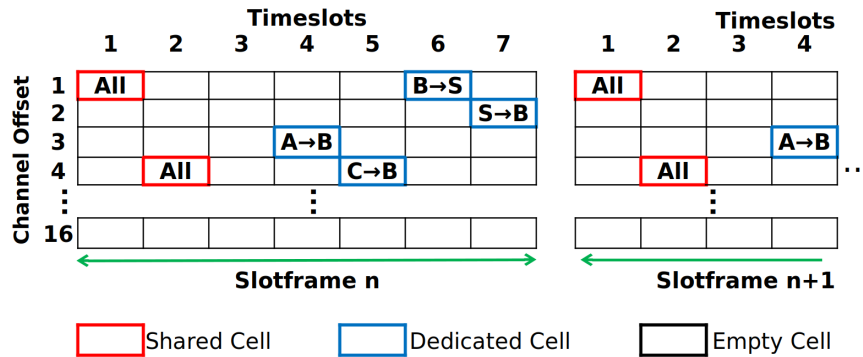


Figura 2-6: Macrotrama o *Slotframe* del protocolo TSCH

embargo, la relación entre channel offset y la frecuencia o canal físico no es directa, debido a que TSCH implementa un mecanismo de salto de canal. Este mecanismo genera una rotación de los canales físicos en cada slotframe para así reducir los puntos de interferencia y el desvanecimiento de trayectorias (*path fading*), aumentando la fiabilidad.

En consecuencia, los recursos en el slotframe se asignan por un par de valores que corresponden al timeslot y al channel offset, este par de valores conforman un slot o celda dentro de la planificación TSCH. Las celdas pueden ser de dos tipos, compartidas (*Shared*) o dedicadas. En el caso de ser compartidas se sigue el esquema CSMA-CA donde se escucha el medio antes de las transmisiones para reducir las colisiones. En cambio, si es dedicada existen unas fuertes restricciones temporales, ya que se establece un enlace unidireccional entre dos nodos y solo el nodo fuente puede transmitir durante esa celda. La planificación permite establecer que acción va a realizar el nodo en cada celda. El nodo podrá recibir, transmitir o apagar la radio. Para que esto funcione de forma correcta los nodos dentro de la planificación deben estar correctamente sincronizados para que así las celdas de transmisión y recepción coincidan. Esto se consigue con el intercambio de paquetes de balizamiento (EB - *Enhanced Beacon*) que incluye información sobre el número de secuencia absoluto (ASN - *Absolute Sequence Number*) que los nodos toman como referencia temporal para determinar el timeslot actual. La información sobre el ASN también permite compensar las desviaciones de reloj, que se producen debido a la naturaleza de los cristales en los osciladores y así conservar el sincronismo. Además, TSCH incluye unos tiempos de guarda que permiten encender la radio del nodo receptor unos microsegundos antes del inicio del slotframe. Así se reduce aún más la pérdida de información por las desviaciones de reloj. Sin embargo, tener encendida la conexión radio más tiempo afecta al consumo de energía.

El salto de canales permite que las transmisiones no sean afectadas de forma constante por la interferencia, esto se consigue al realizar las transmisiones de una misma celda en canales físicos diferentes. La frecuencia que se debe usar en cada caso se obtiene de acuerdo a la ecuación (2-1). Esta se utiliza para determinar la frecuencia o canal físico que deben usar los nodos en los estados de transmisión y recepción. La frecuencia a utilizar se encuentra en

función del ASN , el channel offset (Ch_{offset}) y la cantidad de canales (N_{ch}) que se utilizan en la planificación.

$$frequency = F[(ASN + Ch_{offset}) \bmod N_{ch}] \quad (2-1)$$

De acuerdo con [29] [30], TSCH consigue mayores tasas de transferencia y fiabilidad en los entornos industriales gracias a su planificación temporal de las ventanas de transmisión, y a su mecanismo de salto en frecuencia. Debido a que estas características permiten mitigar las interferencias o desvanecimientos de la señal característicos en los escenarios industriales. Esto hace que TSCH sea considerado el mecanismo de acceso al medio *de facto* a la hora de desplegar una IWSN que deba cumplir con unos altos requisitos de calidad de servicio en entornos industriales. Sin embargo, estas características también se incluyen dentro de DSME por lo que la diferencia entre los dos protocolos es mínima. El inconveniente en DSME es que presenta una estructura menos flexible, compuesta por supertramas con periodos de contienda y periodos planificados con longitudes fijas, mientras que en TSCH se puede aumentar de forma libre el slotframe, también es posible utilizar múltiples slotframe de diferentes tamaños simultáneamente. Además, en TSCH las decisiones de planificación están tomadas por capas superiores que no están estandarizadas y que no se incluyen en ninguna norma, a diferencia de DSME, donde la negociación de recursos se realiza de forma distribuida, intercambiando paquetes entre fuente y destino. Para implementar un control completamente centralizado, la falta de estandarización en la planificación de recursos en TSCH presenta una oportunidad más interesante para abordar con SDN al no tener que deshabilitar protocolos distribuidos y evitar tener en ejecución procesos que no se utilizaran en el control con SDN. En la tabla 2-3 se resumen las principales características de estos protocolos.

2.2.4. Planificadores TSCH

Para asignar las acciones que debe ejecutar cada uno de los nodos en el slotframe TSCH, debe haber un proceso de planificación que permita sincronizar los estados de transmisión y recepción. Sin embargo, el estándar IEEE 802.15.4e no incluye ninguna especificación sobre cómo hacerlo. Por este motivo, existen diferentes tipos de planificadores TSCH, que se pueden clasificar en: centralizados, distribuidos o autónomos, dependiendo de la entidad que tome la decisión para asignar las celdas. Entre los planificadores centralizados se encuentra TASA (*Traffic Aware Scheduling Algorithm*) [31] y AMUS (*Adaptive Multi-hop Scheduling*) [12], donde un elemento central se encarga de asignar la planificación a cada nodo. En los planificadores distribuidos como DeTAS (*Decentralized Traffic Aware Scheduling*) [32] y Wave [33] en este caso, los nodos realizan una negociación para asignar las celdas. Finalmente, los planificadores autónomos reutilizan información de otros protocolos y cada nodo asigna su planificación de forma local, entre estos está Orchestra [34], así como algunas variantes como Escalator [35] y ALICE (*Autonomous Link-based Cell scheduling*) [36].

	CSMA-CA	LLDN	DSME	TSCH
Sincronización	✗	✓	✓	✓
Determinismo	✗	✓	✓	✓
Escalabilidad	Media	Media	Muy alta	Alta
Eficiencia Energética	Baja	Alta	Media	Alta
Complejidad de Implementación	Baja	Media	Alta	Alta
Topología	Estrella	Estrella	Estrella, Malla, Árbol	Estrella, Malla, Árbol
QoS	Baja	Media	Alta	Alta
Flexibilidad	Media	Baja	Media	Alta
MAC	CSMA-CA	TDMA	TDMA+FDMA	TDMA+FDMA
Multicanal	✗	✗	✓(Solo en CFP)	✓
Tipo de Planificador	✗	Centralizado	Distribuido	Centralizado/Distribuido
Planificador	✗	✗	Integrado en el LLC (Logical Link Control)	No definido, administrado por las capas superiores
Estructura de Planificación	Contención (CSMA-CA)	Superframe simplificado	Multisuperframe, longitud variable en grupos de 16 timeslots	Slotframe, longitud variable con una resolución de 1 timeslot

Tabla 2-3: Comparativa de Protocolos de Control de Acceso al Medio en 802.15.4e

El uso del protocolo TSCH permite conseguir una tasa de fiabilidad muy alta, cerca del 99,99 % cuando se usa un protocolo de planificación autónomo como Orchestra [34]. Sin embargo, en este caso la capacidad de reconfigurar la red es limitada. Además, la flexibilidad y el determinismo de la red se ven afectados por las decisiones autónomas de cada nodo. En cambio, el uso de un enfoque centralizado con un conocimiento total de la red permite optimizar el enrutamiento, la planificación TSCH y reducir la carga de procesamiento en los nodos. Aun así, el inconveniente principal de los enfoques centralizados es que implican agregar un flujo de tráfico adicional para la información de control. Esto reduce el rendimiento de la red, como se ha demostrado en otros trabajos como TASA [31] y AMUS [12]. La principal limitación de estos, es que no representan un cambio significativo en la operación de los nodos, ya que conservan la pila de protocolos y el intercambio de mensajes tradicional y se ubican solo como protocolos en la capa de aplicación. Por eso se hace necesario un enfoque más disruptivo donde el equipo centralizado, no solo ejecute una tarea específica, como el enrutamiento o la planificación de recursos de forma aislada, sino, que administre de forma conjunta estos y todos los procesos necesarios para controlar de forma precisa el flujo de información. Este equipo centralizado sería el controlador, definido dentro de las Redes definidas por software (SDN), un paradigma diseñado inicialmente para centros de datos, que ha sido ampliamente investigado y estandarizado para redes cableadas, debido a que una visión global de la red permite tomar decisiones con un alto nivel de optimización.

Los procesos actuales de convergencia e hiperconvergencia requieren un mayor control sobre las infraestructuras, donde los flujos de información deben ser completamente deterministas y garantizados, además, de seguir teniendo la flexibilidad y capacidad de reconfiguración

necesaria para la Industria 4.0. Es aquí donde ahora deben evolucionar las IWSN en implementar y converger hacia los nuevos paradigmas de red que les permitan alcanzar estos objetivos, uno de estos son las redes definidas por software.

2.3. Redes definidas por software

El paradigma SDN [6] surgió como respuesta a la compleja administración que se tiene en los centros de datos, donde es necesario gestionar y optimizar una gran cantidad de recursos y equipos de infraestructura. Este consiste en administrar las redes separando el plano de control y el plano de datos. El plano de control estaría totalmente controlado por software, donde un agente centralizado se encarga de tomar las decisiones y utilizarlas para configurar las reglas de flujo en los dispositivos de reenvío. Estas decisiones son tomadas a partir de un conocimiento topológico total de la red y de las políticas establecidas para cada uno de los flujos. El plano de datos o de reenvío está compuesto por los equipos físicos, que operan en modo *plug-and-play*, debido a que el controlador enviará todas las instrucciones necesarias para conseguir el máximo rendimiento. Este enfoque permite que la red pase a ser programable, ya que el controlador puede modificar las tablas de flujo de cada dispositivo.

Esta flexibilidad, obtenida al separar el plano de datos y el plano de control permite superar las limitaciones de las redes tradicionales [37], debido a que se tiene una visión global de la red, optimizando su uso y simplificando el aprovisionamiento de recursos. Con SDN es posible gestionar, configurar y optimizar dinámicamente los recursos de la red mediante programas o funciones automatizadas que se integran en el controlador. Por esta razón, se han realizado varios estudios sobre las ventajas de SDN, especialmente en QoS, donde es posible explotar el dinamismo y la gestión de la red con un mayor nivel de detalle. Un ejemplo de esto es [38] donde el controlador obtiene la mejor ruta para cada tipo de tráfico según los requisitos de QoS utilizando parámetros como: Latencia, Jitter, Throughput y Pérdidas.

Con el fin de explotar estas habilidades de orquestación, además del plano de control y del plano de datos que se separan de los equipos físicos, SDN contempla un plano adicional, el de administración. Este plano agrupa todas las funciones automatizadas que alteran el comportamiento de la red, tales como el enrutamiento, los cortafuegos, los balanceadores de carga, la monitorización, etc. El plano de administración se encuentra por encima del plano de control, como se muestra en la figura **2-7(a)**, esto permite reutilizar el conocimiento global del controlador para tareas específicas. El objetivo de esto es simplificar el desarrollo de aplicaciones, ya que no es necesario entrar en detalles sobre el funcionamiento de la infraestructura de red, será el controlador el que se comunique con cada uno de los elementos de la red. Así, el controlador SDN maneja una gran diversidad de protocolos, que se dividen en *Northbound* (NB) y *Southbound* (SB). Como se muestra en la figura **2-7(b)** los protocolos NB son una interfaz que permite al plano de aplicación comunicarse con el controlador. Esta interfaz abstrae los conjuntos de instrucciones de bajo nivel para programar los dispositivos

de reenvío, facilitando la programación de la red. Por otro lado, los protocolos SB se utilizan para comunicarse con la infraestructura de red, siendo el más extendido el OpenFlow desarrollado por la *Open Networking Foundation* (ONF) [39] debido a que es un estándar abierto que incluye todas las especificaciones necesarias para implementar equipos de reenvío y el canal de comunicaciones entre el plano de control y el de datos.

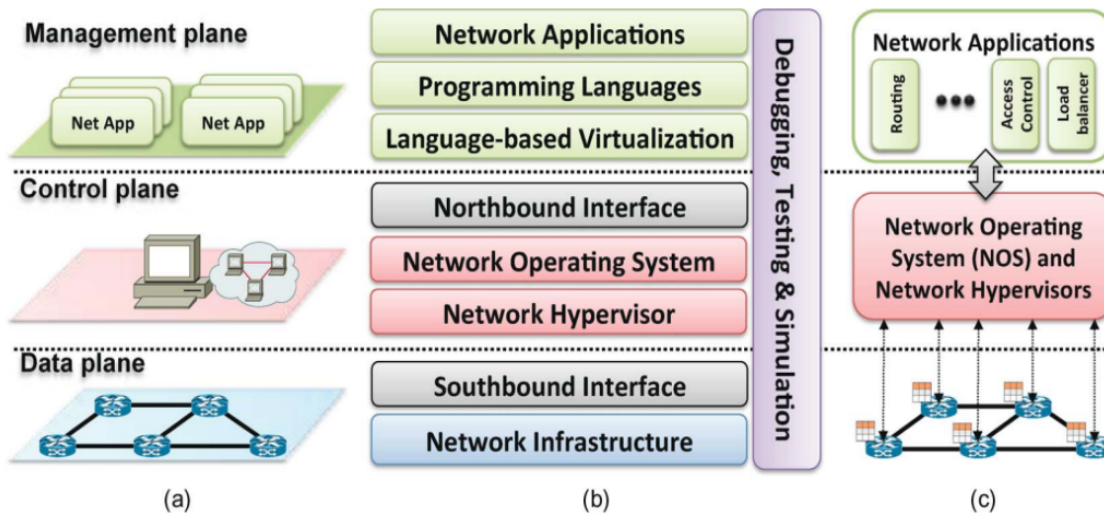


Figura 2-7: Redes definidas por software en: (a) Planos, (b) Capas y (c) Arquitectura [6].

El protocolo OpenFlow proporciona tres fuentes de información, en primer lugar, los dispositivos de reenvío envían mensajes basados en eventos al controlador cuando se produce un cambio en alguno de sus enlaces. En segundo lugar, los dispositivos de reenvío generan estadísticas para cada flujo y son recogidas por el controlador y por último se generan mensajes cuando hay un paquete que no puede ser enviado. La figura 2-7(c) muestra la arquitectura, en el plano de datos la inteligencia de cada uno de los dispositivos ha sido reemplazada por una tabla de flujo que define cómo deben tratarse los paquetes. Estas contienen las reglas de flujo, que comparan los campos en el paquete y el valor que debe coincidir. Cada vez que llega un nuevo paquete, se realiza un proceso de búsqueda empezando por la primera tabla y termina cuando se encuentra una coincidencia. En caso de que no se encuentre una coincidencia, existe una regla por defecto que le ordena al dispositivo enviar el paquete al controlador. En este caso, el controlador analizará el paquete recibido y generará la regla de flujo correspondiente para este tipo de paquete. Además, se observa como cada uno de los equipos establece una comunicación bidireccional con el controlador mediante un canal de control dedicado, de esta forma se garantiza que la información contenida en las tablas de flujo pueda ser actualizada con una mayor frecuencia.

La flexibilidad, simplicidad en el despliegue y la administración que ofrece SDN, ha impulsado su expansión a todos los puntos de la red cableada. De acuerdo con CISCO en el *Global Networking Trends* de 2020, el 64% de las organizaciones tenían soluciones SDN en sus

centros de datos, 58 % como tecnología de WAN (SD-WAN) y 40 % en las redes de acceso. Actualmente, todos los fabricantes de equipos de infraestructura, como Cisco, HP, IBM, Juniper, Huawei, ofrecen controladores SDN y equipos en todos los niveles de la red: LAN, WAN y Datacenter. Debido a que todos utilizan como base el protocolo OpenFlow son totalmente compatibles, incluso existen soluciones de código abierto como Open vSwitch e Indigo que pueden ser instaladas en equipos de propósito general.

Uno de los efectos de SDN, es la reducción de la complejidad en los equipos de reenvío, ya que solo deben ejecutar comparaciones con las reglas que se reciban del controlador, todo el procesamiento y la toma de decisiones está centralizado. Las ventajas que supone SDN sobre el enfoque tradicional son esencialmente la versatilidad, flexibilidad y la simplicidad para administrar los recursos. Esto es debido a que los equipos físicos dejan de ser dependientes de los algoritmos o funciones locales para ser adaptables [7], es decir, el funcionamiento de estos equipos puede ser modificado en cualquier momento a través de las órdenes del controlador. Las aplicaciones en SDN están orientadas al controlador, esto quiere decir, que se abstrae la diversidad y complejidad de los elementos físicos al comunicarse directamente con el controlador y simplemente será este, el encargado de gestionar los flujos de información y el comportamiento de los equipos del plano de datos. La mayor eficiencia de SDN se consigue a largo plazo donde sus capacidades de evolución y adaptabilidad son necesarias. Por ejemplo, en las WSN hay una gran diversidad de protocolos (Zigbee, 6LOWPAN, etc.) cuya capa física está basada en el estándar IEEE 802.15.4, sin embargo, estos protocolos son incompatibles debido a que las capas superiores están encapsuladas de forma diferente. Con SDN esta incompatibilidad puede ser solucionada, ya que las capas superiores están definidas por software y el controlador podría establecer reglas de flujo que permitan desencapsular diferentes protocolos [40].

2.4. SDN en WSN

En las IWSN el funcionamiento de SDN es similar, el equipo centralizado o controlador se encarga de definir las reglas de reenvío basándose en un conocimiento topológico total de la red (Plano de control) y a las políticas que tengan establecidas cada uno de los flujos. Los nodos reciben estas reglas y las almacenan como tablas de flujos, después de esto simplemente deben realizar una búsqueda dentro de esta tabla y ejecutan las acciones enviadas por el controlador (Plano de datos). Con este enfoque se consigue que la red sea programable, ya que el controlador puede modificar las tablas de flujo de cada sensor [7], además, de dividir el tráfico en flujos, mucho más complejos, que una segmentación por IP o MAC, debido a que se analizan múltiples variables a nivel local y global. Así se consigue mejorar de forma considerable el rendimiento de la red, aumentando la calidad de servicio y previniendo posibles puntos de congestión.

Las capacidades ofrecidas por las SDN, sobre todo la flexibilidad que se consigue al poder

generar cualquier tipo de aplicación orientada al controlador, abre nuevas posibilidades para abordar la amplia gama de retos que tienen las WSN. Se pueden añadir mejoras con la implementación de SDN como eficiencia energética, enrutamiento, movilidad, seguridad, fiabilidad y calidad de servicio [41]. Esta última cada día gana más importancia, debido a que, en las redes de nueva generación, circulará tráfico de fuentes muy variadas, con diferentes necesidades según su aplicación (tiempo real, entrega asegurada, ancho de banda. . .).

Por lo tanto, a cada tipo de tráfico se le debe garantizar parámetros diferentes de calidad de servicio [42]. Generalmente, la QoS se basa en la clasificación y el marcado de tráfico para asignar los recursos de red de acuerdo con las diferentes clasificaciones realizadas. Un diseño óptimo de la red no debe dejar a un lado los requisitos de QoS, pues esto podría provocar comunicaciones deficientes. Sin embargo, implementar técnicas de QoS en las WSN no es algo trivial, ya que este tipo de redes fueron diseñadas para aplicaciones con baja tasa de tráfico y no para aplicaciones en tiempo real, [43]. Además, en las redes industriales es vital poder ser previsible, para esto se deben garantizar y sincronizar los tiempos de envío y recepción de la información, haciendo que la red sea altamente determinista. Para cumplir con esto se deben tener en cuenta múltiples factores de la topología (congestión, calidad del enlace), la segmentación en flujos por tipos de tráfico y la integración con otro tipo de redes. Por lo tanto, el conocimiento global que aporta el enfoque SDN es más adecuado para conseguir este objetivo, debido a que tiene la capacidad de conocer, monitorizar y configurar cada una de estas etapas. Por ello, las WSN basadas en SDN pueden ofrecer una QoS optimizada, ya que las aplicaciones programables orientadas al controlador pueden ser desarrolladas para implementar y manejar los requisitos de QoS relacionados con los nodos de sensores [44].

Sin embargo, la aplicación de SDN en las WSN requiere un enfoque específico para tener en cuenta las limitaciones de los dispositivos de estas redes. Estos generalmente presentan limitaciones de ancho de banda, interfaces de comunicación y número de enlaces simultáneos, a diferencia de los equipos de centros de datos para los que se diseñó SDN.

Por esta razón, se han realizado diversas investigaciones sobre la implementación de SDN en redes de sensores (WSN), partiendo de las ideas de Tie Luo en Sensor OpenFlow [7], donde se intenta trasladar el protocolo OpenFlow, estándar para redes cableadas. Pero debido a las grandes diferencias del entorno WSN, la cantidad de modificaciones que se debían hacer al protocolo lo hacían inviable. A partir de aquí han surgido diferentes frameworks de SDN aplicados en las redes de sensores como: TinySDN [45] basado en el sistema operativo TinyOS y su aplicación a SDN, está limitado debido a dos únicas acciones (*Drop y Forward*) que realiza la tabla de flujos. SD-WSN [40] contiene todas las características presentes en sensor OpenFlow y lo extiende a través de funciones como “*in-network data aggregation*” que permite unir flujos que comparten características de enrutamiento (INNP). uSDN [46] donde se usa el protocolo 6TiSCH y sus nuevos elementos para crear la planificación TSCH de forma híbrida (Centralizada y Distribuida), con esta planificación realiza un slicing de la red para enviar el tráfico de señalización, sin embargo, el uso de capas superiores estandarizadas

reduce la flexibilidad de la red definida por software. Atomic SDN [47] transmite el tráfico de señalización de una forma más eficiente, rápida y aislada de los otros procesos, utilizando mecanismos de inundación síncrona, pero debido a los diferentes tipos de flujos se vuelve complejo, ya que necesita un algoritmo para cada tipo. Muchas investigaciones en SD-WSN están enfocadas a la optimización del enrutamiento y el control del flujo de datos, pero se pueden mejorar otros aspectos como el control de la topología, tal y como se ha hecho en [48]. En [49] se propone SDN-WISE, una implementación SDN para WSN que mejora en la definición de reglas de flujos, donde es posible analizar cualquier byte del paquete a diferencia de los frameworks anteriores que, se limitaban a analizar campos específicos como en las redes cableadas. SDN-WISE se centra en la flexibilidad y seguridad, es *stateful*, es decir, mantiene la información de estado dentro de cada sensor, el cual puede ser modificado mediante la ejecución de acciones o procesos. El controlador que utiliza es ONOS (*Open Network Operating System*) y lo amplía para soportar virtualización. Una evolución de este protocolo es SD-WISE [50], que efectúa múltiples acciones en la tabla de flujos (Drop, Forward, Sleep, Modify, INNP), ha incorporado elementos de TSCH minimal, pero aún no tiene en cuenta la planificación desde el controlador. Este framework se ha estudiado a través de diferentes artículos, como en [51], donde se agrega un indicador del estado del buffer para informar al controlador los niveles de congestión de cada nodo y balancear el tráfico. También en [52] donde se analiza la respuesta del enfoque SDN con nodos en movimiento, que requieren una actualización rápida de la topología.

2.5. Movilidad en IWSN

La inclusión de nodos móviles se ha investigado ampliamente en las redes móviles ad-hoc (MANETS), y las redes ad-hoc vehiculares (VANETS) [53]. Sin embargo, las IWSN deben garantizar unos estrictos requisitos de calidad de servicio para obtener el mayor determinismo posible. Un objetivo difícil de alcanzar, teniendo en cuenta los constantes cambios en la topología producidos por el uso de nodos móviles, que afectan directamente al rendimiento de la red. El reto de los nodos móviles en IWSN ha sido abordado en múltiples trabajos, donde se ha evaluado el efecto de los nodos móviles en RPL con Orchestra [54] y TSCH minimal [55]. Los resultados obtenidos muestran cómo el tráfico de control de RPL aumenta, debido a las desconexiones de los nodos móviles. Cada vez que un nodo móvil se desconecta, necesita iniciar un proceso de descubrimiento para obtener el nuevo nodo padre, lo que requiere un tiempo de convergencia. Además, en este caso los nodos dependen de los temporizadores para saber que están desconectados e iniciar un nuevo proceso de descubrimiento. Esto también aumenta la pérdida de paquetes y la latencia.

En [56] se intentó superar esta dificultad con el uso de temporizadores que permiten actualizar rápidamente los nodos padres que abandonan el área de cobertura. También se han probado diferentes técnicas para optimizar el proceso de enrutamiento, como la segmentación de la

red en áreas circulares definidas por las raíces del grafo acíclico dirigido (DAG) [57] o el uso de filtros de Kalman para mejorar las decisiones de RPL con predicciones de posición [58]. Además, se han mejorado las funciones de programación distribuida, incluyendo la predicción de la posición, el tamaño de la cola y la tasa de generación de paquetes para negociar el número de celdas necesarias [59]. Este proceso de negociación tiene limitaciones en entornos de movilidad, ya que cuando se produce un cambio de padre el nodo necesita tiempo para descubrir a los nuevos vecinos e iniciar el proceso de negociación. Esto puede reducirse reservando celdas dedicadas a la negociación y añadiendo múltiples paquetes de control para detectar los cambios de topología [60]. Sin embargo, aunque representan una mejora respecto a la RPL tradicional, están lejos de ofrecer el mismo determinismo que los nodos fijos.

Estos estudios previos muestran cómo el soporte de nodos móviles aumenta considerablemente la complejidad de las decisiones en los nodos y la cantidad de paquetes de señalización. Por ello, la implementación de Redes Definidas por Software (SDN) en las IWSNs se convierte en una mejor opción, ya que supone un aumento del tráfico de control a cambio de ventajas adicionales, como son: la reducción de la complejidad de los procesos en los nodos y una visión global de los elementos de la red. La combinación de SDN y TSCH aumenta la fiabilidad, el determinismo y la flexibilidad de la red. Esto se demuestra en [61], donde es posible garantizar diferentes parámetros de QoS para diferentes tipos de flujos de forma dinámica y con baja complejidad en comparación con la RPL con Orchestra.

Para dar soporte a los nodos que se mueven utilizando SDN sobre TSCH en [62] proponen FTS-SDN (*Forwarding and TSCH Scheduling over SDN*). El objetivo es aumentar la frecuencia y fiabilidad en el intercambio de paquetes de control. Para disminuir el tiempo de actualización de la topología, el controlador enviará las nuevas reglas de enrutamiento con cada cambio de nodo padre. Para adaptarse a los cambios, el controlador debe actualizar la topología cada vez que el nodo móvil cambie de nodo padre. En este caso, los nodos deben informar de su existencia a través de mensajes de baliza que se difunden para ser recibidos por todos los nodos en el rango de cobertura. Además, deben enviar la información de las balizas recibidas al controlador. Esto se hace mediante el envío de paquetes de reporte, que incluyen todos los nodos de los que se han recibido mensajes de baliza. Por tanto, cuanto mayor sea el número de paquetes de control, más actualizada estará la topología y más rápido se adaptará a los cambios del nodo móvil. De hecho, la velocidad a la que se mueve el nodo móvil es un parámetro que puede afectar a este enfoque. Cada vez que el controlador actualiza la topología, debe generar las rutas e informar a los nodos de la nueva ruta con un paquete OpenPath.

El problema general es que la mayor parte de los estudios están enfocados en apartados específicos, omitiendo algún componente importante para una red TSCH SDN. Un ejemplo de esto, son los estudios [63], [64] que utilizan los elementos de 6TiSCH para lograr optimizar la planificación TSCH de forma centralizada e híbrida. Además, incorporan el uso de *tracks*

para servicios de alta prioridad. Sin embargo, las pruebas se realizan con rutas y celdas instaladas de forma manual en cada nodo, lo que omite el cálculo, la generación y el envío de la planificación a los nodos, además, del impacto que genera el tráfico de señalización. Otros trabajos como [44] y [65], tratan la necesidad de asociar SDN y TSCH en las aplicaciones industriales, pero en este caso se limitan a estudios completamente teóricos. Teniendo en cuenta la necesidad de tratar los requisitos de las SD-WSN de modo distinto a las SDN clásicas, en [52] los autores proponen la combinación de TSCH con el protocolo SDN-WISE para poder cumplir con los requisitos de movilidad de las IWSNs. Dado que los resultados obtenidos son preliminares y no abordan la viabilidad en una implementación real, los mismos autores presentan en [62] el protocolo FTS-SDN que proporciona buenos resultados tanto en experimentos como en simulación de escenarios reales. No obstante, tampoco en este trabajo se aborda en detalle el planificador TSCH y el tráfico de control que aumenta en una implementación SDN sobre WSN, ya que el estudio se centra más en la movilidad de los nodos. Por lo tanto, una implementación Software defined WSN completa, debe tener una centralización total de la red, donde los procesos de enrutamiento, planificación TSCH y tráfico previsto en la red, estén relacionados.

Capítulo 3

Integración de SDN en IWSN

3.1. Introducción

El uso de SDN amplía las posibilidades de control sobre la red y permite mejorar su rendimiento a nivel industrial, ya que ofrece una gestión centralizada con una visión global de los recursos disponibles, donde se pueden crear arquitecturas flexibles y dinámicas que soporten simultáneamente fuentes de tráfico con diferentes características. Sin embargo, el envío de tráfico de control desde cada uno de los nodos hacia el controlador representa una carga de tráfico adicional que puede reducir el rendimiento y aumentar el consumo de energía. Esta dificultad puede ser superada con la segmentación del tráfico en flujos, y una correcta planificación de recursos a nivel MAC, lo que permite realizar un *slice*, que consiste en una segmentación lógica y física de la red donde no es posible tener interferencia de otros flujos.

En este capítulo se aborda la integración de las IWSN con las SDN, como una solución para garantizar la calidad de servicio en entornos con múltiples flujos de datos de forma independiente, lo que permite reducir el impacto del tráfico de control en el rendimiento. Para esto, el controlador SDN realiza un proceso de asignación de recursos que integra los procesos de enrutamiento, la planificación TSCH y el manejo del tráfico por flujos deterministas a través de los descriptores de tráfico. Estos elementos se encargan de forma conjunta de hacer una planificación y segmentación del tráfico que será enviada a los nodos a través de un único paquete desde el controlador.

Para la implementación de estos elementos se toma como punto de partida el protocolo SDN-WISE (Software Defined Networking solution for WIreless SEnsor Networks), el cual ofrece capacidades SDN en un entorno CSMA donde no es posible garantizar parámetros de calidad de servicio. La arquitectura y la pila de protocolos en SDN-WISE están orientadas específicamente a adaptar el paradigma SDN a las limitaciones de las IWSN. Por lo tanto, está compuesto por una pila de protocolos propietarios o no estandarizados a excepción de

la capa física y el control de acceso al medio. En este caso, el uso de protocolos propietarios reduce las restricciones, ya que están disponibles de forma abierta y no es necesario cumplir ningún estándar, lo que proporciona una mayor libertad para modificar y optimizar el comportamiento de toda la arquitectura a través de las diferentes capas.

Finalmente, el rendimiento es evaluado mediante simulación y con un testbed, donde se compara la arquitectura obtenida con el SDN-WISE original. Los parámetros a evaluar son la cantidad de tráfico de control, el retardo extremo a extremo, pérdida de paquetes, tasa de satisfacción de deadline (DSR - *deadline satisfaction ratio*) y el tiempo de vida de la red para el consumo de energía.

3.2. Arquitectura

3.2.1. SDN-WISE

SDN-WISE es un protocolo que extiende el paradigma SDN hacia las IWSN. Está basado en el protocolo OpenFlow y considera los retos planteados en [7] para su implementación en entornos inalámbricos. Para solucionar estos retos se requería un cambio profundo del protocolo OpenFlow, como poder tomar decisiones de acuerdo con los estados locales sin intervención del controlador (*Stateful*), o poder realizar una agregación de paquetes para reducir la congestión de la red. Debido a la complejidad de los cambios no es posible adaptar de forma directa el protocolo OpenFlow, por ello se crean protocolos nuevos siguiendo los conceptos fundamentales de OpenFlow y las restricciones de los entornos inalámbricos como es el caso del protocolo SDN-WISE. Los cambios están orientados a reducir el tráfico de control y conservar el ancho de banda de la red, ya que en las IWSN no hay un aislamiento físico entre los flujos de control y de datos, por ello, el tráfico de control impacta de forma directa en el rendimiento de la red. Sin embargo, las ventajas que tiene una implementación SDN, como versatilidad, flexibilidad y la simplicidad de la administración [50], son elementos clave dentro de las IWSN, debido a que los flujos de tráfico en la red pueden ser programados y modificados por el controlador, simplemente modificando la tabla de flujo en cada nodo.

El protocolo SDN-WISE está compuesto por 5 capas: Aplicación, Descubrimiento de la topología (TD - *Topology Discovery*), Reenvío (FWD - *Forwarding*), el control de acceso al medio (MAC) y la capa física que en este caso es 802.15.4. Además, una subcapa de Procesamiento de paquetes en red (INNP - *In-Network Packet Processing*) que utiliza componentes de TD y FWD. Las dos capas centrales de acuerdo a la figura **3-1(a)**, es donde se concentra el funcionamiento de SDN-WISE, son capas completamente propietarias que no requieren el uso de otros protocolos, la función principal de cada capa es la siguiente:

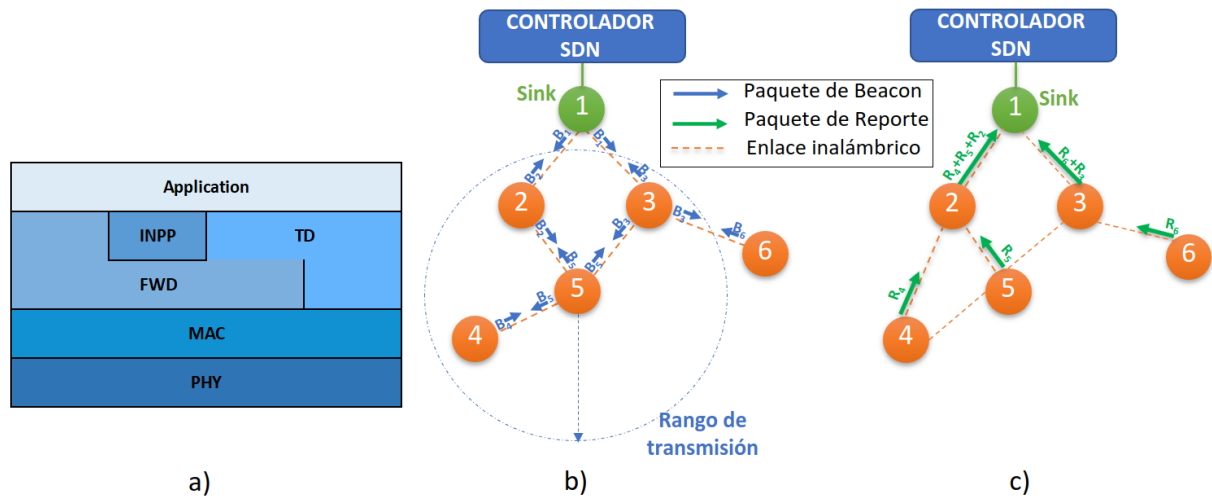


Figura 3-1: (a) Pila de protocolos SDN-WISE, (b) Paquetes de beacon, (c) Paquetes de reporte

Descubrimiento de la topología (TD): esta capa es la encargada de descubrir los nodos adyacentes e informar al controlador de sus vecinos. Para descubrir los nodos adyacentes todos los nodos generan paquetes de beacon, que son enviados de forma periódica en modo broadcast para informar de su presencia a todos los nodos en el rango de cobertura. Los nodos que reciben este paquete añaden este nodo a su tabla de vecinos. Esta información de todas las adyacencias de cada nodo es enviada al controlador en un paquete de reporte. El controlador recibe los paquetes de reporte de cada uno de los nodos y construye una topología basada en la información recibida, obteniendo una visión global de la red.

Reenvío (FWD): se encarga de procesar los paquetes salientes de acuerdo con la tabla de flujo WISE. Esta tabla está compuesta por las reglas de flujo enviadas desde el controlador que tienen 3 secciones: reglas de coincidencia (*matching rules*), acción y estadísticas. Las reglas de coincidencia contienen los campos que se deben comparar en los paquetes y el valor esperado, el paquete puede ser inspeccionado en cualquiera de sus bytes a diferencia de OpenFlow que solo compara valores del encabezado. Es posible añadir hasta 3 reglas de coincidencia por cada regla de flujo. Cuando el paquete pasa a través de las reglas de forma exitosa se ejecuta la acción asociada que puede ser de 5 tipos: *Drop*, *Forward*, *Sleep*, *Modify* o *INNP* y se contabiliza en las estadísticas. La acción de reenvío (*forward*) se ejecuta de acuerdo con el identificador del nodo (*node ID*) que tiene una longitud de 2 bytes. El uso de este direccionamiento reducido en lugar de la dirección MAC es común en equipos que tienen limitaciones de memoria y tamaño de los paquetes. De esta forma se optimiza la cantidad de memoria y la información útil que se transmite. Para el caso de los nodos puede ser configurado de forma manual o con los 2 bytes menos significativos de la dirección MAC.

Procesamiento de paquetes en red (INNP): es una subcapa que se utiliza para la agregación de datos. Pequeños paquetes pueden ser combinados cuando tienen el mismo destino para minimizar el número de transmisiones y evitar la congestión de la red. Se debe tener en cuenta que el tamaño máximo de un paquete SDN-WISE es de 116 Bytes y este límite no puede ser superado.

SDN-WISE, envía diferentes tipos de paquetes que permiten al nodo tener una comunicación con el controlador. Todos los paquetes en SDN-WISE tienen un encabezado de 10 Bytes que consta de los siguientes campos: Longitud del paquete (1 Byte), ID de la red (1 Byte), nodo fuente (2 Byte), Nodo destino (2 Byte), tipo de paquete (1 Byte), TTL (Time To Live) (1 Byte) y próximo salto (2 Byte). A partir de aquí la información que se incluye depende del tipo de paquete, como se describe a continuación.

Data (Tipo 0): Este tipo de paquetes es el que se encarga de enviar el tráfico común, como las medidas de los sensores. Los bytes posteriores al encabezado conforman el payload, sobre este tipo de paquetes se aplicarán las prioridades.

Beacon (Tipo 1): Es un mensaje Broadcast (255.255) que se envía para informar a los nodos que están dentro del rango de cobertura. Incluye información del estado de la batería (0xFF=Llena, 0x00=Vacía) y la distancia al sink de acuerdo al número de saltos. Se pueden agregar más parámetros para informar por ejemplo del nivel de congestión medido por la ocupación del buffer. Un ejemplo del envío de este paquete se muestra en figura 3-1 (b). El nodo 5 envía los paquetes de beacon (broadcast) a todos aquellos nodos que están dentro de su rango de transmisión. En el ejemplo los nodos 2, 3 y 4 recibirían este paquete.

Report (Tipo 2): Este tipo de paquete es el que usa el controlador para crear la topología de la red. El paquete incluye la cantidad de vecinos e información de cada uno como el identificador de nodo y el RSSI. En la figura 3-1(c) el nodo 4 envía un paquete de reporte hacia el controlador para indicar cuáles son sus vecinos y lo hace seleccionando el camino más corto, el cual conoce debido a que los beacon contienen la distancia al sink.

Request (Tipo 3): Cuando un paquete no tiene una coincidencia en la tabla de flujos, este se envía al controlador para preguntar qué debe hacer con este tipo de tráfico. El controlador responde con un mensaje tipo 4, con las reglas de flujo.

Response (Tipo 4): El controlador envía mensajes con reglas de flujo cuando un nodo hace una petición sobre el tráfico que no está especificado en su tabla de flujo.

OpenPath (Tipo 5): Este paquete se encarga de instalar una regla de flujo específica en múltiples nodos con una sola transmisión desde el controlador. Los nodos en los que se instalan forman parte de una ruta. El paquete es procesado y enviado al nodo siguiente en el camino o path. Este proceso reduce el tráfico de control.

Config (Tipo 6): Se utiliza para establecer las opciones de configuración en el nodo, como el periodo de beacon, reporte o incluir reglas de flujo específicas para un nodo.

RegProxy (Tipo 7): Cada nodo sink envía este paquete al controlador para anunciar su existencia por medio de la interfaz habilitada para la conexión con el controlador, donde envía el identificador del sink y su MAC. Con esta información el controlador decide si acepta la comunicación con este sink, en caso de establecer la comunicación, los paquetes de control de toda la IWSN pasan por este enlace.

	10 Bytes									1:16 Bytes					2*NN Bytes							
Byte	0	1	2	3	4	5	6	7	8	9	10	11 : (10 + NoR*5)										
Description	Length (Bytes)	Network ID	Source Node		Destination Node		Packet type	TTL	Next Hop		(NoR) Number of Rules	Rule 1	:		Rule NoR	First node in the path		:		Last node		
	Header									Flow Rules					Nodes in the Path							
(a)																						
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Description	Length (Bytes)	Network ID	Source Node		Destination Node		Packet type	TTL	Next Hop		(NoR) Number of Rules	Rule n.1 (Each routing rule occupies 5 Bytes)				First node in the path		Second node		Last node		
Value	26	1	1	1	3	3	5	100	0	1	1	114	0	0	0	40	1	1	3	3	6	6
	Header									Flow Rules					Nodes in the Path							
(b)																						

Figura 3-2: Estructura del paquete OpenPath original. (a) ejemplo genérico, (b) ejemplo con path de 3 nodos y 1 Regla de flujo

En la figura 3-2(a) se muestra un paquete OpenPath tal y como se define en el protocolo SDN-WISE, [50]. Los bytes después del encabezado son utilizados para definir las reglas de flujo y los nodos en la ruta como se explica a continuación. El Byte 10 indica la cantidad de reglas de coincidencia (*NoR*) que debe instalar el nodo, hasta un máximo de 3. Cada regla tiene una longitud establecida de 5 bytes y está compuesta por una condición y una acción. En la condición se pueden comparar valores específicos del paquete en cualquiera de sus campos, incluido el payload, cuando esta condición se cumple se ejecuta la acción.

El siguiente grupo de bytes son los relacionados con los nodos en el path, de forma genérica este grupo empezará en el Byte $(11 + 5 \cdot NoR)$, ya que depende del número de reglas. En estos bytes se incluye en orden de reenvío el identificador de cada uno de los nodos que compone la ruta. Cuando un nodo recibe un paquete OpenPath se busca entre el grupo de nodos, y se instalan las reglas de flujo teniendo en cuenta el nodo siguiente en el grupo de nodos. Cuando termina de procesar el paquete lo reenvía al nodo siguiente en el grupo de nodos. Este proceso se repite en cada nodo hasta llegar al destino final, instalando las reglas de flujo en todos los nodos de la ruta con un solo paquete. En la figura **3-2(b)** se muestra también la estructura del paquete OpenPath, pero en este caso, para un ejemplo concreto de 3 nodos en el path, concretamente los nodos 1-3-6. Además, se ha definido una única regla de enrutamiento ($NoR = 1$), por lo que en este caso de los 16 posibles bytes a ocupar por este grupo *Flow Rules*, se han ocupado únicamente 6 bytes. En el caso del ejemplo, a partir del Byte 16 se identifican los nodos que intervienen en la ruta, usando dos bytes para identificar cada uno de ellos.

3.2.2. Arquitectura propuesta: SDN WISE-TSCH

Para garantizar los requisitos altamente deterministas en el ámbito industrial, se propone SDN WISE-TSCH, que amplía las prestaciones de SDN WISE. Para ello, en la capa de control de acceso al medio, se ha sustituido el protocolo CSMA por el protocolo TSCH. Además, el paquete OpenPath, que instalaba las reglas de enrutamiento en cada nodo, se modificó para integrar la planificación de recursos TSCH, lo que da lugar al paquete OpenPathTSCH, que permite una rápida configuración de la red por flujos, con un aumento mínimo del tráfico de control. Los periodos de generación de paquetes se sincronizan en función de la planificación recibida. Para explotar esto, se ha desarrollado un sistema en el controlador que integra tres procesos de la capa de aplicación: el administrador de tráfico (*Traffic Manager*), el proceso de enrutamiento (*Routing Process*) y el planificador TSCH (*TSCH Scheduler*) que se muestra en la figura **3-3**. Estos serán abordados en este capítulo, el administrador de nodos móviles (*Mobile Node Manager*) se explica en profundidad en el capítulo 5.

En la arquitectura SDN de la figura **3-3**, se muestran los dos planos principales típicos en SDN, Control y Datos. Además, se incluye una capa sobre el plano de Control, la capa de aplicación (*Application layer*), donde se ofrecen diferentes servicios a través de distintas aplicaciones SDN (SDN App). En el plano de Control se sitúa el controlador SDN que se encarga de gobernar y dirigir la manera en que los datos son transportados. Por último, en el nivel inferior de la arquitectura se sitúa el plano de Datos, aquí se encuentran los dispositivos SDN como el switch SDN, encargado de reenviar los datos basándose en las instrucciones recibidas por el controlador. También en esta tercera capa se implementa una IWSN situándose en primer lugar el sink (S) que enlaza con el switch SDN y el resto de nodos (N) mayoritariamente fijos, aunque alguno de ellos podría ser móvil (M).

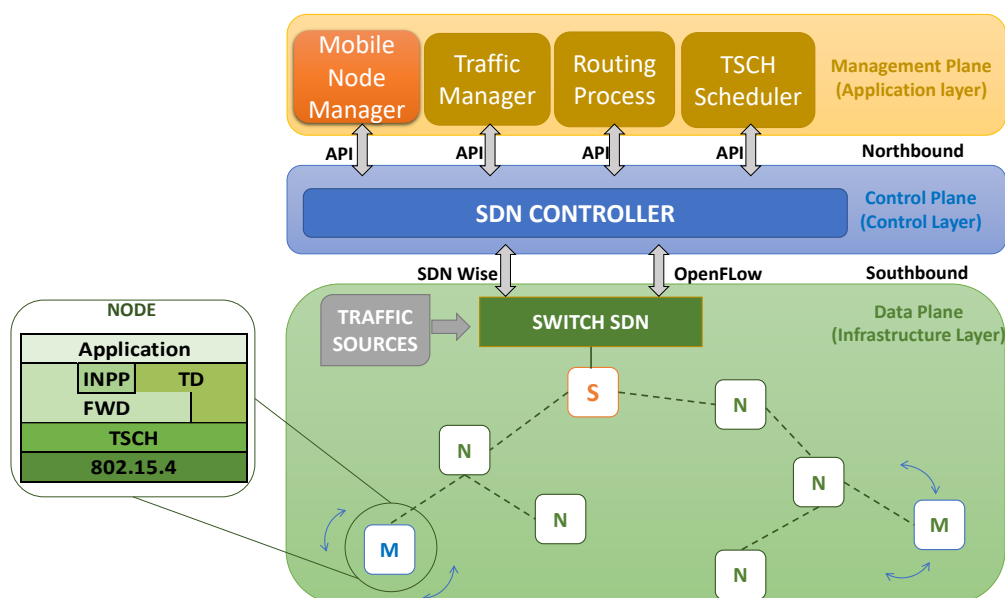


Figura 3-3: Arquitectura general SDN WISE-TSCH

La implementación de SDN en la IWSN otorga un conocimiento total de la red a cambio de aumentar el tráfico de control y el tiempo de convergencia. Este tráfico extra es debido a los mensajes periódicos de los nodos, donde se informa sobre los vecinos, la batería, RSSI y el nivel de congestión. Con un conocimiento topológico completo de la red, la planificación de recursos TSCH puede hacerse de forma optimizada basándose en diversos parámetros, como la periodicidad del tráfico, nivel de congestión, separación entre los nodos, y rutas con diferentes políticas de QoS. Esta planificación se puede ir adaptando de acuerdo con las estadísticas de congestión de los nodos y calidad del enlace. El objetivo es conseguir rutas diferentes sobre las que se puedan distribuir los distintos flujos de tráfico, garantizando los parámetros de calidad de servicio. El uso de múltiples rutas permite que se distribuya el tráfico sobre la mayor cantidad de nodos posible, logrando un balanceo de carga que reduce la congestión y consumo de energía de los nodos centrales. Con esta arquitectura se consigue una red IWSN TSCH completamente definida por software, para ello se han realizado los siguientes desarrollos:

- Establecer en el controlador un administrador de tráfico, que recibe como parámetros los descriptores de flujos esperados de la red, deadline, sensibilidad al delay y a la pérdida de paquetes.
- Desarrollar un proceso de enrutamiento que genere múltiples rutas para los flujos especificados en los descriptores de tráfico, considerando los parámetros de calidad de servicio.
- Diseñar e implementar un planificador TSCH que reciba las rutas y planifique los flujos de acuerdo con sus deadlines.

- Se ha realizado un cambio de capa MAC en SDN WISE a 802.15.4e TSCH sobre Contiki-NG.
- Se han modificado elementos originales del SDN-WISE como el OpenPath, para enviar información de la planificación TSCH y evitar un incremento del tráfico de control.

El resultado es una separación lógica de los flujos de datos, con una asignación específica de recursos físicos (celdas) para cada tipo de tráfico, denominada slicing. Esta fragmentación de la red permite diferenciar los flujos y garantizar diferentes calidades de servicio. Por eso, los apartados siguientes muestran la integración de las diferentes componentes hasta tener una red de sensores totalmente programable, desde el plano de aplicación en el controlador hasta el control de acceso al medio en los nodos. La primera etapa es la integración de los 3 procesos de la capa de aplicación: QoS, algoritmo de enrutamiento y planificador TSCH; la segunda etapa es el control de la capa MAC en los nodos, que incluye la adición de la capa TSCH al SDN WISE, y el paquete que permite configurar la capa MAC con el resultado de la etapa anterior.

3.3. Descriptor de tráfico y QoS

Debido a que esta implementación de SDN-WISE permite tener control sobre la capa MAC (TSCH) y la capa de reenvío (*Forwarding*) de la IWSN, es posible aplicar conceptos de las redes deterministas (*DetNet*) [66], que se basan en describir los requisitos de cada flujo y garantizar los parámetros de QoS, por medio de la reserva de recursos. Esta información por flujo está contenida en el propio controlador en el descriptor de tráfico. El descriptor de tráfico en las *DetNet* contiene especificaciones del tráfico, indicando parámetros como el tamaño de paquete, el intervalo de transmisión y el número máximo de paquetes por intervalo [66]. Estas especificaciones se utilizan para asignar los recursos en la red, como el ancho de banda y el tiempo de transmisión. En esta implementación, el gestor de tráfico alojado en el controlador SDN contiene el descriptor de flujos, el cual está conformado por los campos: prioridad, deadline [ms], nodo origen y nodo destino. Los requisitos de QoS están relacionados con la prioridad de acuerdo con la tabla 3-1. El controlador SDN ordena los flujos de esta tabla según prioridad y deadline, ya que el orden influye en los resultados de los procesos posteriores.

Prioridad	Retardo/Delay	Pérdida de paquetes	Tipo de datos
1	Sensible	Sensible	Eventos
2	No sensible	Sensible	Imágenes y datos periódicos
3	No sensible	No sensible	Datos de telemetría

Tabla 3-1: Clasificación de las prioridades

Generalmente, la QoS se basa en la clasificación y el marcado de tráfico para asignar los recursos de red de acuerdo con las diferentes clasificaciones realizadas. Un diseño óptimo de la red no debe dejar a un lado los requisitos de QoS, pues esto podría provocar comunicaciones deficientes. Sin embargo, implementar técnicas de QoS en las IWSN no es algo trivial, ya que este tipo de redes fueron diseñadas para aplicaciones con una tasa de datos baja y no para aplicaciones en tiempo real [43]. Por ello, las IWSN basadas en SDN pueden ofrecer una QoS optimizada, ya que las aplicaciones programables orientadas al controlador pueden ser desarrolladas para implementar y manejar los requisitos de QoS relacionados con los nodos de sensores [44].

Para poder establecer los requisitos en tiempo real para los distintos tipos de datos, se van a establecer tres niveles de prioridad como se muestra en la tabla **3-1**. Las prioridades se establecen según la sensibilidad de los datos al retardo o delay y a la pérdida de paquetes. En el caso de la prioridad 1, mayor nivel de prioridad, los datos son sensibles al delay y a la pérdida de paquetes, rutas con la menor congestión y menor número de saltos en el path. El tipo de tráfico típico para esta prioridad son los eventos. La prioridad 2, se ha establecido como sensible a la pérdida de paquetes, rutas con la mejor calidad de enlace y bajos niveles de congestión. En este caso se trata de la transmisión de información que debe llegar a su destino sin que se pierda ningún paquete, aunque llegue más tarde. Por último, se establece el nivel de prioridad 3 para datos periódicos como medidas de telemetría. En este caso los datos no son sensibles al delay y a la pérdida de paquetes. Sin embargo, los datos periódicos pueden ser asignados a cualquier prioridad. Esto depende de la aplicación específica que utilice estos datos.

Basándose en estas prioridades se deben conservar las métricas delay, jitter y pérdida de paquetes, de acuerdo con cada uno de los 3 tipos de tráfico. Además de esto, en el caso de detectar la caída de uno de los nodos del path, los flujos de mayor prioridad serán enviados haciendo uso de los recursos de las prioridades inferiores. Esto permitirá reducir el impacto del fallo sobre el tráfico de mayor prioridad.

3.4. Algoritmo de enrutamiento

El enrutamiento es la fase central del proceso de integración en SDN-WISE TSCH. Se encargará de posicionar dentro de la IWSN los flujos definidos en el descriptor de tráfico. Este proceso se realiza mediante la selección de los saltos de origen a destino, formando una ruta. La selección de rutas tiene un impacto crítico en el rendimiento de la red. Por esta razón, existen múltiples estudios que proponen protocolos de enrutamiento para obtener el mayor rendimiento en situaciones determinadas como reducir el consumo de energía, aumentar el throughput o implementar QoS [67]. Uno de los más comunes actualmente en las WSN es el RPL donde es posible configurar funciones objetivo (FO) con múltiples parámetros, balanceo de carga y tolerancia a fallos. Sin embargo, las rutas no son óptimas y esto se refleja

provocando congestión en los nodos centrales [68].

En esta fase se plantea segmentar los flujos de tráfico por múltiples rutas, por esto se diseña un algoritmo de enrutamiento multipath y asignar una ruta a cada flujo. Los procesos de enrutamiento multipath, permiten hacer un mejor uso de los recursos de la red, al distribuir el tráfico sobre una mayor cantidad de nodos. Para lograr esto, se debe tener una métrica diferente, que permita obtener múltiples rutas del nodo origen al sink como [51], donde se usa el algoritmo de Dijkstra con una métrica compuesta por múltiples parámetros como el nivel de congestión, nivel de señal (RSSI), y el nivel de la batería de los nodos. Esto permite generar múltiples rutas para diferenciar el tráfico de acuerdo con la QoS.

El controlador del SDN-WISE usa el algoritmo de Dijkstra para calcular la ruta con el menor costo, por defecto la métrica que utiliza es el valor del RSSI. Sin embargo, se puede usar el conocimiento topológico que ofrece SDN para implementar una métrica que acumule la cantidad de tráfico planificado para cada nodo, sin la necesidad de enviar esta información desde los nodos. Esto es posible debido a que la capa MAC será TSCH y los flujos de tráfico están completamente definidos en el descriptor de tráfico. La métrica será la suma de la acumulación del tráfico planificado en cada nodo. El parámetro de acumulación de tráfico se obtiene de acuerdo con el deadline y la ruta obtenida para cada flujo. Por lo tanto, el proceso en cada iteración selecciona la ruta con menor congestión.

El Algoritmo de Dijkstra recibe la topología de la red, que se define mediante los nodos y cada uno de sus enlaces. Cada enlace tiene asignado un peso (*WeightLink*), que es utilizado por el algoritmo para encontrar la mejor ruta. Sin embargo, utilizar una métrica por cada enlace no es óptimo en una IWSN, ya que los nodos solo pueden manejar un enlace unidireccional en un instante de tiempo. Por esto, para representar mejor el comportamiento y las limitaciones de una IWSN se utiliza una métrica global para cada nodo (*NodeUse*), que afecta a cada uno de sus enlaces.

Como el algoritmo de Dijkstra opera sobre los pesos de los enlaces y no sobre la métrica de los nodos, el peso de los enlaces se ajusta en función de las métricas globales de los nodos que componen el enlace (nodos i y j), como se representa en la ecuación 3-1. La variable *NodeUse* es la métrica global de cada nodo que representa la densidad de tráfico, y corresponde a la suma de cada uno de los flujos enrutados previamente, esto es desde $k = 1$ hasta $k = n - 1$. La tasa de paquetes para cada flujo k se obtiene en función del *MaxDeadline* de la tabla de flujos y su *deadline* configurado. Esto se representa por un $deadline_{ik}$, que es el deadline en el primer nodo (i) del $link_{ij}$ para el flujo k . El proceso en el segundo nodo (j) es equivalente con los flujos l previamente encaminados para este nodo.

$$WeightLink_{ij} = NodeUse_i + NodeUse_j = \sum_{k=1}^{n-1} \frac{MaxDeadline}{deadline_{ik}} + \sum_{l=1}^{n-1} \frac{MaxDeadline}{deadline_{jl}} \quad (3-1)$$

Por lo tanto, la mejor ruta se define por la suma de los pesos de los diferentes enlaces en cada salto, ecuación (3-2). Ahora el $link_{ij}$ cambia en cada salto p , por lo que los nodos i y j dependen del salto p , representados como ip y jp . El proceso itera sobre una lista con n flujos, en cada iteración se selecciona la ruta con menos congestión. Además, para garantizar que los recursos se asignan primero a los flujos más prioritarios, el gestor de tráfico organiza la lista de flujos por prioridad y plazo.

$$WeightFlow = \sum_{p=1}^{\#hops} \left(\sum_{k=1}^{n-1} \frac{MaxDeadline}{deadline_{ipk}} + \sum_{l=1}^{n-1} \frac{MaxDeadline}{deadline_{jpl}} \right) \quad (3-2)$$

Otra de las ventajas de SDN en IWSN, es que a diferencia de los nodos el controlador no tiene limitaciones de procesamiento. Por ello es capaz de realizar el cálculo de las rutas óptimas y actualizar las métricas de los nodos y los enlaces cada vez que el proceso de enrutamiento asigna una ruta. Debido a que el deadline determina la frecuencia con la que se envía tráfico, es un factor que aumenta la métrica en cada nodo.

Algoritmo 1: Enrutamiento con peso acumulativo

```

1 Entrada:
2    $G=(N, E)$ ; (Topología, donde N es el conjunto de nodos y E el conjunto de
   aristas, cada arista está compuesta por Nodo1, Nodo2 y Peso)
3    $TrafficDescriptor[n][4]$ ; (Matriz con  $n$  flujos y 4 parámetros: Prioridad,
4   NodoFuente, NodoDestino, Deadline)
5 Salida:
6   Path; (Ruta con la menor cantidad de congestión)
7 Inicialización:
8   MaxDeadline = getMaxValue de TrafficDescriptor[n][4];
9   NodeUse[N]; (Acumulador de recursos de cada nodo)
10 for  $i=1 \rightarrow n$  do
11   |   Path[i]=Dijkstra(G, SourceNode[i], DestNode[i]);
12   |   for cada Nodo en Path[i] do
13   |   |   NodeUse[N]+=MaxDeadline/Deadline[i];
14   |   end
15   |   for cada Arista en G do
16   |   |   Edge weight=NodeUse[Node1]+NodeUse[Node2];
17   |   end
18 end

```

En general, en el Algoritmo 1 se observan las siguientes etapas:

- Calcular el mayor deadline de los flujos que se van a enrutar. (Línea 8)
- Tomar una entrada del descriptor de tráfico. (Línea 10 y 11)
- Usar el algoritmo de Dijkstra de acuerdo a la topología y los parámetros del flujo para calcular la ruta. (Línea 11)
- Usar el deadline para actualizar las métricas de los nodos utilizados. (Línea 13 y 16)

Este proceso se repite para cada una de las entradas del descriptor de tráfico. Por lo tanto, las primeras rutas que se calculen tendrán menor latencia. Las rutas siguientes se reparten sobre los nodos menos congestionados. Con esto se consigue generar múltiples rutas para llegar al sink, lo que permite que las transmisiones y recepciones no se concentren en los nodos centrales, balanceando la carga sobre la mayor cantidad de nodos posibles y a su vez aumentando el tiempo de vida de la red.

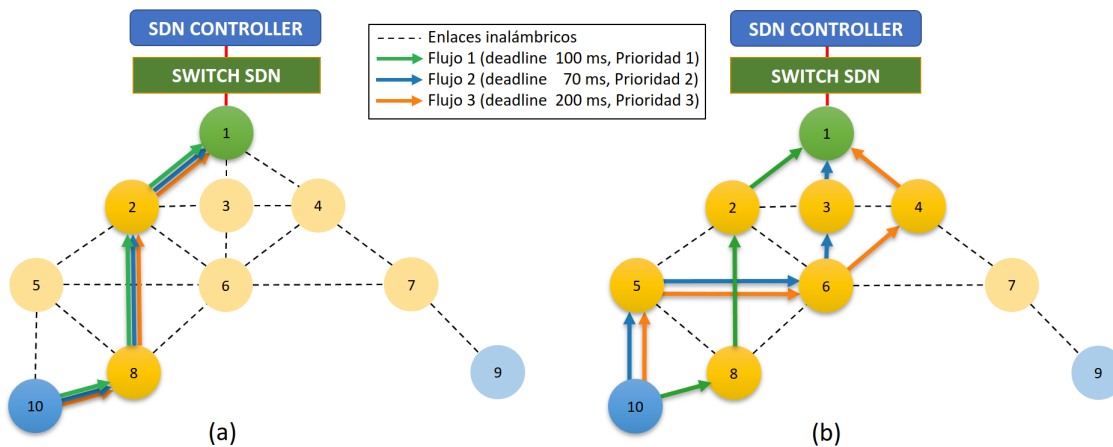


Figura 3-4: Flujo entre los nodos 10 y 1. (a) Dijkstra simple basado en el RSSI, (b) Dijkstra acumulado con balanceo de carga

En la figura 3-4, se compara el algoritmo de Dijkstra utilizando la métrica del RSSI y el Dijkstra con la métrica propuesta. En la topología, el nodo 10 tiene tres flujos organizados y planificados según su prioridad. Al generar paquetes en el nodo 10, de acuerdo con los deadlines establecidos, en un intervalo de un segundo se generan diez paquetes de prioridad uno, catorce de prioridad dos y cinco de prioridad tres, lo que da como resultado una tasa de 29 paquetes por segundo.

En la figura 3-4(a), se puede observar definida una ruta con tres saltos, donde los nodos intermedios 8 y 2 reciben la carga completa del nodo 10 (29 paquetes/segundo). La métrica propuesta modifica estas rutas, como se observa en la figura 3-4(b). En promedio, estas nuevas rutas presentan un número mayor de saltos, ya que hay dos rutas de 4 saltos y una de 3. De este modo, la carga de tráfico para el nodo 8 y 2 se reduce a 10 paquetes/segundo,

ya que se ha balanceado el tráfico a través de diferentes nodos. Aunque el mayor número de saltos obtenido por la métrica propuesta implica un mayor consumo total de energía en la red, la distribución del tráfico entre más nodos evita la concentración de consumo en un nodo específico. Este balanceo de carga permite reducir la congestión y consumo de energía en los nodos centrales. Además, el uso de nodos diferentes aumenta el paralelismo en la etapa siguiente de planificación de recursos.

En general, el uso de la métrica propuesta consigue un aumento del throughput y el tiempo de vida. Otra ventaja de utilizar múltiples rutas es el aumento de la tolerancia a fallos, ya que es posible redirigir el tráfico por rutas alternas sin esperar que el controlador reconfigure la red. Por ello, en caso de fallo de un nodo, el tráfico de mayor prioridad se envía por la ruta de menor prioridad, así se reduce el impacto del fallo sobre los paquetes de alta prioridad y se descarta el flujo menos prioritario. Esta tolerancia a fallos demuestra inicialmente que la configuración desde el controlador SDN no representa una limitación significativa en estos casos. No obstante, esta tesis no pretende profundizar en los mecanismos específicos de tolerancia a fallos que se pueden implementar en una red SDN. El proceso de enrutamiento es independiente de los nodos, no tiene ninguna limitación de hardware y puede ser modificado directamente en el controlador, ya que los nodos solo deben agregar las rutas que se den como resultado. Esto incrementa la flexibilidad de la red, ya que es fácilmente adaptable a entornos que requieran métricas más específicas.

3.5. Planificador TSCH en SDN WISE-TSCH

La planificación TSCH es el último proceso a integrar. Se encarga de plasmar las rutas calculadas por el proceso de enrutamiento de modo que se optimicen las transmisiones por medio de la planificación de los slots de recepción y transmisión de cada nodo. El resultado obtenido en este proceso será el que el controlador envíe a los nodos. El problema general es que la mayor parte de los estudios están enfocados en apartados específicos, omitiendo algún componente importante para una red TSCH SDN. Los trabajos [64], [63] utilizan los elementos de la pila de protocolos 6TisCH para lograr optimizar la planificación TSCH de forma centralizada e híbrida. Además, incorporan un concepto similar al de *slicing* denominado *tracks* para servicios de alta prioridad, sin embargo, las pruebas se realizan con rutas y celdas instaladas de forma manual en cada nodo, lo que omite el envío de la planificación a los nodos y el tráfico de señalización que esto genera. Otros trabajos como [44] y [65], tratan la necesidad de asociar SDN y TSCH en las aplicaciones industriales. Sin embargo, en este caso se trata de estudios completamente teóricos. Teniendo en cuenta la necesidad de tratar los requisitos de las SD-WSN de modo distinto a las SDN clásicas, en [52] los autores proponen la combinación de TSCH con el protocolo SDN-WISE para poder cumplir con los requisitos de movilidad de las IWSNs. Dado que los resultados obtenidos son preliminares y no abordan la viabilidad en una implementación real, los mismos autores presentan en [62]

el protocolo FTS-SDN (*Forwarding and TSCH Scheduling over SDN*) que proporciona buenos resultados tanto en experimentos como en simulación de escenarios reales. No obstante, tampoco en este trabajo se aborda en detalle el planificador TSCH y el tráfico de control que aumenta en una implementación SDN sobre WSN, ya que el estudio se centra más en la movilidad de los nodos.

Por todo esto, en esta sección se implementa un planificador TSCH centralizado, como parte de un sistema de planificación de recursos multicapa con SDN, que se termina de desarrollar en la sección 3.4. El planificador TSCH recibe las rutas del algoritmo de enrutamiento y el descriptor de tráfico para planificar los flujos de datos y cumplir los requisitos de QoS. Además, los resultados de esta planificación serán enviados a los nodos con el menor tráfico de señalización posible.

El proceso propuesto permite planificar los flujos de acuerdo con sus deadlines y reducir el delay en función de su prioridad. Para cumplir los deadlines, cada flujo se planifica múltiples veces con un desplazamiento u *offset*, que depende de la longitud del slotframe y el deadline. Dado que el slotframe es cíclico en el tiempo, su longitud determina la distancia máxima entre celdas es decir, el intervalo máximo que un nodo puede permanecer en modo de espera (sleep) antes de que se active un slot para transmitir o recibir datos. Modificar el tamaño del slotframe de acuerdo con los deadlines del descriptor de tráfico permite cumplir estos requisitos minimizando las transmisiones, lo que reduce el consumo de energía. Para cada repetición de un flujo dentro del slotframe, se planifica con una rotación de canal. El delay es un parámetro que se considera dentro del protocolo de enrutamiento y que, en una red TSCH, está relacionado con el número de celdas necesarias para llegar al destino. Por lo tanto, los flujos con sensibilidad al delay están compuestos por una agrupación consecutiva de celdas.

Los procesos para el planificador TSCH propuesto, se muestran en el Algoritmo 2, el cual contiene dos matrices con dimensiones $[NChannels]$, $[SlotFrameLength]$. Estas se encargan de registrar los nodos planificados, una matriz es para los nodos de transmisión ($NodesInTX$) y la otra para los de recepción ($NodesInRX$). Cada vez que se va a asignar una celda, se accede a estas matrices para verificar que el nodo y el canal no estén previamente asignados, garantizando que las transmisiones estén libres de conflictos.

Algoritmo 2: Planificador SDN WISE-TSCH

```

1 NChannels; (Número de canales físicos utilizados);
2 TrafficDescriptor[n][4]; (Matriz con  $n$  flujos y 4 parámetros: Prioridad,
   NodoFuente, NodoDestino y deadline, ordenada por prioridad y deadline);
3 Path[n]; (Resultado del algoritmo de enrutamiento para cada entrada en
   TrafficDescriptor);
4 NodesInRX[NChannels][SlotframeLength]; (Matriz que almacena los nodos
   planificados en Rx);
5 NodesInTX[NChannels][SlotframeLength]; (Matriz que almacena los nodos
   planificados en Tx);
6 for cada ruta en Path[ $n$ ] do
7   Nodes[k]=getNodesfrom ruta; (Toma cada uno de los nodos de la ruta y los
   asigna a un array con k elementos, donde k es el total de nodos de la ruta);
8   for  $j=1 \rightarrow k$  do
9     (Para cada uno de los nodos (Node[ $j$ ]) buscar un slot libre de conflictos);
10    while Slot está ocupado do
11      if NodesInTX[Slot] contiene Node[ $j$ ] o Node[ $j+1$ ] then
12        | Slot = siguiente Slot;
13      else
14        if NodesInRX[Slot] contiene Node[ $j$ ] o Node[ $j+1$ ] then
15          | Slot = siguiente Slot;
16        else
17          | Slot está libre;
18        end
19      end
20    end
21    FreeChannel = ChannelOffset libre en este Slot;
22    NodesInTX[FreeChannel][Slot] = Node[ $j$ ]; (Planificar los nodos Tx y Rx) ;
23    NodesInRX[FreeChannel][Slot] = Node[ $j+1$ ];
24    if Last hop then
25      | Planificar(Path, Slot+deadlineOffset); (Planificar de nuevo este Path,
        con un deadline offset.);
26    else
27      | Planificar próximo salto;
28    end
29  end
30 end

```

El proceso comienza con la matriz de *Paths* generada en el Algoritmo 1, que contiene las rutas para cada uno de los flujos configurados. Esta matriz está ordenada por prioridad y deadline, siendo la prioridad más alta la primera en planificarse. El procedimiento a seguir es el siguiente:

- Descomponer la ruta en saltos.
- Para cada salto, verificar la disponibilidad de los nodos en ese timeslot. Si alguno de los nodos ya está planificado, pasar al siguiente timeslot y verificar nuevamente. Continuar hasta encontrar un timeslot en el que ambos nodos estén disponibles.
- En ese timeslot, buscar un canal libre y agregarlo a la planificación.
- Cuando se termine de planificar un path, verificar el deadline y planificarlo nuevamente, con el desplazamiento de timeslots necesario.

Para generar la planificación se deben tener en cuenta las siguientes restricciones:

- Evitar conflictos en los timeslots. Los nodos son *halfduplex*, no se puede transmitir y recibir en el mismo instante de tiempo.
- Tener en cuenta la interferencia entre los nodos. Este algoritmo no reusa el canal en el mismo timeslot.
- Los nodos sin conflictos pueden planificarse en paralelo.
- Planificar los deadlines del descriptor de tráfico.
- Los deadlines deben cumplirse dentro del slotframe y al enlazar al siguiente.
- Los flujos que se repitan en el slotframe deben estar planificadas con una rotación de canal.

A continuación, se indican las consideraciones generales a tener en cuenta para decidir el tamaño del slotframe:

- Ancho de banda y consumo de energía: La longitud del slotframe está directamente relacionada con el consumo de energía, slotframes cortos aumentarán el ancho de banda y el consumo de energía, mientras que los slotframes largos reducen el consumo de energía y el ancho de banda.
- Tiempo de actualización de la topología: El controlador se actualiza generalmente al inicio de cada slotframe en los slots compartidos, usados por los nodos para inundar la red con beacons, que permiten construir la topología. Por lo tanto, en redes estables el slotframe puede extenderse. En casos específicos es posible agregar slots compartidos en diferentes puntos del slotframe.

- Para asegurar que la comunicación realiza una rotación sobre todos los canales disponibles, el número de slots en el slotframe debe ser un número primo.

El uso de deadlines diferentes para cada flujo, permite obtener las ventajas de los slotframe largos y cortos, ya que se puede tener mayor ancho de banda y deadlines cortos en un flujo específico, sin necesidad de repetir los demás flujos del slotframe, reduciendo el consumo de energía

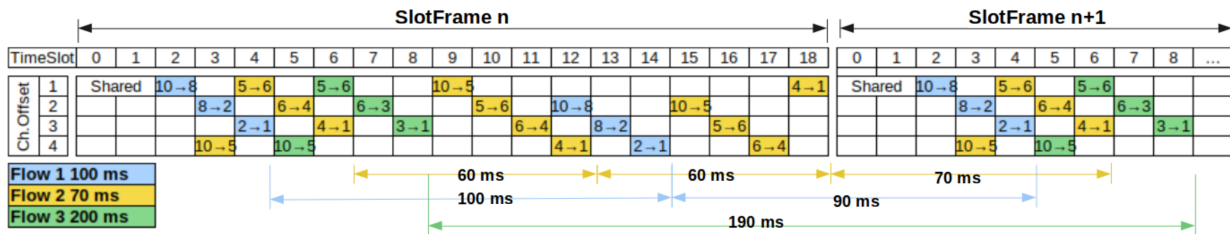


Figura 3-5: Panificación de múltiples flujos con diferentes deadlines

La figura 3-5, muestra la planificación de 3 flujos en un slotframe de 19 timeslots (190 ms). Esta longitud permite cumplir el mayor deadline (200 ms) en cada repetición del slotframe. Los demás deadlines son planificados múltiples veces con una rotación del channel offset, de forma que siempre el tiempo entre los paquetes sea menor al deadline. Si esta planificación fuera recortada a 8 timeslots, los tres flujos tendrían una periodicidad de 80 ms, lo que se acerca a cumplir todos los deadlines. Sin embargo, cumplir los deadlines con demasiada holgura eleva de forma considerable el consumo de energía. Por esto la importancia de planificar los flujos de forma dinámica.

3.6. Modificación SDN WISE

Una de las aportaciones de esta tesis es una red IWSN TSCH completamente definida por software, integrando múltiples procesos del plano de aplicación. Estos procesos permiten hacer un slicing por flujos para aplicar diferentes parámetros de calidad de servicio. Para lograrlo, en los apartados anteriores se ha explicado la segmentación lógica y física de las rutas para asignar a cada flujo. En este apartado se especifica la operación de los nodos. Para esto se han realizado modificaciones significativas al protocolo SDN-WISE. En primer lugar, se ha cambiado la capa de control de acceso al medio (MAC), y en segundo lugar se ha extendido el funcionamiento del paquete OpenPath para llevar la planificación TSCH.

Las modificaciones se realizan sobre la versión 3.0.6 de SDN WISE Contiki, disponible de forma abierta. Debido a que Contiki tiene una nueva versión que dispone de mejoras en los temporizadores para TSCH y soporte para una mayor cantidad de nodos [69], se migró el proyecto a esta plataforma. Para testear dicha implementación de SDN en la IWSN se tomará como referencia la arquitectura propuesta en la figura 3-3. Para el sink y los nodos

se van a utilizar la plataforma OpenMote-B.

A continuación, se explican las modificaciones realizadas en el paquete OpenPath, al que se denominará OpenPathTSCH, para codificar la información de la planificación TSCH y se muestra un ejemplo detallado para su validación además de mostrar cómo los nodos llevan a cabo la decodificación de la trama recibida.

Modificación del paquete OpenPath

Para incluir funcionalidades TSCH sin incrementar el tráfico de control, se extiende el funcionamiento del paquete OpenPath añadiendo campos específicos para la asignación de la planificación TSCH, como se muestra en la figura 3-6. Los campos añadidos se muestran resaltados en naranja, y corresponden a dos grupos de bytes a los que se les ha nombrado “*Indicators*” y “*TSCH Schedule*”. El grupo *Indicators* está formado por 3 bytes, situados después de Flow Rules. Cada byte corresponde con los siguientes parámetros:

- *NR* (Número de Repeticiones, 1 Byte): es un campo que indica la frecuencia con la que se repite la ruta dentro del slotframe. Está asociado de forma directa al deadline del flujo de tráfico que usará esta ruta y al tamaño del slotframe. Se usa para informar al nodo de la cantidad de celdas de TX que debe instalar. Además, el bit más significativo (MSB) indica la dirección del flujo, así se utiliza un 0 para el flujo del sink a los nodos (Downlink) y un 1 para el tráfico de los nodos hacia el sink (Uplink).
- *NN* (Número de Nodos, 1 Byte): el número total de nodos que conforman la ruta. En el OpenPath original este parámetro no era necesario, ya que se podía obtener a partir de la longitud del paquete y el número de reglas de flujo. Por el contrario, para las modificaciones desarrolladas, es necesario informar para que estén correctamente delimitados los bytes de planificación de cada nodo.
- *SS* (Tamaño del slotframe, 1 Byte): el tamaño del slotframe es un parámetro que tiene implicaciones en el throughput y en la eficiencia energética, ya que para una misma asignación de celdas incrementar el tamaño del slotframe implica más celdas con la radio apagada, y reducirlo se traduce en mayor frecuencia entre transmisiones. Sin embargo, para mantener el sincronismo los nodos deben tener la misma configuración por lo que se hace de forma explícita con este byte.

Los bytes asociados al *TSCH Schedule* se han situado al final del paquete. Cada celda en la planificación está codificada con 2 Bytes, uno para el Channel Offset y otro para el timeslot. El número de celda que se añade aquí, depende del *NN* y el *NR*. Cada nodo debe tener una celda por repetición del path, por lo tanto, el número de celdas de un nodo será igual al *NR*. Estas celdas están ordenadas en el mismo orden que el path, es decir, las primeras *NR* celdas corresponde al primer nodo en el path y así sucesivamente.

Byte	10 Bytes										1:16 Bytes			3 Bytes			2*NN Bytes		2*NR*(NN-1) Bytes				
Description	Length (Bytes)	Network ID	Source Node	Destination Node	Packet type	TTL	Next Hop	NoR Number of rules	RuleID	:	Rule NoR	NR Number of Repetitions	NN Number of nodes	SS Slotframe Size	First node in the path	:	Last node	Channel offset	Slot				
	Header							Flow Rules				Indicators		Nodes in the Path		TSCH Schedule							

Figura 3-6: Estructura genérica del paquete OpenPathTSCH

Validación de la implementación del OpenPathTSCH

Para validar los mecanismos del paquete OpenPathTSCH se tendrá en cuenta una topología con 10 nodos. En este ejemplo el path está formado por 5 de ellos como muestra la figura 3-7. En este ejemplo se utiliza tráfico downlink, donde el bit más significativo (MSB) de *NR* está a 0. En el path seleccionado, el tráfico debe ir del nodo 1 al 10 y salta en cada uno de los nodos intermedios. Se comprobará que a partir del paquete OpenPathTSCH se puede modificar el comportamiento de la capa MAC TSCH en todos los nodos de la ruta de este ejemplo (1-2-5-8-10). Este flujo tendrá 2 repeticiones (*Number of Repetitions*) dentro del slotframe que tiene una longitud de 11 timeslots, figura 3-7, donde se muestra únicamente la planificación de los nodos pertenecientes al path.

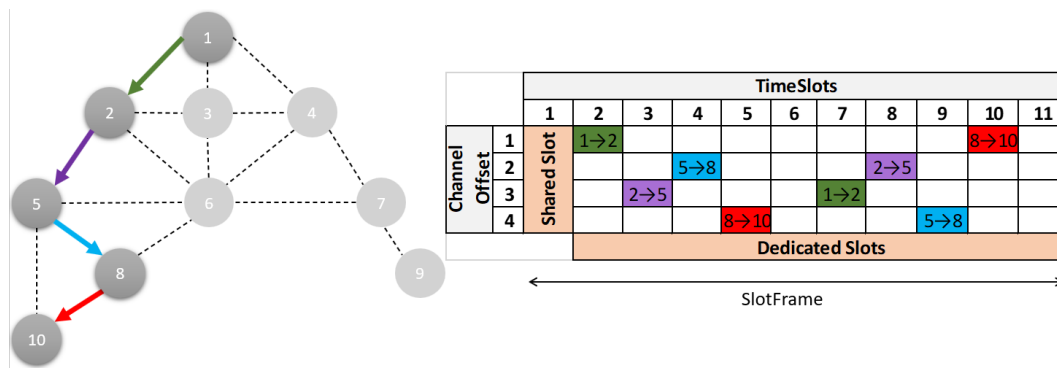


Figura 3-7: Topología de 10 nodos, planificación TSCH de un path con 5 nodos y NR=2

Los timeslots en el slotframe deben asignarse en el orden del flujo de tráfico para evitar que se acumulen paquetes en el buffer de los nodos. Por lo tanto, en este ejemplo se asigna primero el nodo 1 que deberá transmitir al nodo 2 en el Channel Offset 1, timeslot 2 y también en el Channel Offset 3, timeslot 7 según la planificación TSCH. Esto se codifica en el paquete OpenPathTSCH en los bytes dedicados a la planificación TSCH utilizando 4 Bytes del siguiente modo (Channel Offset-Slot-Channel Offset-Slot): 1-2-3-7. Como es el nodo origen no hay celdas de recepción para este flujo, ya que la planificación solo contempla las transmisiones, puesto que las recepciones están implícitas. La información de transmisión codificada en las celdas puede agruparse como la información de transmisión de cada nodo. La longitud de cada agrupación depende del número de repeticiones. En la figura 3-8, se

muestra el paquete OpenPathTSCH para el caso concreto del path de 5 nodos. En esta imagen se etiqueta como “Tx Node1” al grupo de bytes que incluye las dos celdas con información del Nodo 1. El grupo ocupa 4 bytes porque $NR = 2$ en este caso. La agrupación de las celdas se organiza según el orden de reenvío, es decir, primero el grupo de celdas del primer salto en el path y luego el grupo de celdas del salto siguiente, y así sucesivamente.

Del mismo modo, el nodo 2 deberá transmitir al nodo 5 en el Channel Offset 3, Slot 3 y en el Channel Offset 2, Slot 8, esto se codificará de nuevo utilizando 4 Bytes 3-3-2-8. Este nodo sí debe tener celdas de recepción para recibir los paquetes enviados por el nodo 1, en las celdas 1 y 2: Channel Offset 1 Slot 2 y Channel Offset 3 Slot 7. Como estos corresponden a los Bytes de transmisión del nodo 1 no se agregan al paquete. De este modo, la codificación de la planificación estará compuesta solo por la información de transmisión de cada nodo y las celdas de recepción se deducen a partir de las celdas de transmisión del nodo anterior. La planificación del resto de los nodos sería, Nodo 5: 2-4-4-9 y para el Nodo 8: 4-5-1-10. Para el Nodo 10 no habría codificación de la planificación, ya que solo se codifica la transmisión y este nodo solo recibe.

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44									
Description	Length (Bytes)	Network ID	Source Node	Destination Node	Packet type	TTL	Next Hop	NoR Number of rules	Rule n,1 (Each routing rule occupies 5 Bytes)	NR Number of Repetitions	NN Number of nodes	SS Slotframe Size	First node in the path	Second node	Third node	Fourth node	Last node	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot	Channel offset	Slot											
Value	45	1	1	1	2	2	5	100	0	1	1	114	0	0	0	40	2	5	11	1	1	2	2	5	5	8	8	10	10	1	2	3	7	3	3	2	8	2	4	4	9	4	5	1	10									
	Header							Flow Rules					Indicators		Nodes in the Path						TSCH Schedule																																	

Figura 3-8: Estructura del paquete OpenPathTSCH con la planificación del path con 5 nodos y NR=2

Decodificación de los elementos en la trama

El nodo receptor debe decodificar la información contenida dentro de la trama recibida, empezando por el encabezado que tiene una estructura fija de 10 Bytes donde en primer lugar compara los valores del campo destino y próximo salto, para verificar que le pertenece esta trama. La estructura siguiente es variable y la analiza basándose en el tipo de paquete, en este caso OpenPath es tipo 5. Por lo tanto, el byte siguiente determina la longitud de la regla de flujo, que para este ejemplo es de 6 bytes por tener una única regla de coincidencia. Después de las reglas se indicaría el número de repeticiones, $NR = 2$, para el ejemplo en el Byte # 16. El valor de NR también indica la dirección del flujo, valores menores a 128 son downlink (MSB=0). La diferencia entre uplink y downlink, es que la asignación de los nodos de transmisión y recepción se hace de forma inversa, ya que el paquete OpenPathTSCH se transmite en la dirección opuesta del flujo. A continuación, se indica el número de nodos en el path (NN), en este caso $NN = 5$, ya que solo son 5 los nodos que pertenecen al path,

aunque haya 10 en la IWSN. A continuación, se indica la longitud del slotframe, $SF_{sz} = 11$, como se muestra en la planificación TSCH de la figura **3-7**. Después se indica cada uno de los nodos en el path. Luego compara cada uno de los nodos en el path hasta encontrar su ID, donde obtendrá qué posición ocupa dentro del path. Si el nodo que ha recibido el paquete es el 5, el valor que obtiene como posición será igual a 3. A continuación, conoce cuál es el destino del path que corresponde al último nodo, en este caso el nodo 10; cuál es el nodo anterior a él en el path, en el caso del ejemplo sería el nodo 2; y el nodo siguiente donde debe reenviar el paquete que para el ejemplo corresponde al nodo 8.

Finalmente, se encuentran los bytes de la planificación, donde debe obtener la planificación específica para su posición en el path. El proceso es el siguiente, el número de celdas que debe leer es igual al número de repeticiones (NR). La primera celda que le corresponde depende de su posición en el path: $1 + (posición - 1) \cdot NR$. Para este ejemplo, el nodo 5, que se encuentra en la posición 3, debe leer las celdas 5 y 6 o lo que es lo mismo, los bytes 9 a 12 desde el inicio de la planificación. De forma general, el byte inicial de la planificación será variable. Como se ha visto, su posición depende de la cantidad de reglas de flujo y nodos en la ruta. Puede expresarse de la siguiente forma, Byte $(10 + 5 \cdot NoR + 3 + 2 \cdot NN + 1)$. La longitud de esta planificación (L) vendrá determinada de acuerdo a la ecuación 3-3 por la cantidad de nodos en el path (NN), el número de repeticiones (NR) y los dos bytes necesarios para el canal y slot.

$$L = 2 \cdot NR \cdot (NN - 1) \quad (3-3)$$

El tamaño máximo de los paquetes 802.15.4e es de 127 bytes. Al reducir los 11 bytes que necesita la capa MAC se tienen 116 bytes de tamaño máximo de los paquetes SDN-WISE TSCH. Para el caso del OpenPathTSCH esto equivale a la ecuación 3-4, siendo L el valor obtenido mediante la ecuación 3-3.

$$Header + 1 + 5 \cdot NoR + Indicators + 2 \cdot NN + L \leq 116 \quad (3-4)$$

Para una planificación con una longitud típica del slotframe igual a 1 segundo (101 Slots) y un flujo con 4 saltos, el número máximo de repeticiones que podría planificarse para este flujo sería 11, lo que equivale aproximadamente a 1 paquete cada 100 ms.

3.7. Evaluación del rendimiento

El escenario planteado en la figura 3-4 será el utilizado en las pruebas, este consta de una red de sensores formada por 10 nodos y el controlador SDN que recibe la información de la IWSN desde el sink, nodo 1. Los nodos 9 y 10, de color azul, generarán el tráfico de distintos niveles de prioridad. El resto de los nodos, los amarillos, se encargan de enrutar el tráfico generado. La red tiene 3 tipos de tráfico, con distintas prioridades de acuerdo con el delay y el PDR, tal y como se indicó en la tabla 3-1. Cada tipo de tráfico es asignado por el controlador SDN a una ruta diferente, determinada por la congestión esperada en cada nodo. Para cada una de las rutas se genera una planificación TSCH que se adapta a los requisitos. En la tabla 3-2 se muestran los parámetros utilizados en la simulación. En general, se utilizaron los valores por defecto excepto para las retransmisiones MAC, que por defecto tienen un valor de 7 en Contiki-NG.

Parámetros	SDN-WISE TSCH
Tiempo de simulación	7 min (42.000 timeslots)
Número de canales	4 canales
Join sequence	4 canales
Período EB	16 s
Retransmisiones MAC	0
Ráfaga	Deshabilitada
Período de Keep Alive	60 s
timeslot	10 ms

Tabla 3-2: Parámetros utilizados en la simulación

3.7.1. Escenarios y métricas

A continuación, se describen los tres tests que se llevarán a cabo junto con las métricas a evaluar, tanto mediante simulación como en el testbed.

Test 1: Eficiencia del protocolo

En primer lugar, se evalúan las características generales de la implementación. Estas se usarán para comparar la eficiencia y el grado de mejora obtenido respecto de Orchestra de acuerdo con las siguientes métricas:

- Tiempo de convergencia: es el tiempo necesario para que el controlador SDN cree la imagen de la topología basándose en los paquetes de reporte que recibe de cada uno de los nodos.

- Cantidad y tamaño de los paquetes de control: el intercambio de paquetes de beacons y reporte es un tráfico esencial para el funcionamiento, pero debe ser controlado y no debe afectar a los flujos configurados.
- Tiempo de configuración: es el tiempo que transcurre desde que el controlador envía un OpenPathTSCH y en el nodo destino comienza la transmisión.

Para las pruebas de los tests 2 y 3, se definen 5 flujos de tráfico de datos como se muestra en la tabla **3-3**. Los flujos están definidos en el controlador SDN por una prioridad, un deadline, el nodo origen y el destino. Así el controlador SDN puede asignar los recursos necesarios para proporcionar una calidad de servicio óptima. El delay y la pérdida de paquetes determinan el nivel de prioridad (primera columna de la tabla **3-3**), y son parámetros que se deben conservar en la implementación para cada una de las pruebas de flujo.

Prioridad	Deadline	Flujo
1	100 ms	10 → 1
1	150 ms	9 → 1
2	70 ms	10 → 1
2	300 ms	9 → 1
3	200 ms	10 → 1

Tabla 3-3: Flujos de tráfico de datos

Test 2: Retardo en la configuración

En este caso se obtiene el rendimiento de la implementación. Las métricas que se evaluarán son:

- El retardo extremo a extremo o *End-to-End Delay* (E2E delay): cantidad de timeslots necesarios para transmitir los paquetes de origen a destino.
- *Packet inter-arrival time* (PIAT): es la cantidad de timeslots que transcurren entre cada recepción de un mismo flujo en el nodo destino. Es la métrica principal, ya que permite evaluar cómo se adapta la red a los estrictos requisitos temporales. A partir de esta métrica se obtiene la tasa de satisfacción de los requisitos de calidad de servicio, que se engloba dentro de la tasa de cumplimiento de deadline (DSR - *Deadline Satisfaction Ratio*).
- Consumo de energía: en una red TSCH puede obtenerse a partir del tiempo que ha pasado el nodo en los estados de transmisión y recepción. El tamaño de los paquetes afecta a este tiempo. La relación que existe entre el tiempo de recepción y transmisión del nodo con el tiempo total es el ciclo de funcionamiento de la radio (RDC - *Radio Duty Cycle*), que se usará para comparar el rendimiento energético en cada escenario.

- Tiempo de vida de la red o *Network lifetime*: es el tiempo total de la red operativa hasta que el primer nodo se descarga. Para calcularlo se tienen en cuenta los valores de la tabla 3-4, y el tiempo que cada nodo pasa en estos estados.

Modo de operación	Consumo
Microcontrolador – Modo: Sleep	1,3 μ A
Microcontrolador – Modo: Activo	7 mA
Radio – Rx	20 mA
Radio – Tx	24 mA
Radio – Sleep	1 μ A

Tabla 3-4: Modos de operación y consumo de energía para el OpenMote-B

Test 3: Múltiples transmisores

En estas pruebas, los nodos 9 y 10 transmitirán paquetes de diferentes prioridades de acuerdo con la tabla 3-3. Durante la transmisión se simulará el fallo del nodo 8 apagándolo 5 minutos después del envío del primer paquete. Esto permitirá medir el cambio en el retardo y la pérdida de paquetes de cada prioridad. Los paquetes del flujo 1 deben re-enrutarse por la ruta del flujo 2 sin intervención del controlador SDN. Además, se debe medir el tiempo que tarda el controlador en generar y enviar el nuevo esquema de enrutamiento. En este caso las medidas de retardo y pérdida de paquetes del nodo 10 se dividen en 3 instantes: Inicial (todos los nodos están operativos), Fallo (Nodo 8, apagado) y con el nuevo esquema de enrutamiento.

3.8. Análisis de resultados

3.8.1. Simulación

Test 1: Eficiencia del protocolo

El tiempo de convergencia de la red, depende principalmente de la sincronización TSCH. En la figura 3-9(a), se muestra el resultado para un tiempo de escaneo de 1 segundo sobre los 4 canales con redes de 1, 2 y 3 saltos. Como puede observarse, la convergencia con SDN presenta un retardo (líneas continuas), ya que el descubrimiento de la topología se inicia después de la sincronización TSCH para cada nodo. El aumento de este tiempo no es significativo, de acuerdo a la figura 3-9(b) la red tarda 32 segundos más en estar lista para el envío del primer paquete.

Por esto, el tiempo de convergencia total (T) para SDN-WISE TSCH viene dado por el tiempo medio de sincronización de TSCH (T_{S-TSCH}) y el tiempo de descubrimiento de la

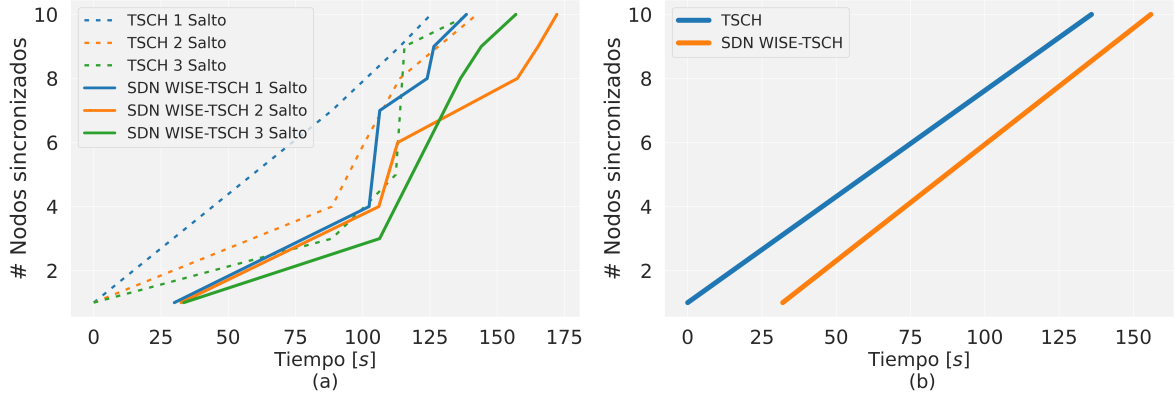


Figura 3-9: Tiempo de convergencia SDN con sincronización TSCH

topología (T_{TD}) según la ecuación 3-5. El tiempo medio de sincronización en TSCH viene determinado por la ecuación 3-5 según [70] el cual incrementa en función del número de saltos (NH).

$$T \cong T_{S-TSCH} + T_{TD} = T_s \cdot NH + T_{TD} \quad (3-5)$$

$$T_s = M_T \cdot \frac{(C + 1)}{2 \cdot PDR} = 2.5 \cdot M_T \quad (3-6)$$

Donde el M_T es el tiempo promedio para la transmisión de EB , en nuestro caso 16 segundos. Teniendo en cuenta que C es la cantidad de canales en la planificación TSCH, 4 en la planificación utilizada y el PDR obtenido es 1 (100%), se obtiene un tiempo de sincronización de 40 s aplicando estos valores en la ecuación 3-6.

Considerando que cada nodo envía la información al controlador por la ruta más corta y teniendo en cuenta el peor caso, los nodos más alejados, utilizarían rutas de 3 saltos para la topología planteada. De este modo, el tiempo de sincronización total en el caso que nos ocupa es igual a 120 segundos. Este tiempo es similar al obtenido en otros trabajos, como por ejemplo, el que se muestra en la figura 3-10 que corresponde a la figura 13 de [71], donde el tiempo medio de sincronización usando el algoritmo predeterminado del estándar 802.15.4E ($PS - Passive Scan$) para 3 saltos se sitúa alrededor de los 155 s. Además, según la ecuación 3-5 habría que considerar el T_{TD} , que en nuestro caso está alrededor de los 32 segundos, con ello finalmente $T \approx T_{S-TSCH} + T_{TD} \approx 120s + 32s = 152s$ aproximadamente, que son los 152 segundos de media obtenidos en las simulaciones.

Después de la fase de convergencia, se ha observado que SDN-WISE TSCH sigue intercambiando mensajes e informando al controlador de la topología de forma periódica. Sin

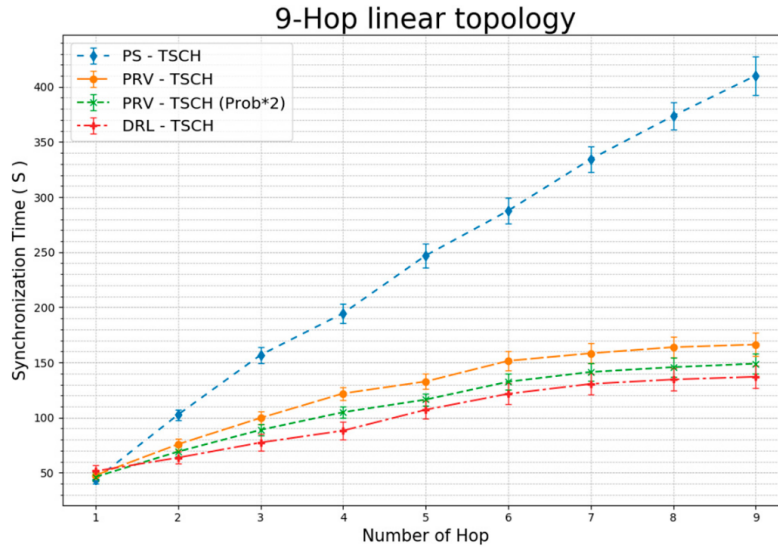


Figura 3-10: Tiempo medio de sincronización por salto en una topología lineal [70].

embargo, aquellos protocolos que hacen uso del protocolo RPL se estabilizan e intercambian un menor número de paquetes de control [11]. La cantidad de paquetes que genera SDN-WISE TSCH es significativamente superior a la de RPL, como se muestra en la figura **3-11(a)**. Sin embargo, parte de esta diferencia se compensa con el tamaño en bytes de los paquetes RPL, los cuales tienen el doble de bytes, como se muestra en la figura **3-11(b)** donde se muestra la cantidad total de bytes enviados por cada protocolo. Aunque la cantidad de bytes enviados sigue siendo menor para RPL, la mayor parte de estos bytes son transmitidos en la fase inicial, mientras que en SDN se realiza de forma constante, por lo tanto, aumentara el pico inicial en el RDC provocado por la sincronización TSCH de acuerdo a la figura **3-11(c)**. Posterior a esto, en la fase de estabilidad el RDC sigue siendo menor para SDN-WISE TSCH, debido a que el controlador no ha configurado ningún slot, mientras que los protocolos distribuidos habilitan de forma automática un determinado número de slots por vecino, aunque no sea necesario.

La capacidad de reacción que tiene el controlador para modificar el comportamiento de la WSN es el tiempo de configuración. Este está compuesto por dos elementos, el tiempo de llegada al nodo y el tiempo de procesamiento e instalación de la planificación, que se encarga de sincronizar la generación de paquetes. Este último elemento es totalmente dependiente de la planificación.

La figura **3-12** muestra el envío de dos OpenPathTSCH, uno en el timeslot 0 y el otro en el timeslot 590. Para el primer paquete de configuración, el nodo configura la frecuencia de envío a $1 \text{ paquete/slotframe}$. Para el segundo paquete, se modifica la planificación anterior y la frecuencia de envío pasa a $2 \text{ paquetes/slotframe}$. Debido a que el paquete OpenPathTSCH se transmite por los slots compartidos, el tiempo de llegada al nodo equivale al Shared E2E

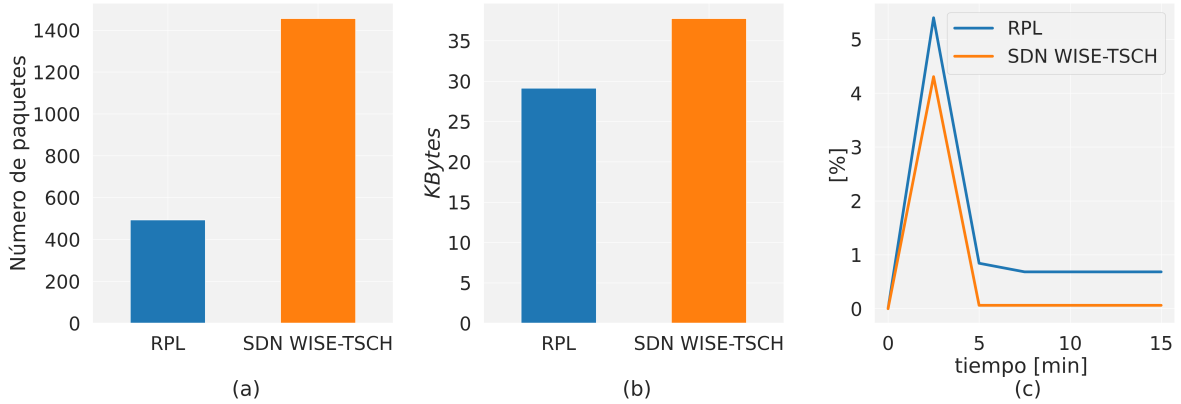


Figura 3-11: Tráfico de control generado y uso de la radio en los protocolos RPL y SDN-WISE TSCH

delay ($Shared_{E2E}$). Este tiempo depende de la longitud del slotframe (SF_{sz}), el número de saltos al nodo destino (NH), la cantidad de slots compartidos Sh_{SF} y el PDR. Para un PDR de 100% y la topología de la figura 3-4, donde el nodo 10 se encuentra a tres saltos del sink, la planificación tiene dos shared slots y el slotframe una longitud de 19 slots, se tiene un delay de 38 timeslots, según la ecuación 3-7. La nueva planificación se ejecuta a partir del segundo timeslot activo, ya que el primero se utiliza para sincronizar la generación de paquetes.

$$Shared_{E2E} = \frac{(NH - 1) \cdot SF_{sz}}{Sh_{SF} \cdot PDR} \quad (3-7)$$

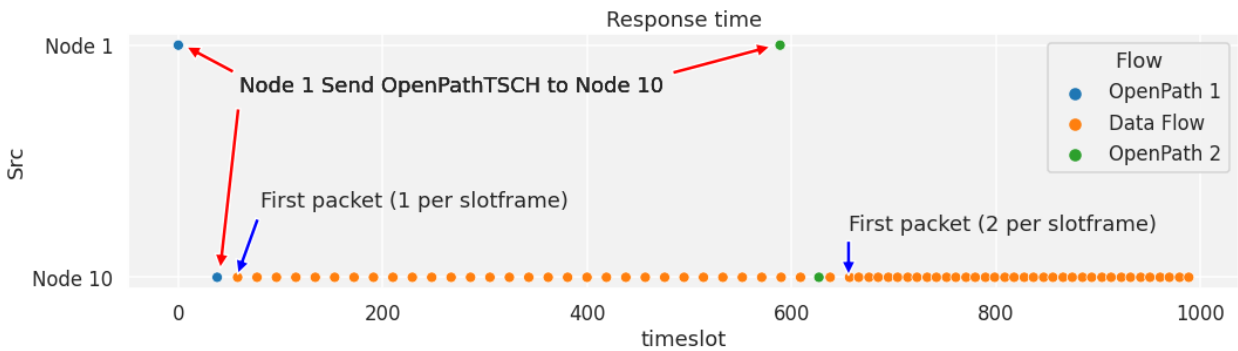


Figura 3-12: Tiempo de respuesta y cambio de planificación

Test 2: Retardo en la configuración

Para evaluar el rendimiento de SDN-WISE TSCH se realiza una transmisión durante 42.000 timeslots para cada uno de los flujos del nodo 10. Se realiza con cada uno de forma separada y se repite la prueba con todos los flujos transmitiendo. De este modo se puede evaluar cómo

se adapta el protocolo a las frecuencias de envío configuradas sin que interfieran los demás flujos. Se utilizarán los flujos del nodo 10 que corresponden a los deadlines de 100 ms, 70 ms y 200 ms. El algoritmo de planificación que usa SDN-WISE TSCH, permite asignar múltiples slots de transmisión para adaptarse a las necesidades de cada flujo. Como se ha obtenido en el apartado 3.5 de planificación, el tamaño óptimo del slotframe para este caso es de 19 slots, donde se asigna al nodo 10 un total de 6 slots de transmisión, de acuerdo a la planificación presentada en la figura 3-5. La configuración de tamaño del slotframe y número de slots de transmisión se realiza para que el PIAT esté siempre por debajo de los deadlines.

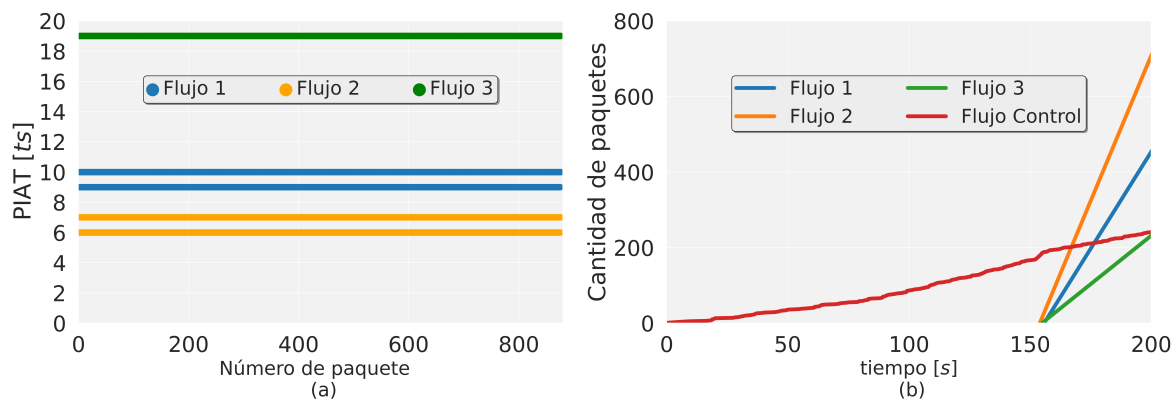


Figura 3-13: (a) PIAT para todos los flujos, (b) Total de paquetes enviados por flujo

El resultado se encuentra en la figura 3-13(a), el comportamiento es igual en el caso aislado y combinado. Los flujos se comportan de forma totalmente determinista, de acuerdo con la planificación obtenida en la figura 3-5. El flujo 1 tiene asignado dos repeticiones y están separadas con 10 slots dentro del mismo slotframe y con 9 slots entre slotframe. Por esto se observan dos líneas del tiempo entre llegadas de paquetes (PIAT). El mismo comportamiento muestra el flujo 2 con 6 y 7 slots, y solo el flujo 3 presenta un solo valor, ya que coincide con el tamaño del slotframe. La figura 3-13(b) muestra la cantidad de paquetes para cada uno de los flujos que hay en la red, el aumento en los flujos de datos es completamente lineal, no presentan ninguna interrupción. Esto se debe a que la generación de paquetes está sincronizada con el slotframe, por lo tanto, cada uno de los flujos tiene los recursos necesarios asignados, evitando que se produzcan pérdidas y garantizando un DSR del 100%. Para el caso del flujo de control, la parte inicial tiene un crecimiento lento, ya que solo se intercambian los *EB*, a medida que se sincronizan los nodos, hay un incremento más notorio del flujo, debido a que se introducen los paquetes de control de SDN-WISE TSCH.

SDN-WISE TSCH mantiene las características de transmisión y recepción en cada uno de los flujos sin importar si están aislados o combinados. Esto se debe a que, con las rutas, la planificación TSCH y el marcado de los paquetes, se ha creado un slicing que permite asignar y conservar los recursos de la red para cada uno de los flujos. De esta forma, no existe ningún

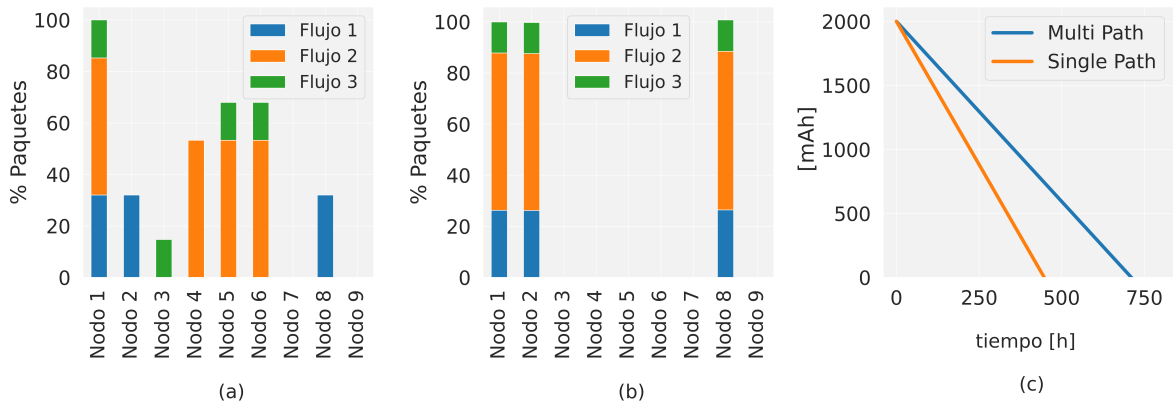


Figura 3-14: Distribución de los flujos sobre los nodos (a) SDN-WISE TSCH (b) única ruta para todos los nodos, (c) network lifetime para ambos casos.

tipo de interferencia en condiciones normales.

Como se ha observado hasta el momento, en esta implementación de SDN-WISE TSCH, se pueden garantizar todos los flujos de información, de forma que ni el exceso de tráfico de control ni los demás flujos alteren la planificación de la red, de esta forma, se obtiene un alto determinismo. La figura 3-14 muestra las ventajas de uso de múltiples rutas. SDN-WISE TSCH diferencia los flujos por prioridades, lo que permite asignar múltiples paths, distribuyendo el tráfico sobre una mayor cantidad de nodos, como se observa en la figura 3-14(a). En la figura 3-14(b) se muestra el uso de la misma ruta para todos los flujos. Esta distribución tiene un impacto directo sobre el rendimiento de la red y el tiempo de vida de la red, como se puede ver en la figura 3-14(c). El uso de una sola ruta genera un agotamiento prematuro de los nodos centrales. Para el caso del single path, el nodo 2 tiene el mayor consumo y es 33% superior al consumo del nodo 6, que es el de mayor consumo para el caso de multipath. Aunque, el consumo general del caso multipath es un 9% superior, debido al mayor número de saltos, este consumo no se centra en un solo nodo, se distribuye sobre una mayor cantidad de nodos, lo que permite aumentar el tiempo de vida de la red.

Una de las ventajas de los protocolos distribuidos, es la capacidad de reconfigurarse de forma autónoma, la cual les permite adaptarse de forma automática a cambios o fallos de la red. Por el contrario, en una SDN tradicional se requiere que los nodos informen al controlador SDN de los cambios para que este realice una reconfiguración, lo que implica un retardo mientras se recalcula la topología y envía la configuración a los nodos. Estos retardos y el exceso de tráfico de control que se produce limitan el uso de enfoques centralizados, por esta razón en SD-WSN se deja la elección de algunas decisiones a los nodos. A través de un sistema de estados previamente configurados por el controlador SDN, se permite que los nodos puedan alterar las decisiones del controlador basándose en información local.

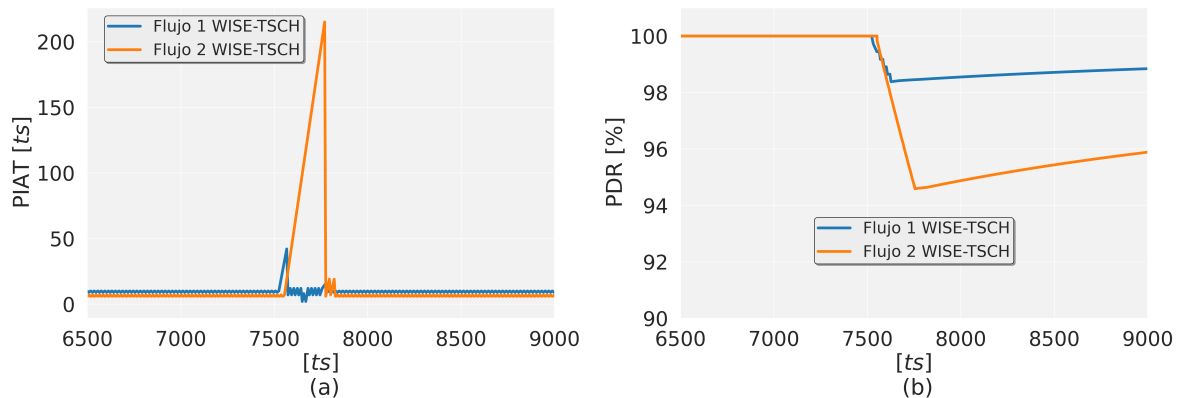


Figura 3-15: PIAT y PDR con cambio de ruta después del fallo del nodo 8

En este caso se explota esta posibilidad y el uso de múltiples rutas para conservar el PDR de los flujos prioritarios, desviando el tráfico de mayor prioridad cuando exista un fallo. En la figura **3-15(a)** se muestra el PIAT para el nodo 10 durante 7000 timeslots se mantiene constante y cumpliendo el deadline de 100 y 70 ms para los flujos 1 y 2. A los 7500 timeslots el nodo 8 falla, y se produce un incremento en el PIAT del flujo 1 hasta los 40 timeslots. Este tiempo es el que tarda el buffer en superar el límite de ocupación configurado, en este caso 3 paquetes. Con el periodo de generación actual del flujo 1 (10 timeslots), el límite se alcanza después de 30 timeslots. El paquete generado después de alcanzar el límite es enrutado por el path asignado al flujo 2. En este momento el PDR se estabiliza para el flujo 1, como puede observarse en la figura **3-15(b)**, a cambio de perder paquetes del flujo 2 de menor prioridad. El pico que se produce en el flujo 2, es debido al tiempo de reconfiguración del controlador. Después de estos 200 timeslots (2 s) el controlador ha enviado una nueva planificación que cumple con los requisitos establecidos para cada flujo.

El uso del sistema de estados, además de reducir la pérdida de paquetes cambiando de rutas, permite informar al controlador sobre el fallo del nodo 8 de inmediato. Por lo tanto, se evita el tiempo que tarda el protocolo de descubrimiento de la topología en identificar el fallo, acortando el tiempo de reconfiguración a niveles comparables de protocolos distribuidos, como se observa en la figura **3-16** donde se ha hecho una comparativa con Orchestra. Cuando se produce un fallo, timeslot 4400, el PIAT crece de forma rápida hasta los 250 timeslots para el protocolo distribuido, como se muestra en la figura **3-16(a)**. Sin embargo, para SDN-WISE sube solo hasta 40 timeslots, mientras se balancea el tráfico del flujo 1 y 2 por la segunda ruta. Con esto la pérdida de paquetes es atenuada por el uso de esta ruta, de acuerdo a la figura **3-16(b)** mientras que, para el protocolo distribuido cae hasta que se alcanza el tiempo de reconfiguración. Para el caso de SDN-WISE este tiempo puede observarse en el cambio del E2E delay del Flujo 1, como se observa en la figura **3-16(c)**, cuando hay un fallo el flujo pasa por el path del flujo 2, debido a esto, su E2E delay se incrementa de forma temporal

al nivel definido para la ruta 2, en este caso 3 timeslots. Cuando el controlador reconfigura la red, el flujo regresa al E2E delay de la ruta de prioridad 1.

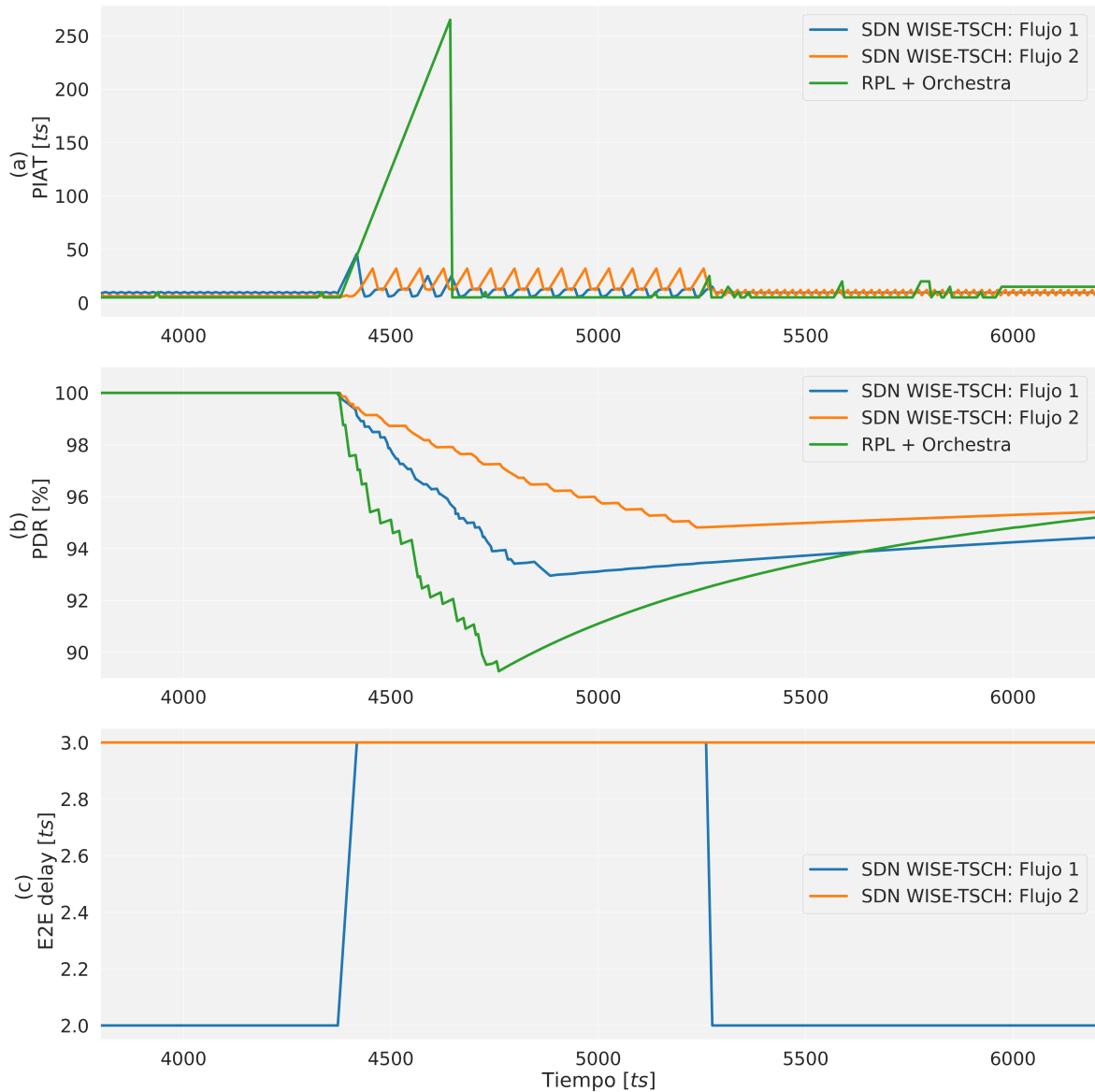


Figura 3-16: Comparación entre SDN-WISE TSCH y planificación distribuida

Test 3: Múltiples transmisores

El último parámetro por evaluar es el comportamiento de la red con tráfico adicional proveniente de otros nodos. En este caso dos flujos del nodo 9. La figura **3-17(a)** muestra el comportamiento del nodo 10, con un comportamiento constante hasta los 7600 timeslots, donde se produce el fallo del nodo 8. El comportamiento es igual que en el test anterior. El flujo 1 se reenvía por la ruta 2 y hay una pérdida de paquetes, de acuerdo a la figura **3-17(c)**, para el flujo 1 hasta que cambia de ruta y para el flujo 2 hasta que el controlador reconfigura. Todo este proceso genera un tráfico de control adicional que debe ser controlado para que no afecte a otros flujos. En el caso del nodo 9, el PIAT para el flujo 1 del nodo 9 oscila entre 7 y 12 timeslots, como se muestra en la figura **3-17(b)**, debido a que el algoritmo ha asignado dos flujos, uno en el slot 3 y otro en el slot 15. Esta configuración permite alcanzar los requisitos de deadline, y presenta un comportamiento cíclico durante toda la prueba, incluso en los instantes donde falla el nodo 8 y el nodo 10 solicita una reconfiguración que incrementa el tráfico de control. No se aprecian cambios ni alteraciones en ninguno de los flujos. Tras estas pruebas se comprueba que bajo condiciones normales y alteraciones externas al nodo se puede garantizar un *slice* para cada flujo y no existe ningún tipo de interferencia entre ellos, a menos que se produzca un fallo en la red, donde el nodo afectado modificará el envío de los paquetes para minimizar el impacto del fallo en el flujo de mayor prioridad. Los parámetros medidos en la figura **3-17(d)**, muestran cómo no existe ninguna alteración en los flujos del nodo 9, el PDR se mantiene en 100 % y no hay ninguna fluctuación no prevista en el PIAT. En las pruebas que se muestran en la figura **3-17**, se asignó al flujo 2 del nodo 9 una repetición del slotframe, lo que representa un PIAT de 190 ms. Sin embargo, el deadline requerido para este flujo es de 300 ms y cumplirlo con un margen tan amplio implica un exceso de transmisiones que se pueden evitar. Por esto, el algoritmo de planificación elige el tamaño del slotframe de acuerdo con el flujo con mayor deadline configurado en el descriptor de tráfico.

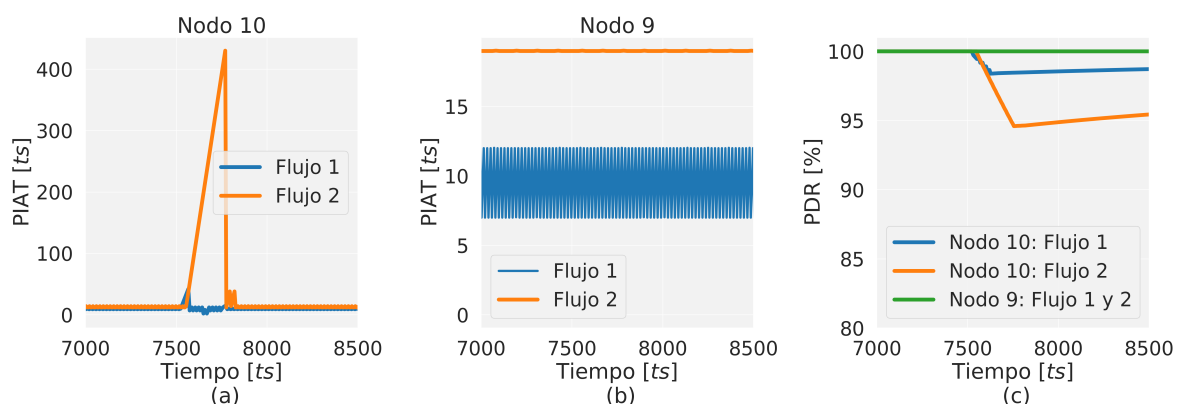


Figura 3-17: Efecto de múltiples transmisores simultáneos en la topología de prueba

3.8.2. Testbed

Para comprobar la implementación de forma real, se han utilizado 10 Openmote-B. La potencia de transmisión de los nodos ha sido limitada a -15 dB para lograr la red multi-salto que corresponde a la figura **3-18**. Además, se disminuye el número de canales a 2 para reducir la cantidad de nodos de medición (Sniffers). Esta reducción de canales no representa una pérdida de rendimiento para las pruebas realizadas. El sink está compuesto por la combinación de un Openmote-B y una ODROID N2, que establece la conexión con el controlador a través de Ethernet.

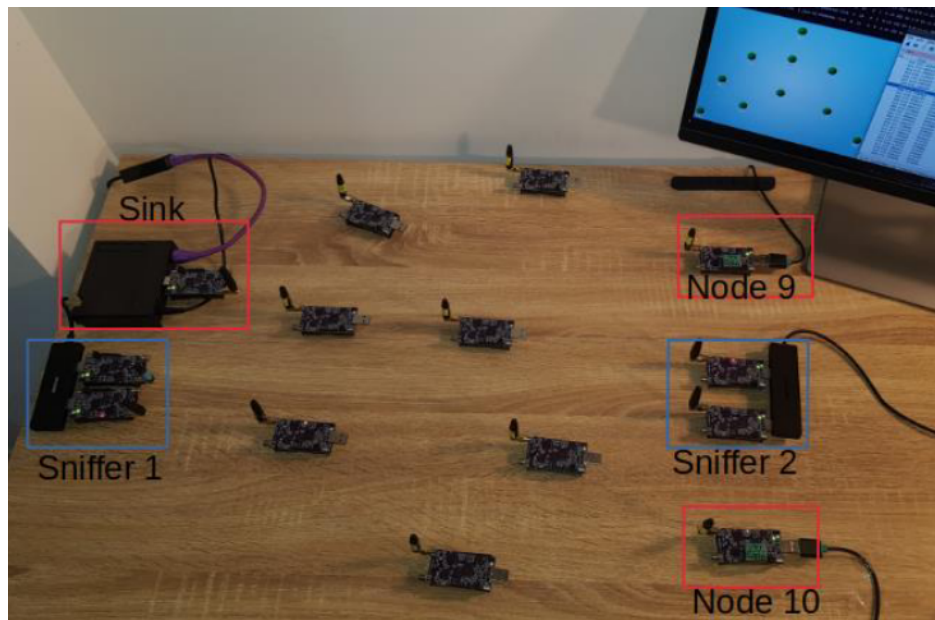


Figura 3-18: Elementos del testbed para topología de 10 nodos

En la figura **3-18** se muestran los elementos reales que componen la red, 10 nodos incluyendo el sink, y además 4 OpenMote-B adicionales configurados como sniffers que permiten capturar el tráfico de los nodos 9, 10 y del sink. El tráfico capturado es el utilizado para medir el rendimiento obtenido. Ya que estos no pueden sincronizarse con el TSCH, capturan el tráfico sobre un único canal. Por esta razón se ha reducido el número de canales y creado sniffers de 2 nodos, que permiten capturar todo el tráfico planificado.

Las pruebas realizadas muestran un comportamiento coherente con el apartado de simulación, donde se mantiene el determinismo sobre una red estable. En la figura **3-19** se observa como los flujos mantienen un comportamiento estable hasta los 5 minutos, donde el nodo 8 tiene configurado apagar la radio. En este instante se genera un pico en el PIAT del flujo 1 de 40 timeslots, como se observa en la figura **3-19(a)**, exactamente igual que en la simulación, ya que es un valor que depende solo de la ocupación local del buffer.

Cuando se alcanzan estos 40 timeslots, el nodo intercambia las rutas de los flujos, pasando

la pérdida de paquetes al flujo 2 y atenuando las del flujo 1, obteniendo los resultados del PDR de la figura 3-19(b). Mientras las rutas estén intercambiadas, el E2E delay del flujo 1 será el que tuviera la ruta 2, en este caso 3 timeslots, como se muestra en la figura 3-19(c). Este comportamiento se extiende hasta que el controlador restablece las rutas con una nueva planificación. Este tiempo de reconfiguración se corresponde con el pico del PIAT del flujo 2, 480 timeslots. Esto representa un incremento del 20 % respecto al máximo valor obtenido en la simulación. Este incremento se debe al uso de los slots compartidos, donde existen colisiones que se ven incrementadas en la topología real. Al igual que en el test 3 de las simulaciones, ninguna de las métricas del nodo 9 se ve afectada, por los flujos del nodo 10 o el fallo del nodo 8.

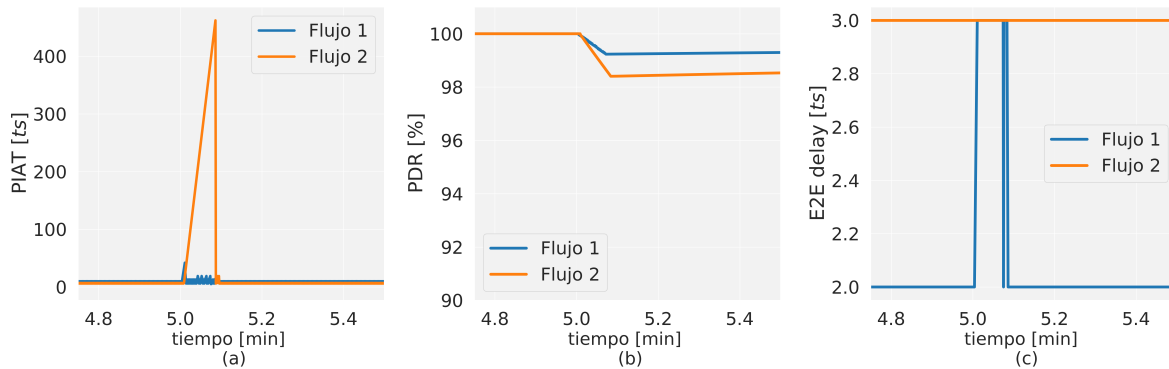


Figura 3-19: Tolerancia a fallos de SDN WISE-TSCH en tesbed

3.9. Conclusiones

En este capítulo se ha abordado la integración del paradigma SDN y las redes de sensores inalámbricos (WSN), como una alternativa que permite a este tipo de redes adaptarse a los requisitos de los entornos industriales. Para esto, se realizó un estudio de la arquitectura SDN WISE y sus componentes, a través del cual se analizaron las limitaciones presentes en las redes de sensores inalámbricos tradicionales como lo son el ancho de banda, la capacidad de reconfiguración, consumo de energía y capacidad de procesamiento.

La integración con SDN permite tener un control preciso y amplio sobre el comportamiento de la red de forma totalmente centralizada, donde el controlador SDN consigue una visión global de la red a cambio de aumentar el tráfico de control generado por la red. Para explotar estas ventajas, se ha desarrollado el algoritmo de enrutamiento, el planificador TSCH y el administrador de tráfico, los cuales se han enfocado en aumentar las capacidades de calidad de servicio de este tipo de redes.

En el caso del algoritmo de planificación se utiliza una métrica multi-parámetro y acumulada, la cual permite conocer la congestión que tendrá cada nodo después de cada asignación

de recursos del controlador. De esta forma, las rutas que obtiene el algoritmo varían para distribuir el tráfico de una forma más homogénea sobre los diferentes nodos de la red, lo que reduce la congestión y el consumo de energía en un único punto de la red. Las rutas obtenidas por este algoritmo, son utilizadas como entrada en el planificador TSCH que asigna las celdas de transmisión y recepción de cada salto de forma consecutiva para reducir el E2E delay. Además, utiliza el parámetro deadline del administrador de tráfico para realizar una asignación de recursos iterativa. Esto permite modelar la cantidad de recursos asignada a cada flujo de forma independiente.

Finalmente, la combinación de los elementos anteriores y la asignación de recursos desde el controlador SDN permiten que cada nodo diferencie los recursos pertenecientes a cada flujo, consiguiendo un slicing completo entre todo el tráfico que circula por la IWSN. Como resultado, se ha conseguido una red altamente determinista, donde es posible garantizar las métricas de calidad de servicio por flujo sin que existan interferencias por la cantidad de flujos en la red. Sin embargo, no se ha realizado una comparación completa del rendimiento obtenido respecto al estado del arte actual, como la implementación RPL + Orchestra, u otros métodos de planificación. Este análisis y comparativa se realiza en el capítulo siguiente.

Capítulo 4

Planificación de acceso al medio determinista

4.1. Introducción

La adopción de las IWSN a nivel industrial requiere que los dispositivos y protocolos, puedan garantizar el correcto funcionamiento físico y un mayor determinismo en las transmisiones. Por esta razón, el estándar IEEE 802.15.4e ha incorporado mecanismos de acceso al medio que les permiten implementar estas funcionalidades, como se detalla en la sección 2.2.3. Sin embargo, el estándar se limita a proporcionar herramientas para crear un acceso al medio libre de contienda, sin especificar cómo realizar la asignación de recursos, dando lugar a múltiples interpretaciones y alternativas para abordarlo. Es por ello que en este capítulo se realiza un estudio y análisis del rendimiento de la planificación de acceso al medio, centrándose específicamente en su capacidad para garantizar parámetros de calidad de servicio. El objetivo es observar las capacidades y limitaciones de la integración de las redes IWSN con SDN desarrollada en el Capítulo 3, en comparación con el estado del arte y los protocolos de uso más extendido.

De acuerdo con la sección 2.2.3, los protocolos con mayor capacidad de adaptación a los sectores industriales son DSME y TSCH, debido a que incorporan mecanismos para limitar la interferencia y aumentar el determinismo. Estos protocolos se basan en la división temporal para asignar timeslots de transmisión sobre diferentes frecuencias. Sin embargo, el protocolo TSCH tiene un uso más extendido debido a que el slotframe que utiliza es más simple y tiene una mayor flexibilidad. Como en ambos protocolos las transmisiones se realizan en un instante acotado y determinado, el transmisor y receptor deben estar sincronizados e informados de la acción a realizar en cada instante de tiempo. Por lo tanto, es necesario un proceso de planificación que asigne los timeslots de transmisión y recepción, así como los canales en que se realizarán. Las diferentes propuestas para generar la planificación TSCH

se clasifican en: centralizadas, distribuidas o autónomas, según la entidad o dispositivo que tome la decisión para asignar los slots.

El estado de arte actual ha convertido al planificador autónomo Orchestra [34], como una solución *de facto* cuando se despliegan redes de sensores basadas en TSCH, ya que, se ha demostrado como consigue un alto rendimiento y baja complejidad de despliegue. Sin embargo, el aumento constante de los requisitos de calidad de servicio, exige garantizar otros parámetros, como E2E delay y deadline, para que las IWSN se adapten a la convergencia de flujos de diferentes criticidades. Por ello, este capítulo se enfoca en analizar las ventajas de las redes definidas por software para garantizar parámetros estrictos de calidad de servicio en entornos multiflujo, utilizando la implementación SDN WISE-TSCH desarrollada en el capítulo 3. Las pruebas consisten en la simulación de diferentes escenarios con características de nivel industrial usando diferentes planificadores TSCH que se encuentran disponibles y que han sido ampliamente estudiados [72] [73] [74]. Se evalúa el rendimiento en términos de E2E delay, packet delivery ratio, deadline satisfaction ratio y power consumption. Además, se realiza la implementación de un testbed donde se comparan los dos protocolos que han demostrado un mejor comportamiento en las simulaciones, evaluándose el E2E delay y el Packet Inter-Arrival Time (PIAT).

4.2. Planificadores TSCH

En la sección 2.2.3 se profundizó en el funcionamiento del protocolo TSCH como método de acceso al medio. Este realiza una división en tiempo y frecuencia, dando lugar a una matriz o slotframe donde se distribuyen en celdas las acciones que deben realizar los nodos. Para asignar estos estados de operación de cada nodo a las celdas correspondientes, es necesario el planificador TSCH. Este decide qué acción realizará el nodo en cada timeslot, donde podrá recibir, transmitir o dormir. Los nodos dentro de la planificación deben estar completamente sincronizados para que los instantes de transmisión y recepción coincidan. En esta sección se describen brevemente las características principales de los tipos de planificadores TSCH existentes, así como el funcionamiento específico de algunos de ellos que se utilizarán en un análisis comparativo.

Las diferencias entre cada uno de los enfoques pueden verse en la figura 4-1, utilizando la misma topología. En el planificador autónomo de la figura 4-1(a), los nodos solo intercambian mensajes RPL a través de enlaces inalámbricos establecidos, y cada nodo reutiliza la información RPL obtenida para realizar la planificación. En los planificadores distribuidos, además de la información RPL, los nodos intercambian paquetes de negociación para solicitar o aceptar celdas en la planificación TSCH, como se muestra en la figura 4-1(b). En las figuras 4-1(c), (d) se muestra el enfoque centralizado donde se utiliza un único dispositivo para crear la planificación, por lo que cada nodo debe enviar tráfico de control con información sobre la topología. Sin embargo, una centralización simple como en (d), mantiene el

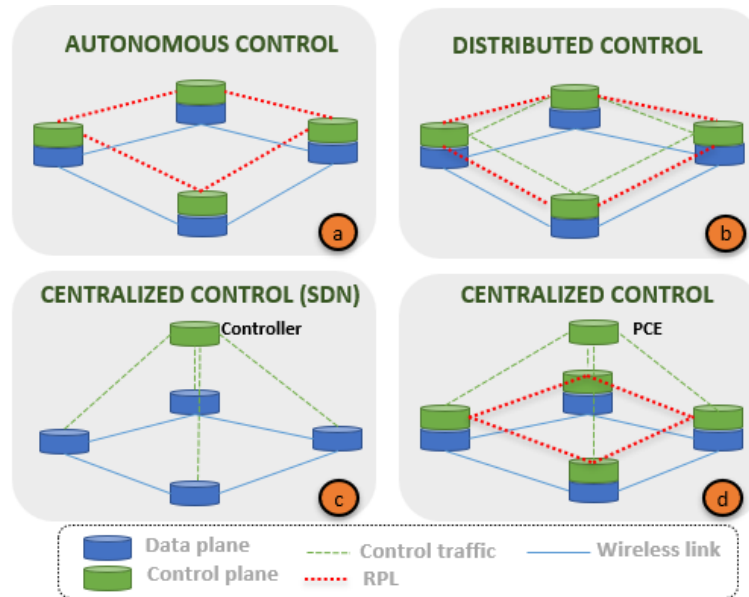


Figura 4-1: Distribución del plano de control para los diferentes tipos de planificadores TSCH (a) Autónomos (b) Distribuidos (c) Centralizados (SDN) (d) Centralizados

funcionamiento de los dispositivos y protocolos distribuidos como en la figura 4-1(a), (b). A diferencia de la figura 4-1(c), donde los dispositivos y protocolos están completamente diseñados para operar con este enfoque, son más sencillos, ya que no tienen que tomar ninguna decisión porque el plano de control está completamente centralizado. En esta sección se describen brevemente las principales características de los planificadores TSCH existentes, así como las prestaciones específicas de algunos de ellos que se utilizarán en un análisis comparativo.

4.2.1. Autónomos

Los planificadores TSCH autónomos se caracterizan por reutilizar la información que tienen almacenada los nodos sobre la topología y el enrutamiento, para la realización de la planificación. Esta información la han obtenido a través del intercambio de mensajes de control del RPL con sus vecinos, como se muestra en la figura 4-1(a). Por lo tanto, no necesitan de señalización adicional para la conformación de las distintas planificaciones y solo es necesaria la dirección MAC e información de la topología actualizada. El planificador autónomo más utilizado es Orchestra [34], debido a su fácil configuración y alto rendimiento.

En Orchestra cada nodo crea y mantiene su propia planificación TSCH y la va actualizando a medida que se suceden cambios en la red. De hecho, solo utilizan la información RPL recibida de los vecinos y de los padres. Una planificación Orchestra se compone de varios slotframes, cada uno de ellos para un tipo de tráfico distinto: mensajes Beacon (broadcast), mensajes RPL o mensajes unicast (tráfico de aplicación).

En Orchestra se definen cuatro tipos de timeslots:

- *Common Shared slot* (CS): una celda común en todos los nodos que utilizan para transmisión y/o recepción.
- *Receiver-based slot* (RBS): Se planifica una celda en recepción y una celda en transmisión para cada nodo vecino, la posición de la celda se calcula con la información del nodo receptor.
- *Sender-based slot* (SBS): Se planifica una celda en transmisión para cada nodo y uno en recepción para los vecinos. La diferencia con el anterior es que la posición de la celda se calcula con la información del nodo transmisor.
- *Sender-based Dedicated Orchestra slots* (SBD): Es una celda de transmisión específica para cada vecino, pero en este caso en lugar de utilizar una celda compartida, se utiliza una celda dedicada.

Por otro lado, los diferentes slotframes que pueden componer la planificación con Orchestra son:

- *EB Slotframe*: es un slotframe formado por slots dedicados basados en el nodo transmisor, y es para la transmisión de mensajes EB.
- *Broadcast Slotframe*: es un slotframe formado por las celdas compartidas, principalmente para tráfico RPL.
- *Receiver-based Unicast Slotframe*: para tráfico unicast consistente en celdas basadas en el nodo receptor.
- *Sender-based Unicast Slotframe*: para tráfico unicast consistente en celdas basadas en el nodo transmisor.

La planificación en Orchestra puede llevarse a cabo en dos configuraciones distintas: basada en el receptor o basada en el transmisor. La diferencia principal entre ambas configuraciones radica en que, en la basada en el receptor, es este último quien asume la responsabilidad de iniciar la transmisión de datos. Por otro lado, en la configuración basada en el transmisor, es el propio transmisor quien toma la iniciativa de iniciar la transmisión de datos.

4.2.2. Distribuidos

Los planificadores distribuidos se basan en la negociación de la asignación de celdas entre nodos adyacentes o vecinos. En términos de fiabilidad y retardos, ofrecen un rendimiento similar a los planificadores autónomos, ya que también se basan en el protocolo RPL. Sin embargo, tienen un coste añadido por la señalización y mensajes de control utilizados en la negociación, como se ve en la figura 4-1(b). En [75], se examinan tres posibles enfoques para la asignación de celdas entre nodos adyacentes. La primera opción implica compartir

una misma celda entre todos los nodos, reduciendo así la complejidad de la señalización y las planificaciones, aunque aumenta la probabilidad de interferencias y colisiones. La segunda alternativa consiste en la negociación de un timeslot, lo que da como resultado un incremento del consumo de energía debido al intercambio de mensajes. La última opción implica la elección de un timeslot arbitrario, requiriendo el uso de funciones hash, las cuales utilizan como entradas, parámetros fijos de cada nodo, como el ID o la MAC para calcular las coordenadas de la celda asignada entre dos nodos adyacentes.

Por ello, la estrategia de asignación de recursos se elegirá en función de los requisitos de la aplicación. Para una aplicación con requisitos de fiabilidad y alta carga de tráfico, se elegirá la opción de señalización adicional. Por el contrario, si los requisitos se centran en la temporización, se recomienda el uso de las funciones hash. *DeTAS* [32], *Wave* [33] y *6TiSCH MSF* [76] son ejemplos de planificadores distribuidos. No obstante, también se han propuesto algoritmos híbridos que combinan ambas estrategias. Uno de estos algoritmos es DIS-TSCH [77].

MSF define cómo se gestiona la planificación de las comunicaciones de forma distribuida. Este, está construido sobre el esquema de planificación más simple, la planificación con TSCH minimal. En este caso, la planificación está almacenada en el nodo, por lo tanto, no existen procesos para la asignación de celdas. De forma predeterminada, MSF se encuentra integrado en el sistema operativo Contiki-NG, junto con Orchestra. Contiene un solo slotframe con una longitud pre-configurada donde se planifica una única celda para todos los nodos. De esta forma, cualquier tipo de tráfico es enviado usando esta celda (EB, Datos, RPL) [78]. Como todos los nodos transmiten y reciben en el mismo timeslot y en el mismo channel Offset es un slot compartido, donde es muy probable que se presenten colisiones. Este funcionamiento con un solo slot obtiene un consumo de energía muy reducido a cambio de una mayor cantidad de paquetes en cola debido a las retransmisiones. Para mejorar el rendimiento, MSF incluye una serie de funciones que le permiten negociar celdas.

Una vez que un nodo se ha unido y sincronizado correctamente a la red, puede añadir/eliminar o reubicar celdas si existe alguna de las siguientes razones.

- Adaptar los recursos de la capa de enlace al tráfico entre el nodo y el padre.
- Modificar la conmutación del padre.
- Gestionar una colisión en las planificaciones.

4.2.3. Centralizados

A diferencia de las soluciones anteriores donde los nodos construyen su topología, en los centralizados un elemento externo denominado de forma general como elemento de cálculo de ruta (PCE - *Path computation Element*) realiza el proceso de asignación de celdas de toda

la red y envía esta planificación a los nodos. Para esto, el PCE recolecta información de la topología, a partir de mensajes específicos creados por cada nodo. Estos mensajes contienen los nodos adyacentes y estadísticas de las transmisiones. Esta centralización permite construir planificaciones con un alto nivel de optimización, haciéndolos ideales para aplicaciones *real-time* [79], debido a que pueden garantizar bajas latencias. Este es el caso de planificadores como TASA [11], AMUS [12] y SDN WISE-TSCH [61]. El problema principal de este enfoque es el tráfico que se genera entre el PCE y los nodos, que tiene un impacto significativo en el ancho de banda de la red.

AMUS [12] es un protocolo para planificación TSCH que aporta fiabilidad en cuanto a lograr baja latencia para aplicaciones con requisitos de temporización bajos, como son la mayoría de aplicaciones y procesos industriales. Está basado en TASA [11] para solucionar sus limitaciones, en comparación AMUS obtiene mejor rendimiento en términos de fiabilidad y delays. Aunque la implementación de AMUS, no se encuentra disponible, en [12] se encuentra una explicación detallada que ha permitido implementarlo para llevar a cabo la comparativa.

Este método consiste en la reserva de las distintas celdas o recursos de la planificación a lo largo de las distintas rutas que siguen los paquetes en una red. En este sentido, el algoritmo propone conseguir ese objetivo mediante un proceso que consta de dos fases:

(a) Recolección de información de la red. Mediante protocolos de enrutamiento y control, como el RPL, se recoge la información de estadísticas de la red para saber parámetros como el número de paquetes que genera un nodo en un periodo, topologías, etc.

(b) Asignación de celdas. Para una correcta planificación sin colisiones y con garantías de temporización, es necesario conocer previamente las secuencias de envíos que se van a llevar a cabo. Debido a que los nodos más cercanos al sink deben encaminar los paquetes de nodos que están más alejados, la celda que se encarga de transmitir los paquetes al sink debe situarse en un timeslot posterior a la recepción del paquete a transmitir. El proceso a seguir es el siguiente:

- Conocimiento de las rutas que seguirán los paquetes de cada flujo de tráfico hasta el nodo sink.
- Construcción de la matriz *Scheduling Sequence* (SS): se trata de una matriz de $2 \cdot n$ (n = número de nodos que transmiten paquetes), donde la primera fila indica el ID del nodo transmisor y la segunda la tasa de tráfico de cada nodo. Cada columna de la matriz SS representa un flujo de tráfico.
- Construcción de la matriz *Multi-hop Scheduling Sequence* (MSS): se trata de una matriz compuesta por tres filas ($3 \cdot n$), siendo la primera el nodo origen, la segunda el nodo destino y la tercera la tasa de tráfico. Cada columna de la matriz representa un enlace entre dos nodos adyacentes.

- *Scheduling*: una vez obtenida la matriz MSS, se asignan las celdas por orden de columna, empezando por el primer timeslot libre y el primer Channel Offset. Se asignarán tantas celdas como indique la tasa de tráfico (tercera fila).

Las celdas deben asignarse de acuerdo con las siguientes restricciones, para evitar colisiones o interferencias:

- Los nodos sólo pueden tener un estado por timeslot (transmisión o recepción).
- Las transmisiones de otros enlaces que puedan causar colisiones o interferencias deben realizarse en otro canal del mismo timeslot.
- Los enlaces que no tengan ningún conflicto o interferencia pueden asignarse en paralelo. En este caso se puede utilizar el mismo timeslot.

En el capítulo 3 se ha desarrollado SDN WISE-TSCH, un sistema altamente determinista que permite garantizar los requisitos de nivel industrial. Tiene una pila de protocolos que permite tener el control centralizado de todos los procesos en los nodos (MAC, Enrutamiento, QoS, Aplicación). Para esto, se basa en el paradigma de SDN que consiste en gestionar las redes separando el plano de control y el plano de datos [6].

Como en todos los planificadores centralizados, en SDN WISE-TSCH [61] se requiere un intercambio de mensajes entre los nodos y el controlador. Sin embargo, como toda la pila de protocolos está diseñado para SDN, el descubrimiento de la topología está optimizado, por lo que no requiere el uso del protocolo RPL como se aprecia en la figura 4-1(c). Por esta razón, el tráfico de control es menor que el usado por AMUS, que además de la comunicación con el controlador, requiere el uso de RPL tal y como muestra la figura 4-1(d). La topología es administrada a través de las siguientes aplicaciones:

- Administrador de tráfico: Contiene las especificaciones de los diferentes flujos de tráfico, que están clasificados por prioridad, deadline, nodo origen y destino.
- Proceso de enrutamiento: Se encarga de crear las rutas de origen y destino para cada flujo. Utiliza como métricas el RSSI y el nivel de congestión de cada nodo que se actualiza cada vez que una ruta es asignada.
- Planificador TSCH: Para cada ruta asigna los recursos TSCH correspondientes, generando planificaciones que garantizan un E2E delay mínimo y el deadline de cada flujo.

La integración de estas 3 aplicaciones permite realizar un slicing de la red, esto consiste en la diferenciación, aislamiento y segmentación lógica de los flujos de datos, por medio de la asignación de recursos TSCH. Este slicing permite garantizar parámetros de QoS para cada flujo de datos de forma independiente.

El algoritmo de planificación está compuesto por múltiples matrices, que se encargan de

almacenar la información de las celdas asignadas. En este caso hay dos matrices con dimensiones iguales a las del slotframe, una que almacena todos los nodos en transmisión y otra para los nodos en recepción. Utiliza las mismas restricciones que AMUS para evitar colisiones e interferencias. Además, los saltos son asignados de forma consecutiva, para obtener el menor delay también utiliza el deadline para adaptar los flujos a la longitud del slotframe. Para esto, se utilizan repeticiones, esto consiste en asignar los recursos de la ruta en diferentes posiciones del slotframe. Así se consigue garantizar el ancho de banda específico para cada flujo y disminuir las latencias de forma independiente a la longitud del slotframe. Los pasos que realiza el algoritmo son los siguientes:

- Obtener la tabla de flujos del Administrador de tráfico organizada por prioridad y deadline.
- Obtener las rutas calculadas por el proceso de enrutamiento para cada una de las entradas de la tabla de flujo.
- Para cada una de las rutas, realiza una descomposición en saltos.
- Calcula la cantidad de repeticiones que debe instalar sobre esta ruta, de acuerdo a la longitud del slotframe/deadline.
- La asignación de celdas se realiza para cada salto, donde se busca un timeslot sin conflictos, esto es, buscar en las dos matrices una columna donde el nodo transmisor y receptor no tengan asignados estados.
- Desplaza la siguiente repetición de la ruta una cantidad de timeslots equivalente al deadline y repite el proceso de asignación de celdas.

Para aumentar la flexibilidad y reducir el número de paquetes de control, el envío de la planificación TSCH se realiza por flujos. Esto permite la transmisión de un único paquete desde el controlador para configurar una ruta completa desde el origen hasta el destino. Este paquete incluye la información de enrutamiento y TSCH. Además, debido al funcionamiento integrado, estos procesos contienen información sobre el tipo de tráfico que se enviará a lo largo de esta ruta, lo que permite la asignación de recursos dedicados para cada tipo de tráfico. Las principales características de cada planificador TSCH se resumen en la tabla 4-1.

	SDN WISE-TSCH	AMUS	MSF	ORCHESTRA
Construcción de topología	SDN WISE: Beacon and Report	RPL	RPL	RPL
Tipo de planificación	Centralizado	Centralizado	Distribuido	Autónomo
Reserva de recursos	Dinámica y específica según deadline y tasa de datos	Específica según tasa de datos	Una sola celda compartida con celdas negociadas	Una sola celda unicast
Respuesta a cambios	Sí, configuración dinámica y balance de carga	No, pre-configurado	Sí (mensajes RPL)	Sí (mensajes RPL)
Prevención de colisiones	Sí	Sí	No	Sí
Clasificación de tráfico	Sí, clasificación y marcado del todo el tráfico, aplicación de reglas específicas para cada tipo.	No	No	Sólo: Broadcast (mensaje RPL), EB (Beacon), Unicast

Tabla 4-1: Resumen de las características de cada uno de los planificadores TSCH.

4.3. Escenarios y métricas

Se ha utilizado el simulador Cooja, disponible como parte del sistema operativo Contiki-NG, para analizar el rendimiento de los planificadores descritos. El simulador Cooja ofrece un modelo de propagación de radio realista que se basa en el modelo *Multi-Path Ray-tracer Medium* (MRM). Este modelo tiene en cuenta los efectos de la propagación, como la dispersión y la interferencia. En concreto, se utilizan los parámetros MRM $n = 2.04$ y $\sigma = 6.7$ que de acuerdo a [80] son adecuados para representar un entorno industrial. El hardware utilizado para la simulación es un intel core i7 con 16 GB de RAM y los nodos están modelados como nodos OpenMote-B. La carga útil utilizada es de 110 bytes, ya que el tamaño máximo de paquete del estándar 802.15.4 es de 127 bytes, de los cuales 17 se utilizan para la cabecera MAC.

Se ha utilizado una red con topología de malla típica en el sector industrial, ya que ofrece mayor fiabilidad al poder encaminar los datos por múltiples rutas [81]. La topología se muestra en la figura 4-2. Está compuesta por 1 nodo sink, 6 nodos repetidores y 3 nodos (8, 9, 10) que introducirán flujos de tráfico a la red con diferentes requisitos de calidad de servicio.

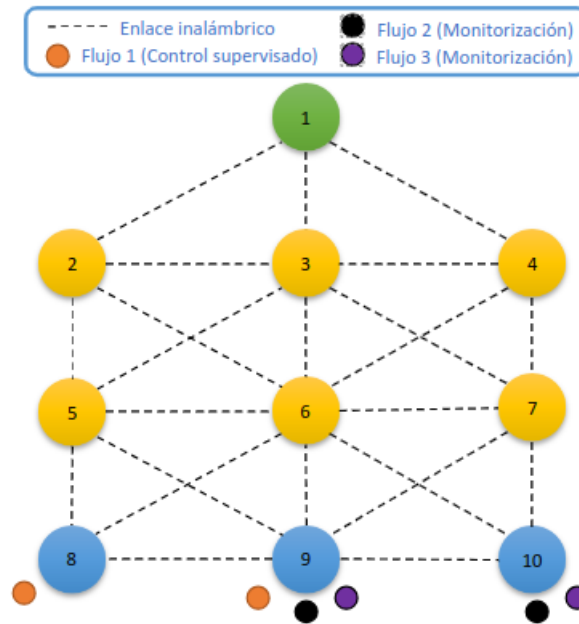


Figura 4-2: Topología con 10 nodos fijos.

Para caracterizar los flujos se han utilizado los requisitos de fiabilidad y latencia, típicos de los sistemas industriales, que se encuentran definidos en IEC 62264 (ANSI/ISA 95). De acuerdo a esta norma, los flujos de datos utilizados pueden clasificarse como:

- Control supervisado basado en tareas (*Supervisory Control Job Based*): Latencia inferior a 100 ms, pérdida menor a 10^{-6} y una cantidad de dispositivos menor a 10.
- Monitorización (*Monitoring*): Latencia menor a 1 segundo, pérdida menor a 10^{-5} y una cantidad de dispositivos menor a 10000.

Como se muestra en la figura 4-2 los flujos tienen un periodo de paquete diferente: 70 ms (flujo 1, tipo: Control supervisado), 140 ms (flujo 2, tipo: Monitorización) y 500 ms (flujo 3, tipo: Monitorización) respectivamente.

En la simulación se han realizado dos tests. El test 1, consiste en observar el comportamiento de la red con una baja saturación, usando flujos de tipo monitorización y control supervisado de forma independiente. En este caso el nodo 9 envía cada uno de los flujos de forma independiente al nodo sink. Para cada uno de los periodos de paquetes se analizaron las siguientes métricas:

- *End-to-End delay* (E2E delay): milisegundos que transcurren desde que se genera un paquete en el nodo transmisor y llega al destino, se mide para cada uno de los paquetes recibidos.
- *Packet Delivery Ratio* (PDR): relación entre la cantidad de paquetes enviados y los

recibidos por el nodo destino.

- *Radio Duty Cycle* (RDC): porcentaje de tiempo que permanece la radio encendida en estado de transmisión o recepción.
- *Packet Inter-Arrival Time* (PIAT): es la cantidad de milisegundos que transcurren entre cada recepción de un mismo tipo de paquete en el nodo destino. Esta métrica permite evaluar cómo se adapta la red a los estrictos requisitos temporales.
- *Deadline Satisfaction Ratio* (DSR): esta métrica se obtiene a partir del PIAT, es el porcentaje acumulado de los paquetes que se mantienen por debajo del deadline.

El test 2, se ha desarrollado usando múltiples flujo de datos para evaluar el comportamiento en una red con más tráfico. Para ello se han utilizado conjuntamente flujos de tipo control de supervisión y monitorización. En este caso los nodos 8, 9 y 10 transmitieron todos los flujos asignados simultáneamente al nodo 1, que actúa como sink. A continuación, se han observado los tiempos de llegada de los paquetes. Esto nos permite conocer el comportamiento del nodo con flujos de diferentes prioridades típicos en el sector industrial.

En TSCH el ancho de banda de cada nodo, depende de la cantidad de timeslots de transmisión en cada slotframe, por lo tanto, un menor número de timeslots en el slotframe aumenta el ancho de banda. Dado que Orchestra y MSF envían por defecto un paquete por slotframe, se utilizará una longitud de 7 timeslots, ya que es la longitud mínima configurable para el número de nodos de la topología, sin que Orchestra sobreasigne las celdas. Este tamaño de slotframe permite un mayor ancho de banda para adaptarse al flujo con el periodo de paquete de 70 ms. Los parámetros utilizados en las simulaciones se resumen en la tabla 4-2.

Parámetro	Valor	Unidades
Radio Medium	MRM ($n = 2.04, \sigma = 6.7$)	-
Timeslot	10	ms
Número de canales	4	canales
Periodo de Beacon (EB)	16	s
Longitud de Slotframe	7	timeslots
Número de paquetes	500	paquetes
Periodo de paquete	70, 140, 500	ms
Tamaño de paquete	110	Bytes
Máximo de retransmisiones	10	retransmisiones

Tabla 4-2: Parámetros de simulación.

4.4. Análisis de resultados

En esta sección se presentan los resultados tras la simulación de los 4 métodos de planificación TSCH, para cada una de las métricas propuestas. En el test 1 se han evaluado el E2E delay y el PDR (apartado 4.4.1), RDC (apartado 4.4.2), PIAT y DSR (apartado 4.4.3). En el test 2 se analiza el comportamiento en un entorno con múltiples flujos (apartado 4.4.4).

4.4.1. Retardo extremo a extremo y tasa de pérdida de paquetes

El tamaño de las colas en cada planificador tiene una relación directa con los E2E delays y el PDR, pues es la cantidad de paquetes que se encuentran esperando a ser enviados. La capacidad por defecto es de 8 paquetes, que en este caso es la utilizada por SDN WISE-TSCH y AMUS. Para Orchestra y MSF se utilizan una capacidad de 64 paquetes. Esta capacidad de la cola reduce la memoria que tiene disponible cada nodo, por lo tanto, los planificadores que requieren un menor uso de ella son más eficientes.

La figura 4-3, muestra el E2E delay para los primeros 500 paquetes, donde ya se tiene un funcionamiento estable que permite observar de forma clara el comportamiento de cada planificador. Debido a que los planificadores centralizados consiguen en todos los casos un menor E2E delay. Se ha utilizado una escala diferente en las figuras 4-3(a), (b), (e), (f), (i) y (j) que permite observar las diferencias entre estos.

Las transmisiones de cada flujo se realizan de forma independiente, desde el origen en el nodo 9, hasta el sink en el nodo 1, para los cuatro tipos de planificadores. En las figuras 4-3(a) (b) (c) y (d), se envía un paquete cada 70 ms. Los planificadores centralizados como SDN WISE-TSCH y AMUS, son los que consiguen tener el menor E2E delay, 50 ms para SDN WISE-TSCH y para AMUS una variación entre 50 ms y 120 ms. Las variaciones de AMUS se deben al tráfico de control que se acumula en la misma cola que el flujo de datos, en este caso la cola nunca es superior a 2 paquetes. Sin embargo, con Orchestra se tiene un E2E delay medio de 2300 ms, debido a que por defecto, Orchestra solo asigna un slot de transmisión unicast en cada slotframe. Por lo tanto, este periodo de envío (70 ms) es el máximo que puede tener Orchestra con una longitud de slotframe de 7 timeslots. Debido a que también debe enviar paquetes de control RPL, hay una acumulación de paquetes en las colas, que se observa como un aumento progresivo a partir del paquete número 100. En este caso, las colas alcanzan un tamaño de hasta 40 paquetes, lo que produce un incremento en los E2E delays. El caso más crítico es el del MSF, que presenta el mismo inconveniente de Orchestra, 1 slot de transmisión por cada slotframe. Además, al inicio de las transmisiones se utiliza un timeslot compartido con otros nodos, por lo que se producen colisiones. Al estar habilitadas las retransmisiones, la cantidad de paquetes en cola crece más rápido que en Orchestra, ya que se acumulan los paquetes que colisionan, provocando un aumento en el E2E delay y una gran cantidad de pérdida de paquetes.

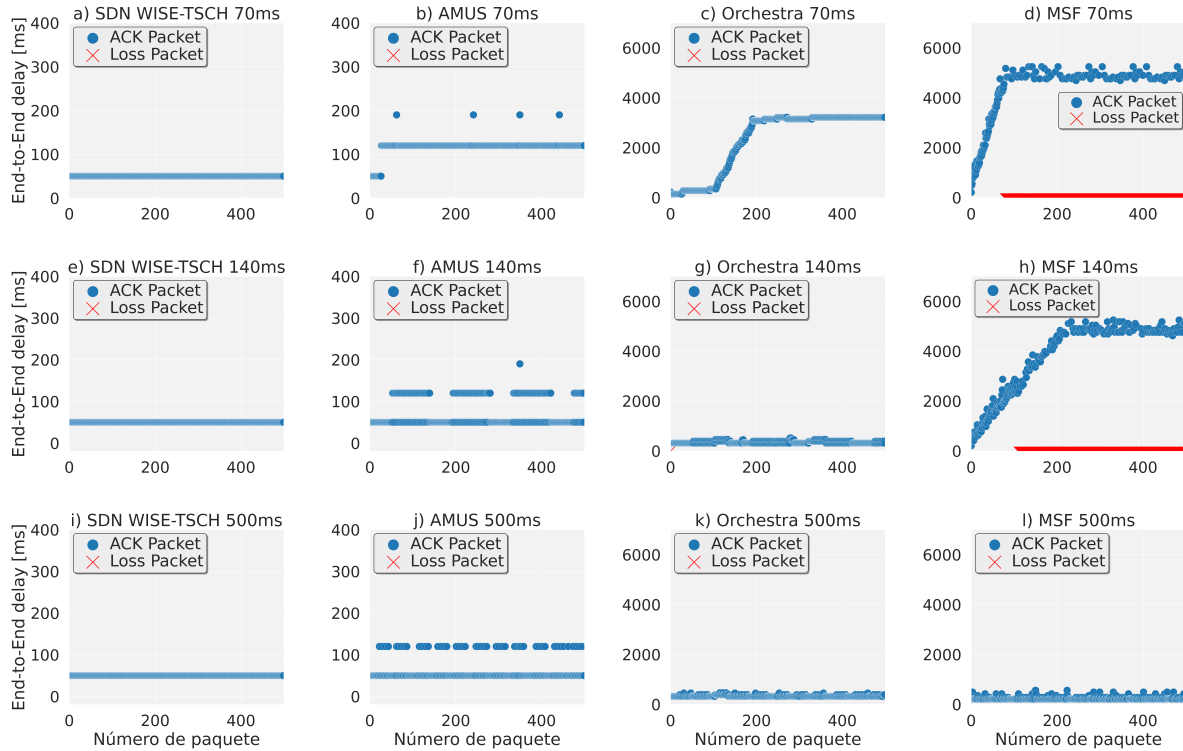


Figura 4-3: E2E delay para distintos periodos de transmisión.

Al aumentar el periodo de envío a 140 ms se obtienen las gráficas de la figura 4-3(e) y (f), donde se observa que para los planificadores centralizados no hay variación y se siguen manteniendo los 50 ms para SDN WISE-TSCH y los 50 ms y 120 ms para AMUS. En la figura 4-3(g) para Orchestra se observa un comportamiento más estable debido a una menor cantidad de paquetes en cola, al igual que en la figura 4-3(h), donde se observa una clara mejora en la pérdida de paquetes y los E2E delay, ya que un mayor ancho de banda disponible permite negociar las celdas unicast con los otros nodos.

Para un periodo de 500 ms, no hay cambios significativos en los planificadores centralizados, como se muestra en la figura 4-3(i) SDN y (j) AMUS. Orchestra en (k) mejora por la menor cantidad de paquetes en colas y su E2E delay medio se encuentra en los 330 ms. En (l) MSF ya no presenta pérdidas y el E2E delay medio de 230 ms es menor que el de Orchestra, ya que con una red poco saturada los slots compartidos tienen una probabilidad menor de colisiones y permiten enviar tráfico de las colas unicast.

A diferencia del E2E delay que es diferente para cada uno de los planificadores, el PDR es cercano al 100 % en la mayoría de los casos. Esto se debe a que SDN WISE-TSCH y AMUS tienen configurada la cantidad de timeslots necesarios para enviar todo el tráfico que generan, por lo tanto, evitan que se pueda desbordar el buffer. Orchestra, tiene un PDR de 100 % debido a que usa buffers de mayor capacidad para evitar la pérdida de paquetes, ya que los

paquetes que no puede enviar por falta de recursos son almacenados aun cuando esto implique aumentar el E2E delay. Sin embargo, MSF es el único que presenta pérdidas significativas en el escenario propuesto por el uso de los slots compartidos que producen colisiones. Solo en casos de baja saturación, MSF consigue aumentar el PDR cerca del 100%.

4.4.2. Consumo de energía

El consumo de energía en TSCH se debe principalmente a la utilización de la radio y puede estimarse como la fracción de tiempo que cada nodo permanece en los estados de transmisión y recepción, lo cual se refleja en el RDC. Por esta razón, el uso de diferentes procesos de planificación genera una variación significativa en los consumos de energía bajo el mismo escenario. En la figura 4-4(a), se observan los diferentes planificadores con un periodo de paquete de 70 ms. MSF tiene el menor consumo, esto es por el uso de un slot compartido y de la asignación dinámica de los slots dedicados. Orchestra tiene un consumo ligeramente superior con un mayor uso de la radio en transmisión, debido a que además del slot compartido, este cuenta con un slot dedicado de transmisión en cada nodo. Para SDN WISE-TSCH y AMUS se utiliza la misma cantidad de estados de transmisión y recepción. La diferencia en el consumo se debe a la cantidad de tráfico de control que envía cada uno. SDN WISE-TSCH utiliza un protocolo propio para intercambiar mensajes con el controlador con un menor impacto sobre el ancho de banda. AMUS además de la comunicación con el PCE utiliza el protocolo RPL.

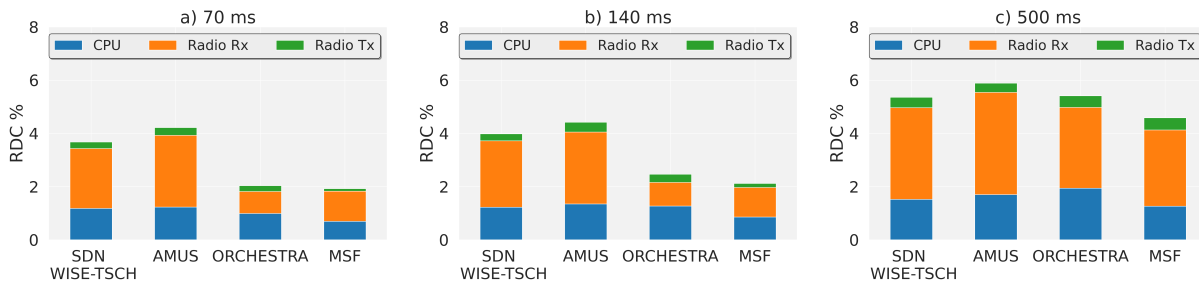


Figura 4-4: Radio duty cycle para cada planificador en función del periodo de paquete.

Al incrementar el periodo de paquete a 140 ms, se incrementa ligeramente el uso de la radio en todos los protocolos, sin embargo, el comportamiento es similar al periodo de paquete anterior. Cuando se utiliza un periodo de paquete de 500 ms, como el mostrado en la figura 4-4(c), se incrementa el uso de la radio en recepción, debido a que el envío con un mayor periodo hará que en total la cantidad de slots de recepción sea mucho mayor a los necesarios. Por esto, el consumo de Orchestra es similar al de SDN WISE-TSCH. Esto se debe a que con SDN es posible adaptar el tamaño de slotframe de acuerdo al periodo de paquete, consiguiendo que la cantidad total de celdas en recepción en SDN WISE-TSCH coincida con

el flujo de mayor deadline. En este caso para 500 ms el slotframe tendría 49 timeslots. En todos los casos AMUS es el planificador con un mayor consumo de energía.

4.4.3. Tiempo entre paquetes y tasa de satisfacción de deadline

El PIAT es una métrica que verifica el efecto de todo el tráfico que fluye a través de la red en un flujo de datos específico. Cuanto más cercano sea este valor al período de paquetes, más determinista será la red, ya que el retardo agregado por los saltos intermedios es constante. En este caso, la figura 4-5 muestra los resultados para diferentes valores del período de paquetes en cada uno de los planificadores TSCH.

Para la figura 4-5(a) SDN WISE-TSCH, se observan 3 líneas claramente diferenciadas, correspondientes a cada uno de los flujos (70 ms, 140 ms y 500 ms). Estas son constantes y coinciden con el periodo de los paquetes, ya que estos son clasificados y enviados por los recursos específicos que tienen asignados. En la figura AMUS 4-5(b) el único flujo que mantiene un retardo constante es el de 70 ms, debido a que su frecuencia siempre tiene paquetes en cola. El resto de flujos presentan variaciones que incluso superan el deadline establecido debido a que no existe sincronización entre la planificación de recursos TSCH y la generación de paquetes. Para Orchestra en la figura 4-5(c) todos los flujos presentan variaciones, producidas por la saturación de los nodos. Sin embargo, la baja saturación de los buffers de Orchestra le permite superar a AMUS, ya que más del 100% de los paquetes están por debajo de 500 ms. En la figura 4-5(d) el comportamiento de MSF tiene una alta aleatoriedad para los flujos con mayor frecuencia, 70 ms y 140 ms, ya que al utilizar las ranuras compartidas pueden producirse colisiones en cualquier salto. En condiciones de baja saturación, tiene un comportamiento más estable con un 80% de los paquetes en torno a los 500 ms.

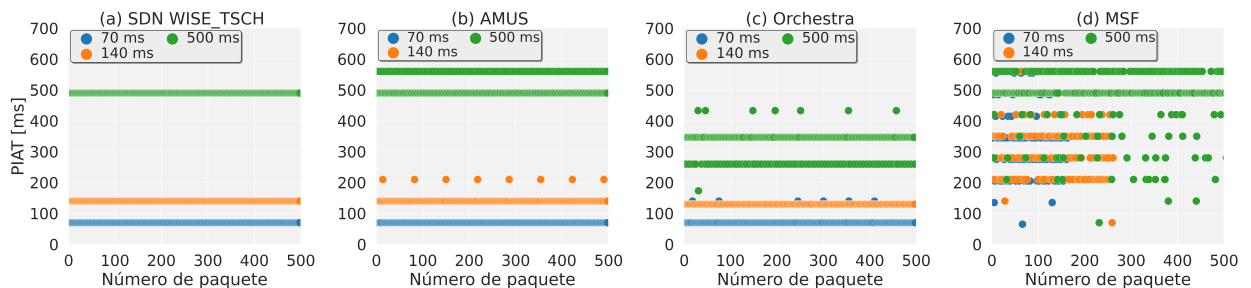


Figura 4-5: PIAT para cada planificador con diferentes periodos de paquetes.

El DSR permite observar de forma acumulada la cantidad total de paquetes que son recibidos en un tiempo menor o igual al periodo de paquete. Un valor alto de DSR implica un flujo constante y predecible entre la fuente y el destino, lo que a su vez implica un mayor determinismo. En la figura 4-6, SDN WISE-TSCH consigue una tasa de satisfacción de deadline

del 100 % en todos los casos, debido a que en este planificador los flujos están diferenciados y la planificación TSCH adaptada para que esté por debajo del deadline. AMUS consigue un 98 %, 90 % y 84 % debido a la falta de sincronismo entre la generación y la planificación que puede producir que la transmisión sea retrasada al slotframe siguiente, cuando esto ocurre los paquetes de control pueden ser enviados por las celdas asignadas al tráfico de datos. Orchestra mantiene una alta tasa de DSR en todos los casos 95 %, 90 % y 95 %, pero al igual que AMUS, se ve afectado por el tráfico de control que no puede diferenciar. Sin embargo, para periodos altos de generación de paquetes este efecto es menor porque el ancho de banda no depende solo del tráfico planificado, sino que también depende de la longitud del slotframe. Por lo tanto, en este caso Orchestra tiene disponible una mayor cantidad de ancho de banda. MSF alcanza el 79 % en condiciones favorables, cuando la congestión de la red es baja.

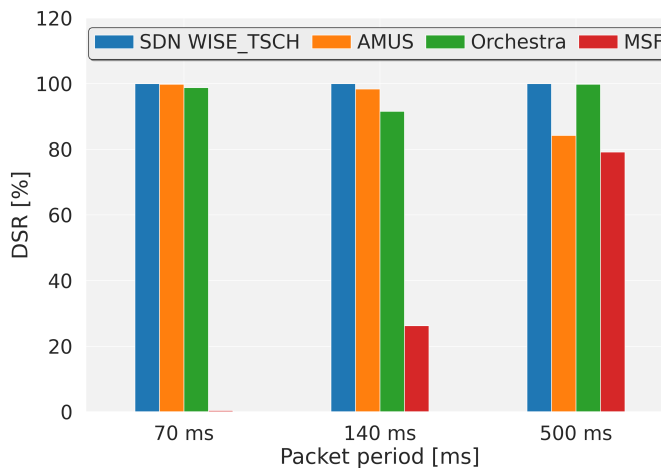


Figura 4-6: Deadline Satisfaction ratio para cada planificador.

4.4.4. Pruebas Multi-flujo

Las pruebas anteriores analizaban el comportamiento de la red de forma independiente con cada flujo de tráfico. Sin embargo, debido a la diversidad de tráfico que fluye por la WSN, en entornos industriales se hace necesario garantizar de forma independiente cada uno de los flujos. Por esto, en este apartado se estudiará el comportamiento de los planificadores con múltiples flujos desde el mismo nodo. Para estas pruebas, se utilizan los dos protocolos con un mayor rendimiento general, SDN WISE-TSCH y Orchestra. En este caso se omite el MSF debido a su bajo rendimiento para cargas de tráfico elevadas y AMUS debido a su dificultad para mantener los deadlines.

Los resultados para el PIAT se observan en la figura 4-7, con un periodo de paquete de 70 ms para el flujo 1, 140 ms para el flujo 2 y 500 ms para el flujo 3. Los nodos que transmiten estos flujos son el 8, 9 y 10, de acuerdo a la figura 4-2. En la figura 4-7(a) puede observarse como SDN WISE-TSCH realiza una clara diferenciación de los paquetes, garantizando la

llegada de forma regular para cada flujo de datos, obteniendo un DSR de 100 % para ambos flujos. A diferencia de Orchestra en **(b)** donde los flujos tienen variaciones que superan el periodo de paquete. Los valores obtenidos de DSR son 17 %, 41 % y 58 %, este rendimiento es claramente inferior al mostrado en figura 4-5, debido a que la red tiene una mayor saturación y al ancho de banda disponible es igual en los dos casos.

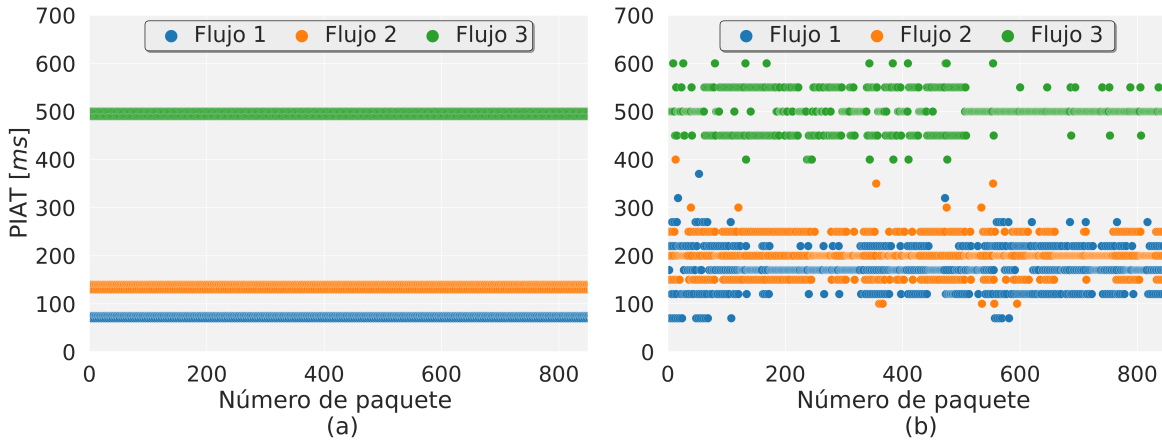


Figura 4-7: PIAT para cada uno de los planificadores con múltiples flujos de datos.

Los resultados obtenidos se pueden resumir en la tabla 4-3. SDN WISE-TSCH es el único que puede diferenciar flujos unicast, por lo tanto, puede garantizar parámetros de calidad de servicio de forma independiente. AMUS tiene un rendimiento similar al SDN WISE-TSCH cuando solo hay un flujo unicast, en cambio, al aumentar el número de flujos pierde determinismo. Esto lleva a incrementos del E2E delay y el PIAT. Por lo tanto, en una red multiflujo su comportamiento es similar al de Orchestra, pero con un consumo de energía mucho mayor. Por esto, en la implementación del testbed, solo se utilizará el planificador Orchestra y SDN WISE-TSCH.

	SDN WISE-TSCH	AMUS	MSF	ORCHESTRA
Construcción de topología	SDN WISE: Beacon y Report	RPL + COAP	RPL	RPL
Latencia (Retardos)	Mínima	Mínima	Alta	Baja
Fiabilidad (PDR)	Alta	Alta	Baja	Alta
Paquetes de control	Medio	Alto	Bajo	Bajo
Consumo de energía	Medio	Alto	Bajo	Bajo
Rendimiento Multiflow	Alto	Bajo	Bajo	Medio

Tabla 4-3: Resumen de características de los distintos planificadores TSCH.

4.4.5. Testbed

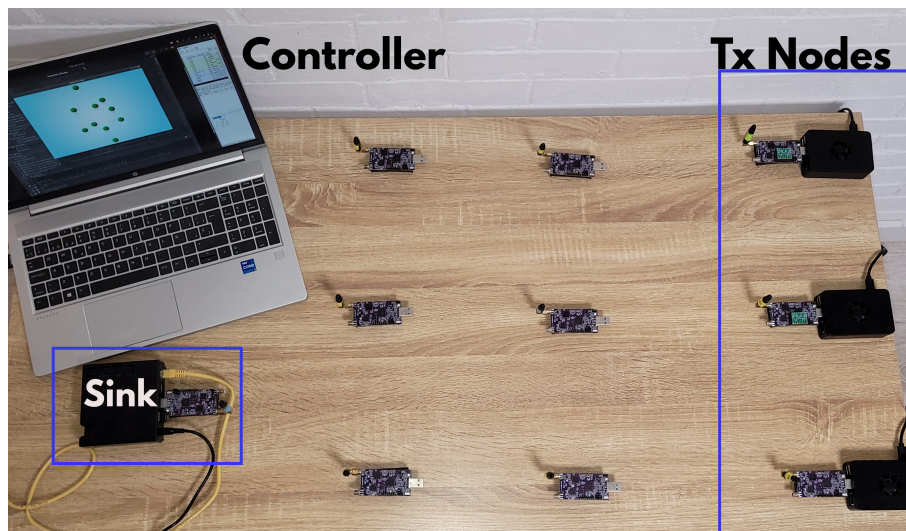


Figura 4-8: Testbed usando OpenMote-B.

La mayoría de estudios que evalúan los algoritmos de planificación solo trabajan en entornos de simulación y son escasas las evaluaciones experimentales. En esta tesis, para comprobar los resultados obtenidos en la simulación se ha replicado la topología de la figura 4-2 en un entorno real, utilizando 10 nodos OpenMote-B con el sistema operativo Contiki-NG. Con el fin de obtener una red multisalto en un espacio reducido, la potencia de transmisión de los nodos fue configurada a -15 dBm, los demás parámetros son iguales a los utilizados en la simulación. Para capturar los paquetes y obtener las estadísticas de rendimiento, se han

instalado 2 sniffers, ubicados cerca de los nodos transmisores 8,9 y 10 y el destino en el Sink. Los sniffers están compuestos por 4 nodos OpenMote-B que se utilizan para capturar los paquetes de cada uno de los canales de forma independiente. Estos van conectados a una Raspberry Pi que combina las capturas de los 4 nodos y los envía al analizador de protocolos Wireshark, el resultado es la topología que se muestra en la figura 4-8.

4.4.6. Retardo extremo a extremo y tiempo entre paquetes

En la figura 4-9 se observa el E2E delay y el PIAT para cada uno de los nodos transmisores utilizando SDN WISE-TSCH y Orchestra. Los nodos tienen asignados diferentes combinaciones de flujos como se muestra en la figura 4-2. Los dos planificadores SDN WISE-TSCH en (a) y Orchestra en (b), consiguen el mismo comportamiento que en la simulación, donde SDN WISE-TSCH tiene el E2E delay mínimo de 50 ms para los tres saltos hasta el destino. Este comportamiento es propio de una planificación optimizada, ya que permite enlazar las celdas de origen a destino entre los diferentes nodos. A diferencia de Orchestra, donde las celdas están asignadas de forma local, por lo tanto, la transmisión de cada salto se hace en diferentes slotframes. Además, se producen encolamientos haciendo que el retardo aumente de forma considerable.

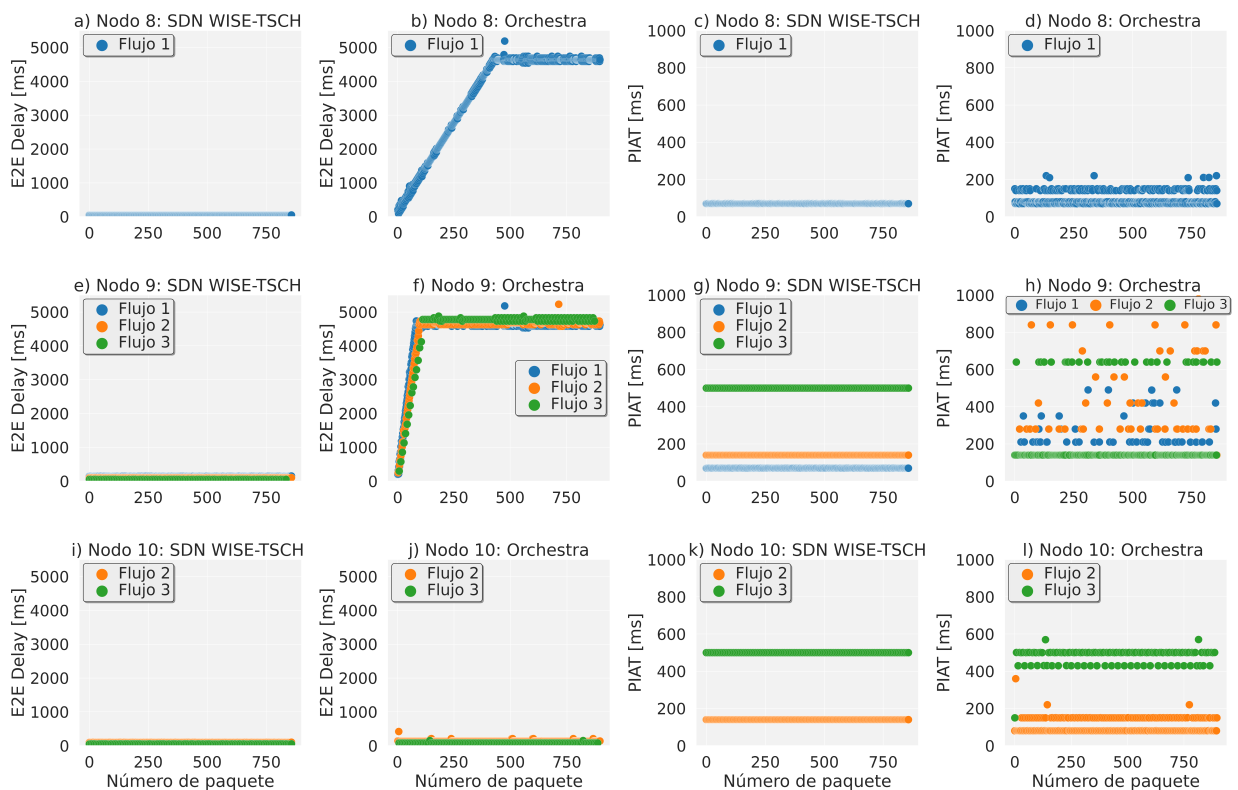


Figura 4-9: E2E delay y PIAT en testbed con SDN WISE-TSCH y RPL + Orchestra.

En este caso con un slotframe de solo 7 timeslots SDN WISE-TSCH consigue una reducción del E2E delay del 75 % respecto al obtenido por Orchestra. Además, para este nodo en (c) se muestra que el PIAT es constante con SDN WISE-TSCH, se mantiene en los 70 ms. Esto junto con el bajo E2E delay significa que entre el origen y destino hay un flujo constante de paquetes y que no se producen retardos en ningún punto de la red. Por el contrario, Orchestra en (d) presenta variaciones de hasta 380 ms. Los resultados del nodo 9, que presenta el mayor caso de saturación al tener los tres flujos se observa en las figuras (e) a (h). SDN WISE-TSCH (e) mantiene el E2E delay mínimo (50 ms) en los tres flujos debido a la asignación completa de los recursos.

Por el contrario, Orchestra en la figura 4-9(f), presenta unos E2E delays para todos los flujos de hasta 4000 ms y una alta cantidad de pérdidas, esto se debe a la congestión de las colas, donde Orchestra no puede configurar el ancho de banda suficiente para los flujos que genera el nodo. Por esto, el PIAT es estable en SDN WISE-TSCH como se muestra en la figura 4-9(g), donde se mantiene cerca del periodo de generación de paquetes, mientras en Orchestra esta falta de ancho de banda se refleja con una alta dispersión, como se puede ver en la figura 4-9(h). El nodo 10, presenta los flujos con menor requisito de ancho de banda, por esto se observa como el E2E delay para los dos protocolos en la figura 4-9 (i) y (j), es muy similar, con pequeñas excepciones que se deben al tráfico RPL que se envía por las colas unicast. En este caso el PIAT también se observa más acotado como en la figura 4-9 (k) y (l), por lo que el rendimiento en este nodo es muy similar para los dos casos de estudio.

4.4.7. Tasa de satisfacción de deadline

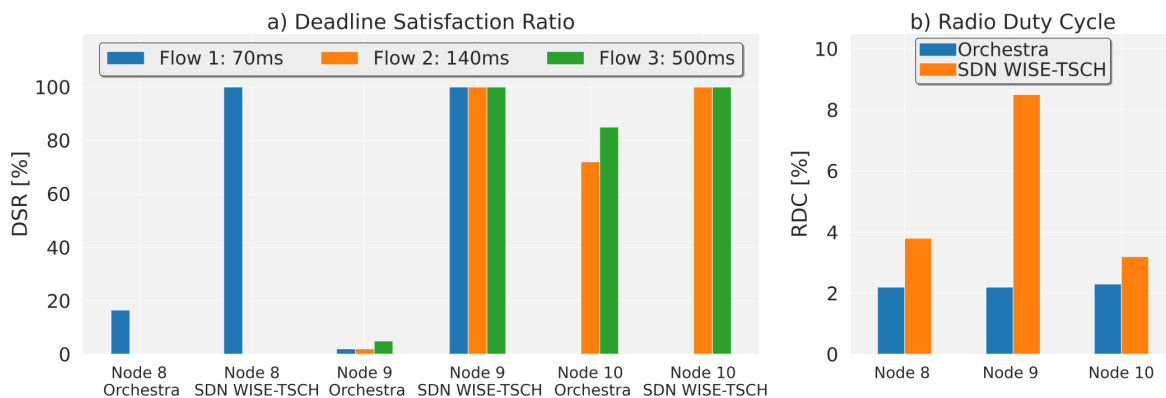


Figura 4-10: DSR obtenido en el testbed para cada nodo Tx.

Los E2E delays mínimos obtenidos por SDN WISE-TSCH y los valores constantes del PIAT, demuestran el determinismo que tiene esta propuesta. Esto puede observarse en la figura 4-10 (a) donde en todos los nodos se puede garantizar el deadline para cada flujo con una tasa

del 100 %. A diferencia de Orchestra que necesita condiciones de baja saturación como la del nodo 10, para estar en el 85 %. Para el nodo 9 que genera la mayor cantidad tráfico, Orchestra no puede garantizar los deadlines obteniendo solo el 2 % en el mejor caso. Finalmente, para el nodo 8, Orchestra obtiene solo el 17 %.

4.4.8. Consumo de energía

En las pruebas anteriores, se observa como SDN WISE-TSCH tiene un rendimiento muy superior al de RPL + Orchestra, sin embargo, también requiere un mayor consumo de energía, el RDC puede observarse en la figura 4-10 (b), para cada nodo usando los dos planificadores. En el primer caso, en el nodo 8, Orchestra tiene un consumo 40 % inferior al obtenido por SDN WISE-TSCH. Para el caso de mayor congestión, el consumo de Orchestra es menor en un 75 %, sin embargo, la diferencia en rendimiento también es notable. Finalmente, el caso en el nodo 10 donde el rendimiento es similar, la diferencia de rendimiento es de un 28 %.

4.5. Conclusiones

En este capítulo se ha realizado un análisis de diferentes algoritmos de planificación de recursos para el protocolo TSCH con el fin de determinar la calidad de servicio (QoS) que obtiene cada uno en distintos escenarios de red, tanto mediante simulación como en un entorno real de pruebas, para diferentes tipos de tráfico.

Se ha demostrado como los protocolos basados en SDN como SDN WISE-TSCH ofrecen una precisión mucho mayor en la clasificación de recursos, permitiendo distinguir con mayor exactitud entre los diferentes tipos de tráfico. De esta forma, se logra una reducción en el E2E delay de hasta un 70 % en comparación con el protocolo Orchestra, mientras que se alcanza una tasa del 100 % de cumplimiento del deadline. Además, en condiciones de saturación, SDN WISE-TSCH es el único protocolo capaz de garantizar los requisitos de cada tipo de tráfico de forma independiente.

Por otro lado, se ha comprobado que protocolos como Orchestra y MSF ofrecen la ventaja de ser fáciles de desplegar, ya que no requieren intervención externa y todos los procesos de reconfiguración se realizan de forma local. Sin embargo, su rendimiento es óptimo solamente en condiciones de muy baja saturación, y cuando se utilizan flujos de tipo monitorización. Además, en entornos con múltiples flujos de información, no permiten garantizar de forma independiente cada uno de los flujos, lo que ocasiona un aumento del delay y del PIAT. Por lo tanto, se aleja del comportamiento determinista, necesario en entornos industriales donde se deben garantizar cada uno de los requisitos de calidad de servicio de forma independiente.

En cuanto a los protocolos centralizados como AMUS, los resultados muestran cómo consiguen un alto rendimiento en términos de tasa de paquetes recibidos correctamente y retraso

extremo a extremo gracias a una planificación de recursos optimizada. Sin embargo, presentan un alto consumo energético y dificultades en el manejo del tráfico de control, provocando una alta variación en el PIAT. Esto reduce su comportamiento determinista debido a la falta de integración entre las diferentes capas del protocolo, lo que lleva a una operación inflexible en la que no es posible diferenciar entre flujos de información.

Por último, la capacidad de garantizar parámetros de calidad de servicio de forma independiente en entornos multiflujo plantea la pregunta de si este comportamiento se puede aplicar a escenarios más complejos, como con nodos en movimiento o en situaciones de alta densidad de nodos. Estos temas se tratarán en los próximos capítulos.

Capítulo 5

Gestión dinámica de recursos en IWSN con nodos móviles

5.1. Introducción

En la actualidad, el sector industrial busca optimizar sus procesos y adaptarse rápidamente a los cambios tecnológicos. Esta transformación requiere interconectividad entre todos los sistemas relevantes para tomar decisiones basadas en grandes volúmenes de información. Por lo tanto, se deben integrar una serie de elementos que, combinados con el dinamismo de los entornos de producción, generan la necesidad de proporcionar soporte para elementos móviles, como los *Automatic Guided Vehicle* (AGV) que se utilizan ampliamente en tareas de logística.

En este capítulo se presenta *Mobile Multicast Forwarding SDN* (MMF-SDN), una alternativa basada en SDN que permite a los nodos móviles mejorar el determinismo y garantizar parámetros de QoS. Se basa en el uso del conocimiento global proporcionado por SDN y la sincronización obtenida con TSCH, para agregar al planificador TSCH centralizado una extensión que realice una asignación de recursos multicast acumulada. Esto consiste en asignar una celda de transmisión que puede tener múltiples receptores para cada nodo móvil. Estas celdas se agregarán a cada uno de los nodos padres a medida que se descubran como posibles puntos de entrada de la red, y permanecerán activas hasta que sean eliminadas por el controlador. Por lo tanto, con cada asignación de recursos, se aumenta la cobertura y conectividad del nodo móvil sin aumentar las transmisiones que este debe realizar. Esta asignación eliminará el problema de la señalización elevada, porque los recursos se asignan solo la primera vez que se agrega un nodo padre. En este enfoque multicast, el uso de una sola celda de entrada para cada nodo móvil y la reducción del ancho de banda requerido para el tráfico de control permite usar slotframes más compactos o tener una menor ocupación de este, lo que permite aumentar el número máximo de nodos y flujos de datos.

5.2. Movilidad en IWSN

El reto de los nodos móviles en IWSN ha sido abordado en múltiples trabajos, donde se ha evaluado el efecto de los nodos móviles en RPL con Orchestra [54] y TSCH minimal [55]. Los resultados obtenidos muestran cómo el tráfico de control de RPL aumenta, debido a las desconexiones de los nodos móviles. Cada vez que un nodo móvil se desconecta, debe iniciar un proceso de descubrimiento para obtener un nuevo nodo padre, lo que conlleva un tiempo de convergencia antes de restablecer la conexión. Además, los nodos dependen de temporizadores para detectar su desconexión y posteriormente iniciar el proceso de descubrimiento. Esto provoca un aumento de la pérdida de paquetes y de la latencia cada vez que un nodo sale del área de cobertura.

En [56] se intentó superar esta dificultad con el uso de *beacons pasivos*, los cuales consisten en usar el ACK de las transmisiones para reducir los temporizadores que permiten actualizar rápidamente los nodos padres que salen del área de cobertura. También se han probado diferentes técnicas para optimizar el proceso de enrutamiento, como segmentar la red en áreas circulares definidas por las raíces del grafo acíclico dirigido (DAG) [57] o utilizar filtros de Kalman para mejorar las decisiones de RPL con predicciones de posición [58]. Además, se han mejorado las funciones de planificación distribuida, incluyendo la predicción de la posición, el tamaño de la cola y la tasa de generación de paquetes para negociar el número de celdas necesarias [59]. Este proceso de negociación tiene limitaciones en entornos de movilidad, ya que cuando se produce un cambio de padre, el nodo necesita tiempo para descubrir a los nuevos vecinos e iniciar el proceso de negociación. Esto puede reducirse reservando celdas dedicadas a la negociación y añadiendo múltiples paquetes de control para detectar cambios de topología [60]. Sin embargo, aunque representan una mejora con respecto al RPL tradicional, están lejos de ofrecer la misma fiabilidad que los nodos fijos.

Estos estudios previos muestran cómo el soporte de nodos móviles aumenta considerablemente la complejidad de las decisiones en los nodos y la cantidad de paquetes de señalización. Por ello, en esta tesis, se propone la implementación de SDN en IWSNs como una mejor opción que las soluciones tradicionales, como RPL. La razón de ello es que SDN ofrece una serie de ventajas adicionales, que compensan el aumento del tráfico de control. Estas ventajas son:

- La reducción de la complejidad de los procesos en los nodos
- La visión global de los elementos de la red

La combinación de SDN y TSCH aumenta la fiabilidad, el determinismo y la flexibilidad de la red. Esto se ha demostrado en el capítulo 3, donde es posible garantizar diferentes parámetros de QoS para diferentes tipos de flujo de forma dinámica y con baja complejidad en comparación con RPL con Orchestra. Por ello, en este capítulo se busca extrapolar este comportamiento a entornos con nodos móviles, donde se requiere una configuración dinámica de los flujos, ya que factores como el cambio de topología afectan de forma directa el

determinismo y la flexibilidad.

Para ofrecer soporte a nodos en movimiento usando SDN sobre TSCH, L. Bello en [62] propone *Forwarding and Time-Slotted channel hopping scheduling over SDN* (FTS-SDN). El objetivo es aumentar la frecuencia y fiabilidad en el intercambio de los paquetes de control, para así disminuir el tiempo de actualización de la topología. El controlador enviará las nuevas reglas de enrutamiento con cada cambio de nodo padre. Para adaptarse a los cambios, el controlador debe actualizar la topología cada vez que el nodo móvil cambia de nodo padre. En este caso, los nodos deben informar de su existencia a través de mensajes de beacon que se envían en broadcast para que sean recibidos por todos los nodos en el rango de cobertura. Además, deben enviar al controlador la información de los beacon recibidos. Esto se realiza mediante el envío de paquetes de reporte, donde se incluyen todos los nodos desde los que se ha recibido mensajes de beacon. Por eso, a mayor cantidad de paquetes de control, más actualizada estará la topología, y más rápido se adaptará a los cambios del nodo móvil. De hecho, la velocidad a la que se desplace el nodo móvil es un parámetro que puede afectar en este enfoque. Cada vez que el controlador actualice la topología, deberá generar las rutas e informar a los nodos de la nueva ruta con un paquete OpenPath. En FTS-SDN la planificación TSCH es fija y está pre-configurada, se realiza de acuerdo a las siguientes reglas:

- Mínimo 1 celda de *uplink* para cada nodo.
- Mínimo 1 celda de *downlink* para cada nodo.
- 1 slot compartido para cada nodo.

Esta solución tiene una alta dependencia del tráfico de control, por lo que una mayor frecuencia y fiabilidad en este, aumentan la capacidad y velocidad de la red para adaptarse a los cambios. Por esto, para garantizar el envío del tráfico de control, se utiliza un caso particular de los slots compartidos, donde se asigna de forma exclusiva un slot compartido de transmisión para cada nodo, eliminando así la probabilidad de colisión. Esto implica, que en cada slot compartido solo un nodo estará transmitiendo y todos los demás estarán en recepción. En estos slots no es posible planificar comunicaciones paralelas. Por esto, el inconveniente principal de este enfoque es el aumento significativo en la longitud y ocupación del slotframe. Esto se refleja en un aumento en el consumo de energía, una reducción del ancho de banda para datos y una administración de red más compleja por necesitar slotframes de mayor tamaño.

Por ejemplo, la figura 5-1 muestra el primer escenario utilizado en [62], una topología sencilla con 5 nodos (Sink o S, A, B, C y el móvil M). La trayectoria de M es circular (azul), y el único nodo que puede comunicarse con el Sink es el B. La planificación en [62] está formada por un slotframe compuesto de 13 slots y 4 canales. El controlador tiene asignado solo la ruta $M \rightarrow B \rightarrow S$, de acuerdo al rango actual de cobertura. En la planificación de la figura 5-1

se observan asignados 5 slots compartidos con el funcionamiento particular de FTS-SDN, 1 por cada nodo (rojo). Además, múltiples entradas del nodo Móvil (M) en diferentes puntos del slotframe (6, 9, 12; verde) y un único slot de llegada al nodo Sink (11; amarillo). Por tanto, en esta solución se utilizan 5 slots compartidos en un slotframe de 13. El E2E delay varía en función del nodo de entrada. Este será 150 ms, 120 ms y 50 ms para las entradas A, B y C respectivamente. Estas diferencias se deben a la distribución de los slots en el slotframe realizada en [62], que para las entradas desde los nodos A y B necesitan esperar hasta el siguiente slotframe para completar la transmisión. Por ejemplo, el nodo móvil envía un paquete de datos que es recibido por el nodo A en el slot 9. Luego, este paquete es retransmitido al nodo B 60 ms después en el slot 2, y finalmente llega al sink en el slot 11, 90 ms después de haber sido recibido por el nodo B. Esto implica un tiempo total de 150 ms desde la recepción en el nodo A hasta la recepción en el sink.

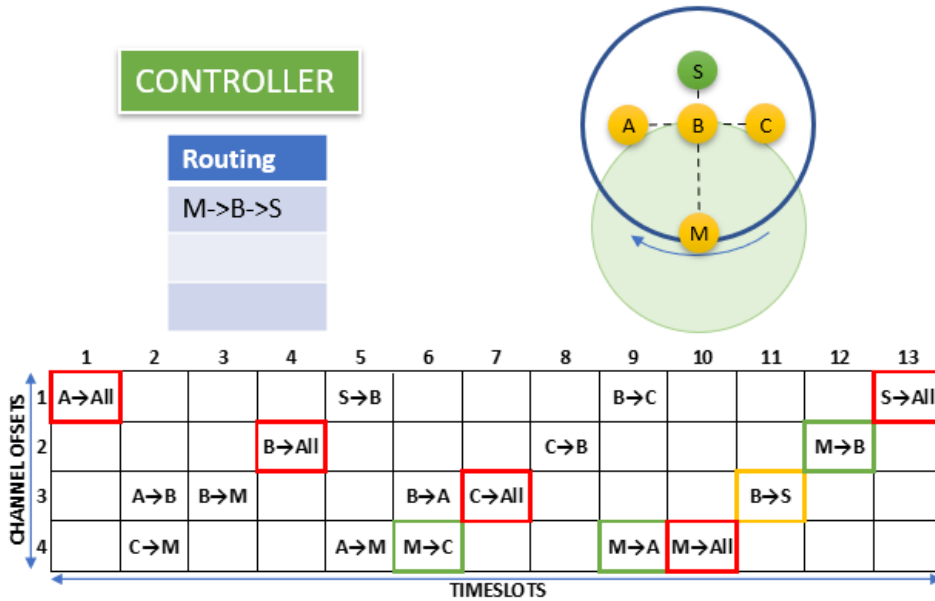


Figura 5-1: Topología y slotframe usado en [62].

5.2.1. Mobile Multicast Forwarding SDN (MMF-SDN)

La implementación usada en FTS-SDN no representa un cambio significativo en la forma como se realiza la conexión y reconexión del nodo móvil. Por ello, se propone Mobile Multicast Forwarding SDN (MMF-SDN) que explota las ventajas de SDN WISE y TSCH, para mejorar el soporte a los nodos móviles y acercarlos a niveles de determinismo similares a los de los nodos fijos. MMF-SDN permite efectuar el cambio de nodo padre de forma instantánea y transparente. Por esto, tiene una baja dependencia del tráfico de control, por lo tanto, no se tienen las limitaciones actuales debidas al cambio de padre o a la saturación con el tráfico de control.

Esta propuesta se basa en el conocimiento global de la red en SDN, los amplios recursos disponibles en el controlador y a la sincronización obtenida por el uso de la capa MAC TSCH, para ofrecer a los nodos móviles un alto nivel de determinismo. MMF-SDN implica la administración de los nodos móviles como una fuente multicast, esto quiere decir que el nodo móvil tendrá múltiples receptores para un mismo paquete. Estos estados de recepción para el nodo móvil estarán sincronizados en una misma celda de la planificación TSCH para todos sus nodos padre. Por lo tanto, los factores que afectan a la topología como la posición y velocidad del nodo móvil tendrán una menor relevancia. Así, será posible aumentar los periodos de beacon y de reporte, lo que reduce el tráfico de control y simplifica la administración de los nodos móviles. Para lograrlo, los paquetes de reporte provenientes de un nodo móvil deben estar diferenciados de los paquetes de reporte de los nodos fijos. Así, cuando el controlador reciba un reporte de un nodo móvil actualizará la topología, pero no hará cambios en el enrutamiento o la planificación TSCH. Por lo tanto, la administración de estos procesos se lleva a cabo mediante la implementación de una nueva aplicación *Mobile Node Manager* que, al estar ubicada en la capa de aplicación, tiene una comunicación directa con el controlador SDN como se muestra en la figura 3-3. Esta aplicación será la encargada de almacenar los diferentes nodos padre de un móvil, determinar qué nodos deben ser receptores multicast y solicitar al controlador la instalación de los recursos necesarios (Routing + TSCH) para estos nodos. Los recursos asignados por solicitud de *Mobile Node Manager* son acumulados. Esto quiere decir que, al asignar los recursos para el nuevo padre, se mantienen los asignados a los padres anteriores. Uno de los recursos instalados, es la celda de recepción multicast, que coincide en todos los nodos padre de un mismo nodo móvil, por lo que en cada asignación de recursos se aumenta la cantidad de receptores multicast. Con este enfoque, se consigue tener planificadas todas las entradas del nodo móvil, sin generar una sobrecarga de actualizaciones y configuraciones frecuentes. Por lo tanto, es posible tener un comportamiento estable con nodos móviles, después de una fase inicial de descubrimiento. Este es el tiempo que tarda el controlador en conocer todas las entradas de un nodo móvil y asignar los recursos de enrutamiento y planificación TSCH. La duración de esta fase, depende de la longitud, de la trayectoria y de la velocidad de desplazamiento del nodo móvil. En un entorno industrial acotado se corresponde con el primer desplazamiento del nodo móvil a través de la trayectoria completa.

Dado que la construcción de la topología en SDN WISE se realiza a través de la información que llega desde los nodos, la velocidad de actualización también depende de la capacidad y fiabilidad de la red para enviar el tráfico de control. Como en FTS-SDN se envía la ruta correspondiente a cada cambio en la topología, es necesario garantizar una alta fiabilidad en los paquetes de control. Por esta razón, en MMF-SDN se busca reducir la dependencia del tráfico de control a través de una asignación de recursos multicast acumulada.

Esta menor dependencia del tráfico de control, aumenta la robustez, lo que permite reducir la frecuencia de dicho tráfico. Por ello, en MMF-SDN es posible utilizar el funcionamiento

por defecto de los slots compartidos, donde los nodos pueden transmitir en cualquiera de los slots compartidos si tiene paquetes en cola o estar en modo recepción en caso contrario. Con esto es posible reducir la cantidad de slots compartidos que se asignan. Sin embargo, se pasa a tener un medio por contienda, lo que implica una probabilidad de colisión. Para reducir la cantidad de colisiones se debe asignar una cantidad óptima de slots compartidos para reducir la probabilidad de colisión. Esta depende directamente de la cantidad de tráfico de control que se genere, por esto es necesario llevar a cabo un análisis del mismo.

5.2.2. Caracterización y análisis del tráfico de control

El tráfico de control es esencial para el funcionamiento de una SDN, ya que es utilizado por el controlador para construir la topología y enviar las reglas de flujo a los nodos. En SDN WISE-TSCH este tráfico, está compuesto por los paquetes periódicos como los de beacon, reporte y los paquetes aperiódicos que son los de configuración (OpenPathTSCH). La cantidad total de tráfico de control depende principalmente del periodo de generación de los paquetes de beacon y de reporte. Con una mayor frecuencia de generación se obtiene una red más actualizada, permitiendo que sea más adaptable a los cambios. Sin embargo, este tipo de tráfico es una sobrecarga considerable para la red y puede afectar a los flujos de datos. Por lo tanto, se deben configurar de forma adecuada los parámetros para que el impacto sobre la red sea el mínimo, en MMF-SDN las técnicas utilizadas son:

- Reducir: Optimizar los periodos de beacon (B_p) y de reporte (R_p)
- Limitar: Enviar este tipo de tráfico únicamente a través de las slots compartidos.
- Optimizar: Asignar un número óptimo de slots compartidos para mantener la fiabilidad del tráfico de control.

Notación	Definición	Unidades
C_{pk}	Paquetes de control	paquetes
C_{BW}	Ancho de banda de control	paquetes
Sh_{SF}	Timeslots compartidos	timeslots
SF_{sz}	Longitud de Slotframe	timeslots
H_i	Cantidad de saltos entre el nodo i y el sink	Saltos
T_s	Timeslots	ms
T	Período de evaluación	s
B_p	Periodo de Beacon	s
R_p	Periodo de Reporte	s

Tabla 5-1: Descripción de los parámetros utilizados

En la tabla **5-1** se muestra la notación utilizada. La cantidad total de tráfico de control (C_{pk}) para SDN WISE-TSCH en un periodo de tiempo T segundos está determinado por la ecuación (5-1), donde n es la cantidad de nodos en la red; Bp y Rp los periodos de beacon y reporte; H_i la distancia en saltos al Sink desde cada nodo i . El primer término de la ecuación (5-1), $\frac{T}{Bp}$, es la cantidad de paquetes de beacon que genera un nodo. Estos paquetes no son reenviados por los nodos, por eso no tiene un factor que dependa de la distancia al Sink, a diferencia del segundo término, que son los paquetes de reporte que deben pasar por múltiples nodos hasta llegar al controlador.

$$C_{pk} = n \frac{T}{Bp} + \sum_{i=1}^n \frac{T}{Rp} H_i \quad (5-1)$$

Debido a que los paquetes de control serán enviados por los slots compartidos, se debe optimizar la cantidad de slots compartidos que se asignen por cada slotframe. Una cantidad baja de slots compartidos creará una alta tasa de colisiones y evitará que el controlador actualice la topología. Por contra, una cantidad muy alta de slots compartidos aumentará el consumo energético y reducirá el ancho de banda disponible para datos. Para el cálculo del ancho de banda de control (C_{BW}) se utiliza la ecuación (5-2), donde se obtiene el ancho de banda en función de la cantidad de slots compartidos en el slotframe (Sh_{SF}) y la longitud del slotframe (SF_{sz}) en un periodo de T segundos, teniendo en cuenta la duración del slot Ts . El primer término $\frac{Sh_{SF}}{SF_{sz}}$, es la relación de ocupación del slotframe con slots compartidos. Un incremento en cantidad de slots compartidos, implica una mayor ocupación y un mayor ancho de banda para control. Se puede conseguir el mismo ancho de banda con diferentes longitudes del slotframe manteniendo la relación de ocupación.

$$C_{BW} = \frac{Sh_{SF}}{SF_{sz}} \cdot \frac{T}{Ts} \quad (5-2)$$

La cantidad óptima de slots compartidos se obtiene a partir de la ecuación (5-3), una variación de la ecuación (5-2) que garantiza el valor mínimo del ancho de banda de control. Al reemplazar la cantidad total de paquetes de control obtenida en la ecuación (5-1) y los parámetros del slotframe, se obtiene un número entero que corresponde con la cantidad de slots compartidos. La figura **5-2** representa la cantidad de tráfico de control que puede ser enviado, variando la longitud del slotframe y la cantidad de slots compartidos asignados.

$$Sh_{SF} = \left[C_{pk} \cdot SF_{sz} \cdot \frac{Ts}{T} \right] \quad (5-3)$$

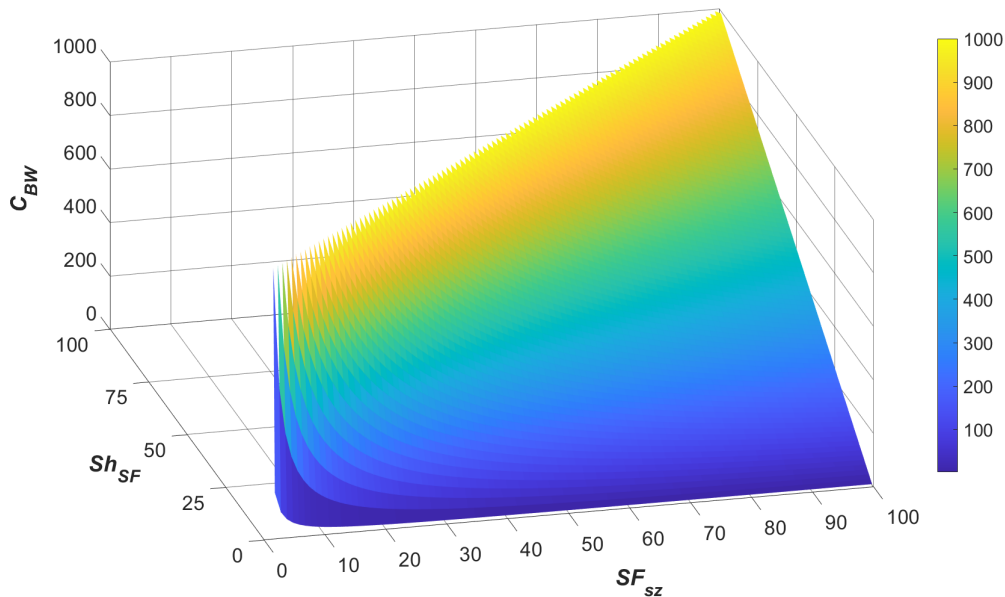


Figura 5-2: Efecto del coeficiente de ocupación en la cantidad de tráfico de control.

5.2.3. Asignación de la planificación TSCH

En MMF-SDN la planificación está basada en flujos, esto quiere decir que las celdas son asignadas desde el origen hasta el destino y el resultado es enviado para cada flujo a través de un solo paquete OpenPathTSCH. Esto permite tener un mayor control sobre las celdas que son asignadas, al poder asignar o eliminar flujos específicos, en lugar de tener que modificar toda la planificación de un nodo. En MMF-SDN se han incluido celdas multicast, estas contienen múltiples receptores en la misma celda para los paquetes de datos enviados por el nodo móvil. Para esto, se ha añadido a los nodos una dirección de multicast global, que es usada por los nodos móviles para transmitir sus paquetes de datos y por los receptores para recibir en la misma celda. Así, la transmisión de los nodos móviles no se realiza a un nodo específico y se garantiza una cola dedicada para el tráfico multicast, lo que permite aumentar el determinismo. Otra de las diferencias respecto a la planificación TSCH en FTS-SDN es la forma en que se asigna la cantidad de slots compartidos. En MMF-SDN se utiliza una cantidad óptima de slots compartidos, que se obtiene de acuerdo a la ecuación (5-3). Con esto, se reduce la cantidad total de slots compartidos, pero se aumenta la cantidad de este tipo de slots en los que un nodo puede transmitir. Este aumento se debe al funcionamiento por defecto de los slots compartidos, donde los nodos pueden transmitir en cualquiera de ellos si tienen paquetes en cola. Así, al tener cada nodo más oportunidades de transmisión, aumenta el ancho de banda para tráfico de control por nodo, aunque se reduzca de forma global.

Este funcionamiento permite que los nodos puedan adaptarse de forma dinámica a las necesidades de tráfico de la red. Por ejemplo, los nodos más alejados del sink solo deben enviar sus paquetes de beacon y de reporte. Por lo tanto, realizan una cantidad menor de transmisiones que los nodos más cercanos al sink, que deben reenviar el tráfico de control recibido desde otros nodos. La asignación de slots dedicados puede congestionar los nodos centrales, a diferencia de la asignación optimizada donde se tiene en cuenta el tráfico máximo que puede generarse y permite a cada nodo acceder a la cantidad de recursos que requiera. Por esta razón, la planificación de las celdas dedicadas se calculó basándose en el planificador desarrollado en el capítulo 3. Este planificador utiliza las rutas calculadas por el algoritmo de Dijkstra en el gestor de rutas y asigna las celdas necesarias para cada salto en la ruta. Los algoritmos presentados en [61] tienen una complejidad computacional de orden cuadrático $O(n^2)$, donde n es el número de nodos. Esto tiene poco impacto en el MMF-SDN propuesto, ya que estamos tratando con un sistema industrial típico tal como se define en la IEC 62264 (ANSI/ISA 95), donde el número de nodos suele estar limitado entre 10 y 30.

De acuerdo a la ecuación (5-1), el ejemplo mostrado en la figura **5-1** genera 90 paquetes de control (C_{pk}) cada 10 segundos, usando periodos de beacon y de reporte de 1 y 2 segundos respectivamente. De estos, 50 corresponden a paquetes de beacon y 40 a paquetes de reporte. Aunque los paquetes de reporte tienen la mitad de frecuencia que los de beacon, dependen también de la distancia al sink. Teniendo en cuenta el peor caso para el nodo móvil que es cuando entra por los nodos A y C el resultado de la ecuación es el siguiente:

$$C_{pk} = T \frac{n}{Bp} + T \sum_{i=1}^n \frac{H_i}{Rp} = T \frac{5}{Bp} + T \left(\overbrace{\frac{2}{Rp}}^{\text{H Nodo A}} + \overbrace{\frac{1}{Rp}}^{\text{H Nodo B}} + \overbrace{\frac{2}{Rp}}^{\text{H Nodo C}} + \overbrace{\frac{3}{Rp}}^{\text{H Nodo M}} \right) = 90$$

En el desarrollo de la ecuación se observa el número máximo de saltos para llegar al sink de cada nodo, para considerar la cantidad total de paquetes que se transmiten en la red. Los reportes del sink no se consideran porque se envían directamente al controlador, sin pasar por la red inalámbrica. Utilizando el valor obtenido para los C_{pk} , la cantidad de slots compartidos de acuerdo con la ecuación (5-3) es [1.1], por tanto, al redondear al entero superior, la cantidad óptima de slots compartidos es de 2 para una longitud del slotframe de 13 slots.

Con el enfoque MMF-SDN, se pasa a tener 2 slots compartidos en lugar de 5, como se observa en la figura **5-3**. Además, en MMF-SDN se consigue estabilizar el comportamiento de los nodos móviles después de la fase inicial de descubrimiento, donde el controlador conoce todos los nodos de entrada que utiliza el nodo móvil y tiene asignados los recursos en enrutamiento y TSCH para todas las rutas. En esta planificación, no hay variación del E2E delay, este será el mínimo (30 ms) sin importar el nodo de entrada, ya que el algoritmo de planificación considera el orden de las celdas actuales y planificadas anteriormente para optimizar el

E2E delay. La cantidad de celdas asignadas se reduce de 17 a 10 celdas, eliminando 3 slots compartidos y 4 celdas dedicadas en comparación con FTS-SDN.

En el slotframe utilizado con 13 slots y 4 canales se tienen 52 celdas disponibles, por lo tanto, la ocupación del slotframe para FTS-SDN es de 61.5 % debido a que los 5 slots compartidos representan una ocupación del 38.5 % al no poder planificar transmisiones sobre los 3 canales restantes, y las dedicadas 23 %. A diferencia de MMF-SDN donde la ocupación total es de 30.8 %, 15.4 % por los dos slots compartidos y 15.4 % por las 8 celdas dedicadas. Esta reducción del 30 % de ocupación se debe en mayor medida (23 %) a la reducción de los slots compartidos.

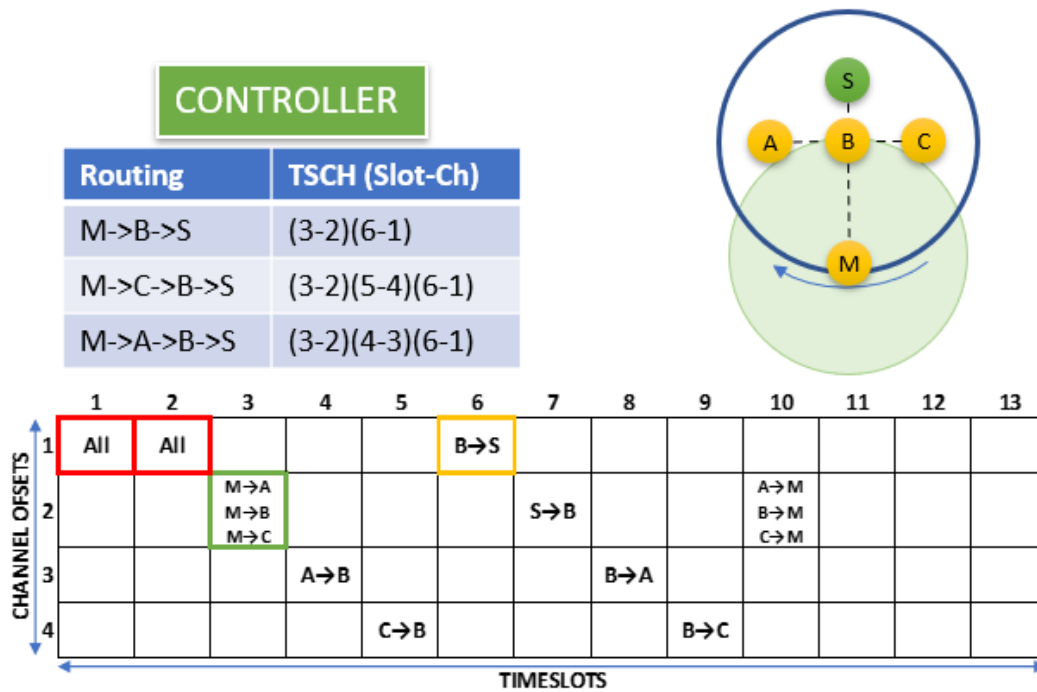
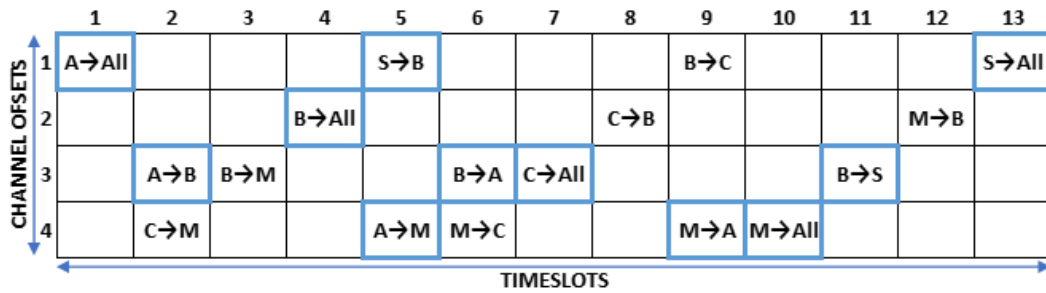


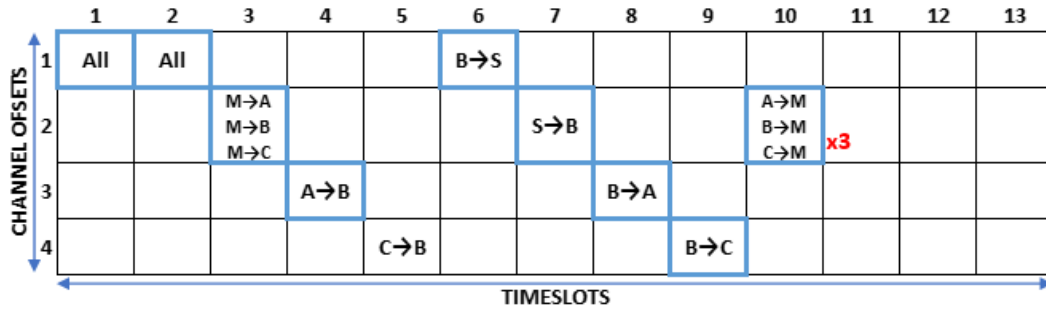
Figura 5-3: Topología de referencia y slotframe con MMF-SDN

El consumo de energía se puede estimar considerando la cantidad de estados de transmisión y recepción en cada uno de los nodos. Para los nodos de transmisión, se toma en cuenta el peor caso, que corresponde a la ruta con una mayor cantidad de saltos $M \rightarrow A \rightarrow B \rightarrow S$. La figura 5-4 muestra en color azul todas las celdas que deben transmitir para el slotframe de FTS-SDN en la figura 5-4(a) y para MMF-SDN en la figura 5-4(b). En ambos casos, se observa un total de 11 estados de transmisión por slotframe. Para FTS-SDN, el tráfico *uplink*, que va desde el nodo móvil al sink requiere 3 estados de transmisión los cuales están asignados en celdas de los slots 9, 2 y 11, al igual que para *downlink* en 3 celdas de los slots 5, 6 y 5. Los 5 estados de transmisión restantes se deben al uso de los slots compartidos en los slots 1, 4, 7, 10 y 13.

En el caso de MMF-SDN, se mantienen los mismos 3 estados de transmisión necesarios para el tráfico *uplink* en los slots 3, 4 y 6. En cambio, requiere una mayor cantidad de estados para el tráfico de *downlink*, ya que se mantienen los recursos asignados para cualquier posición del nodo móvil. Esto implica, utilizar los recursos para que el tráfico fluya desde el sink a los nodos A, B y C simultáneamente, por lo tanto, se requieren 6 estados de transmisión, planificados en los slots 7, 8, 9 y 10. Se puede observar que en este último slot se realizan 3 transmisiones al nodo móvil para garantizar la entrega de información independientemente de su posición. La optimización de los slots compartidos permite reducir a 2 este tipo de slots en MMF-SDN.



(a) FTS-SDN

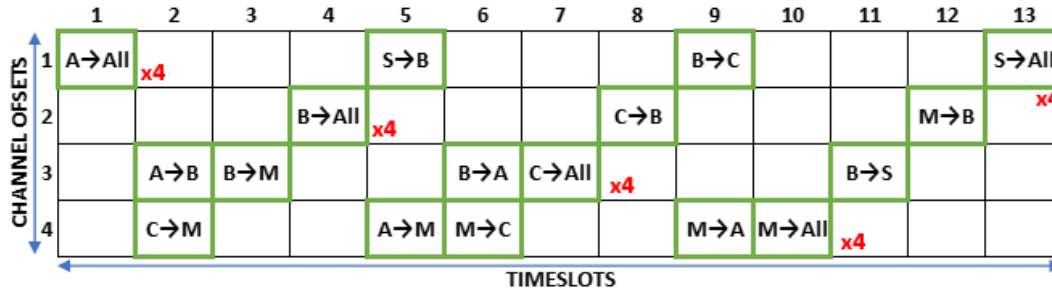


(b) MMF-SDN

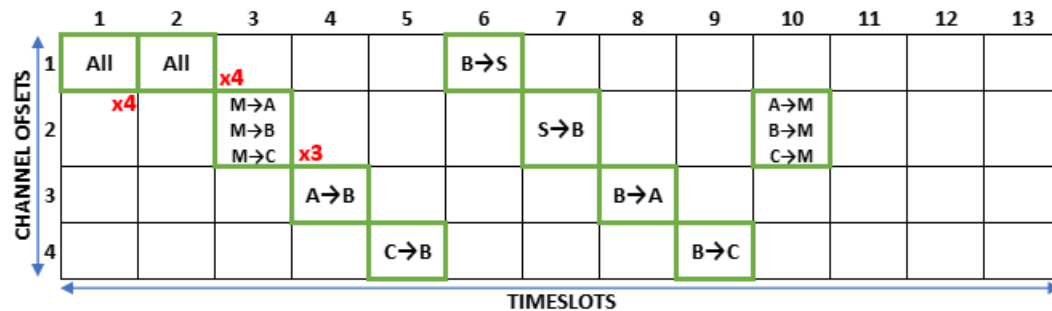
Figura 5-4: Número de estados de transmisión

En recepción se consideran todos los slots planificados, ya que a diferencia de la transmisión, los nodos deben escuchar siempre que estén planificados. Además, en los slots compartidos escucharán todos los nodos excepto el que transmite ($n - 1$), por lo que se multiplicará cada slot compartido por 4. En la figura 5-5(a) en total hay 32 estados de recepción por slotframe, y 18 en la figura 5-5 (b). Si se omiten los slots compartidos se tendrían 12 y 10 estados de recepción respectivamente, por lo que se observa claramente el efecto que tiene el uso particular de los slots compartidos en FTS-SDN. Por esto, la planificación con MMF consigue una reducción del 40% en el consumo de energía en recepción. La mayor parte se debe a la reducción en la cantidad de slots compartidos, ya que es un factor que implica un

consumo adicional en todos los nodos.



(a) FTS-SDN



(b) MMF-SDN

Figura 5-5: Número de estados de recepción

De acuerdo al análisis de tráfico de control y el uso de celdas multicast, se observa como la implementación FTS-SDN puede ser mejorada en aspectos de calidad de servicio (delay, jitter), consumo de energía y escalabilidad (longitud y ocupación del slotframe). Estos aspectos no eran objetivo de [62] por enfocarse en tener una baja pérdida de paquetes en el nodo móvil a través de una frecuencia muy alta de actualización. Esto implica asignar una cantidad muy alta de slots compartidos para asegurar que todos los paquetes de control lleguen al controlador. Los autores de FTS-SDN propusieron en [62] utilizar métodos de clusterización para solventar esta limitación. Por esto, el objetivo de MMF-SDN es limitar las constantes actualizaciones de topología debido al movimiento del nodo móvil. Además, se pretende ofrecer el mismo comportamiento determinista que tienen los nodos fijos.

En MMF-SDN aumentar el tamaño del slotframe no supone un aumento de los delays, ya que se implementa la repetición de los flujos dentro del slotframe. Las mejoras se observarán en los siguientes aspectos:

1. retardo extremo a extremo (E2E delay): Menor y determinista.
2. Tráfico de control independiente de la topología móvil: La cantidad de tráfico de control no depende del número de nodos móviles ni de su velocidad, ya que no se basa en

actualizaciones constantes.

3. Reducción del tráfico de control: Los cambios de planificación y enrutamiento se envían por flujo con un único paquete a todos los nodos de la ruta, en lugar de una planificación específica para cada nodo.
4. Planificación simplificada: Al no depender del tráfico de control, no es necesario asignar un slot compartido por nodo.
5. Mayor escalabilidad: Debido a la reducción de la longitud del slotframe y a la facilidad para modificar y añadir slots.
6. Menor consumo de energía: Debido a un menor número de slots compartidos.

5.3. Evaluación del rendimiento

En esta sección, se presentan los resultados experimentales para tres escenarios, siendo los dos primeros utilizados en FTS-SDN según [62]. Como representación más acorde a un entorno industrial, se ha incorporado un tercer escenario, visible en la figura 5-15. La evaluación de los resultados se realiza considerando el consumo de energía, el E2E delay y la congestión de la red. Para llevar a cabo los experimentos, se empleó el simulador Cooja de Contiki-NG y un conjunto de 500 paquetes de datos. En los escenarios 1 y 3, se estableció una frecuencia de envío de un paquete por segundo, mientras que en el escenario 2, la frecuencia fue de un paquete cada tres segundos. Estos parámetros se seleccionaron de manera que fueran suficientes para observar el comportamiento de los dos enfoques.

5.3.1. Escenario 1

La topología de este escenario se representa en la figura 5-1 [62]. El nodo móvil envía un paquete al sink cada segundo, mientras se desplaza describe una trayectoria circular de 16 m de radio a una velocidad de 1 m/s. En el centro de esta trayectoria está situado el nodo B, por lo que este siempre está dentro del área de cobertura del nodo M.

Parámetro	Valor
Número de paquetes	500
Frecuencia de envío	1 paquete/s
Velocidad del móvil	1 m/s, 2 m/s, 4 m/s
Cantidad de nodos	4 fijos y 1 móvil
Periodo de Beacon (B_p)	1 s, 4 s
Periodo de Reporte (R_p)	2 s, 8 s

Tabla 5-2: Parámetros usados en el escenario 1

El consumo de energía, depende de los estados que contenga la planificación TSCH. En la figura 5-4 y 5-5 se observa la cantidad de slots que se utilizan en transmisión y recepción para FTS-SDN y MMF-SDN. Las transmisiones no representan un consumo adicional de energía, ya que si un nodo no tiene datos que transmitir no se enciende la radio. Sin embargo, en las recepciones la radio debe encenderse un mínimo de 2.2 ms, tiempo por defecto para determinar que no hay tráfico que recibir. Cuando en un slot compartido ningún nodo tiene datos que transmitir, todos los nodos estarán en recepción durante estos 2.2 ms, lo que implica un consumo adicional de energía. Para el peor caso, donde todos los nodos están en recepción, la planificación propuesta con MMF-SDN tendrá 5 nodos en recepción en cada slot compartido.

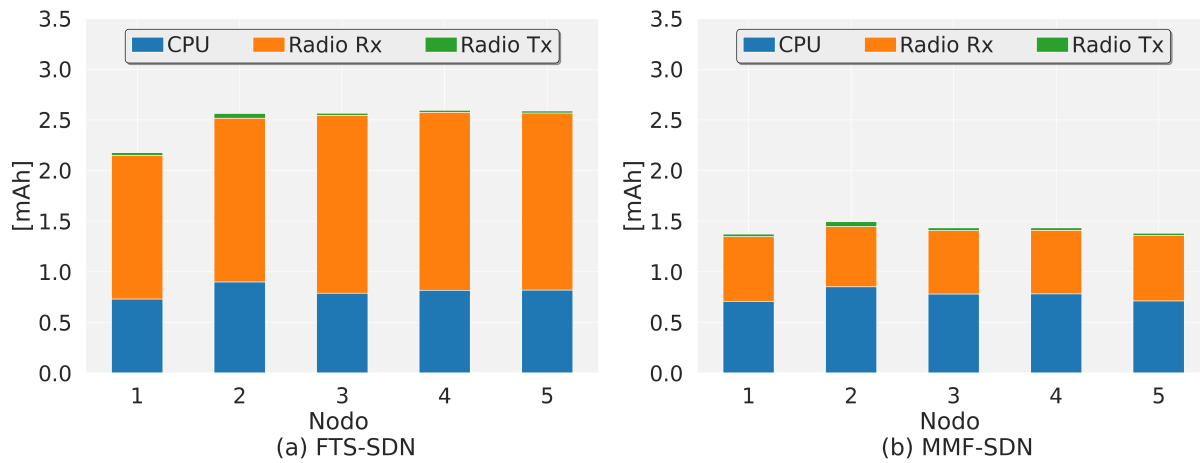


Figura 5-6: Consumo de energía por nodo en el escenario 1.

En el caso de FTS-SDN, 4 nodos estarán en recepción, ya que está definido uno de los nodos en transmisión. En total para el tráfico de control un slotframe tendrá 10 estados de recepción para la planificación MMF-SDN y 20 para la planificación de FTS-SDN. Esto implica una disminución del 50% del consumo de energía en recepción en el peor caso. Para comprobarlo se realiza la simulación de la topología de la figura 5-1 solo con tráfico de control durante una hora, para las dos planificaciones usando un $B_p = 1$ s y $R_p = 2$ s. El resultado se muestra en la figura 5-6, donde se observa el consumo de energía en cada nodo de acuerdo al uso de CPU, Radio Tx y Radio Rx para la misma topología. Para FTS-SDN en la figura 5-6(a), se usan 5 slots compartidos, el consumo en este caso está sobre los 2 mAh para todos los nodos y la mayor parte de esta energía se utiliza en la recepción. En la figura 5-6(b), MMF-SDN usa 2 slots compartidos, el consumo total se encuentra por debajo de los 1.5 mAh para todos los nodos y se observa una clara reducción de la energía destinada a recepción. Esta reducción es de un 62%, se produce al pasar de 1.66 mAh a 0.63 mAh para FTS-SDN en la figura 5-6(a) y para MMF-SDN en la figura 5-6(b) respectivamente. En los valores de CPU y Radio Tx no se aprecian diferencias considerables.

El funcionamiento que tienen los slots compartidos en FTS-SDN es muy específico, debido a que hay un único nodo transmisor para cada slot compartido. Esto requiere una etapa adicional para configurar en cada nodo los slots compartidos correspondientes. Con este método, la transmisión está asegurada, a diferencia de MMF-SDN donde múltiples nodos pueden transmitir en el mismo slot compartido, lo cual puede provocar colisiones. La figura 5-7 muestra la cantidad de paquetes de control enviados durante una hora de simulación usando $B_p = 1$ s y $R_p = 2$ s en los dos protocolos. Los valores del beacon de SDN WISE son levemente inferiores en MMF-SDN. Sin embargo, la cantidad de paquetes de reporte es superior y no coincide exactamente con la cantidad de ACKs. En MMF-SDN hay un 17% más de reportes que de ACKs. Esto se debe a las colisiones producidas en los slots compartidos, que afectan a todos los paquetes de control. Para el caso de FTS-SDN, los paquetes de reporte perdidos se deben a los cambios de padre del nodo móvil y corresponden a un 5% de los paquetes de reporte.

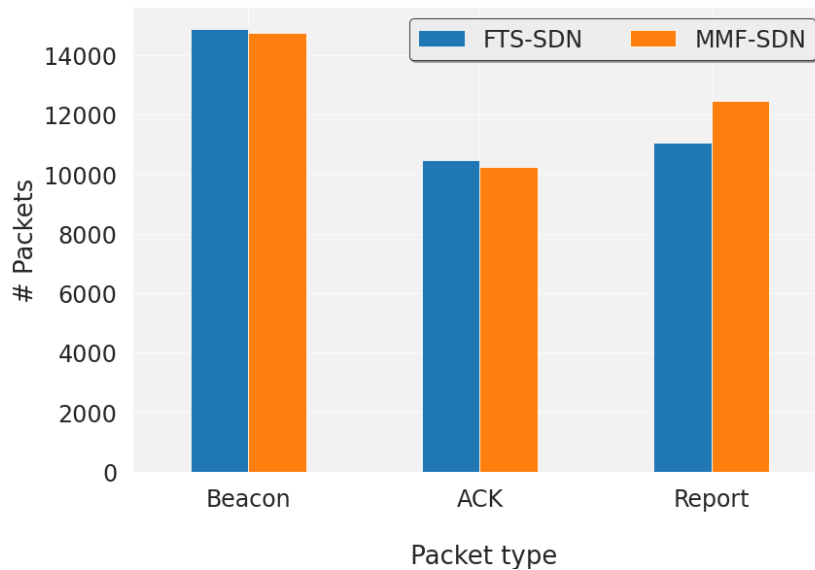


Figura 5-7: Número de paquetes de control enviados durante 1 hora de simulación

Las colisiones que se producen en los slots compartidos, no tienen un impacto considerable en el rendimiento de MMF-SDN, debido a que este, no necesita una actualización inmediata de la topología. Esta menor dependencia de los paquetes de control, también permite reducir la frecuencia de los paquetes de beacon y de reporte sin afectar al rendimiento. La figura 5-8 muestra el E2E delay cuando la velocidad de desplazamiento del nodo móvil es de 1 m/s, y se utilizan las planificaciones de la figura 5-4. En la figura 5-8(a) se observa el E2E delay producido por el algoritmo FTS-SDN con $B_p = 1$ s y $R_p = 2$ s y por el algoritmo MMF-SDN con $B_p = 4$ s y $R_p = 8$ s. En este caso, los dos protocolos tienen el rendimiento esperado según la planificación TSCH y un PDR de 100%. Sin embargo, la optimización de la planificación TSCH en MMF-SDN reduce significativamente el E2E delay. En FTS-SDN,

el E2E delay promedio es de 106 ms, mientras que en MMF-SDN se reduce a solo 30 ms, lo que representa una reducción del 70 %. Los periodos de beacon y reporte utilizados en MMF-SDN son 4 veces superiores a los usados por FTS-SDN, por lo que, con un número menor de paquetes de control se obtiene el mismo PDR.

El consumo de energía se representa en la figura 5-8(b). Este, en recepción se ha reducido un 50 % respecto al consumo de FTS-SDN, ya que la cantidad de slots compartidos es menor. Sin embargo, es mayor al consumo obtenido en la figura 5-6 debido a que los nodos estarán más tiempo en recepción, porque deben reenviar una cantidad menor de tráfico de control. Una menor cantidad de transmisiones reduce la carga de procesamiento lo que requiere un menor consumo por CPU. Con esto, se comprueba que es posible el uso de una menor cantidad de tráfico de control y que la pérdida de este tipo de paquetes no implica una pérdida de rendimiento.

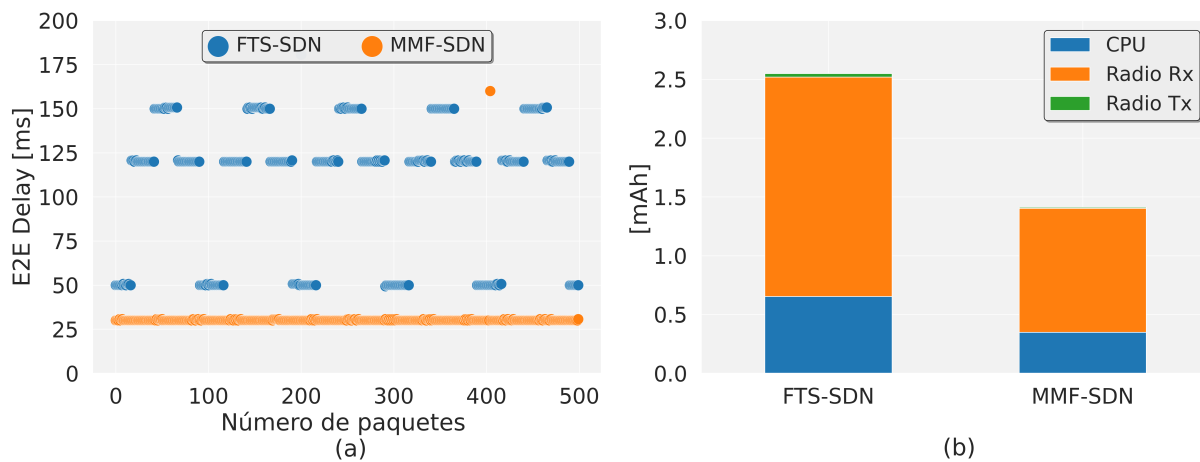


Figura 5-8: (a) E2E delay con $v = 1m/s$ para FTS-SDN y MMF-SDN, (b) Consumo medio de energía en el escenario 1.

Adaptarse a los cambios de la topología producidos por un nodo móvil es complejo, ya que la velocidad y trayectoria del nodo pueden cambiar. Por esto, se analiza el efecto de la velocidad en el rendimiento de MMF-SDN y FTS-SDN. Una alta dependencia del tráfico de control, implica que se puede producir un fallo en cualquier momento, a diferencia de la asignación de recursos acumulada, donde se consigue un comportamiento estable al conservar asignadas las rutas, además, esto también reduce la cantidad de configuraciones que debe enviar el controlador.

La figura 5-9 muestra el E2E delay obtenido con diferentes velocidades de desplazamiento en el nodo móvil. En la figura 5-9 (a) y (b) el móvil se desplaza a una velocidad de 1 m/s, y tanto FTS-SDN como MMF-SDN se adaptan al movimiento sin tener ningún tipo de pérdida.

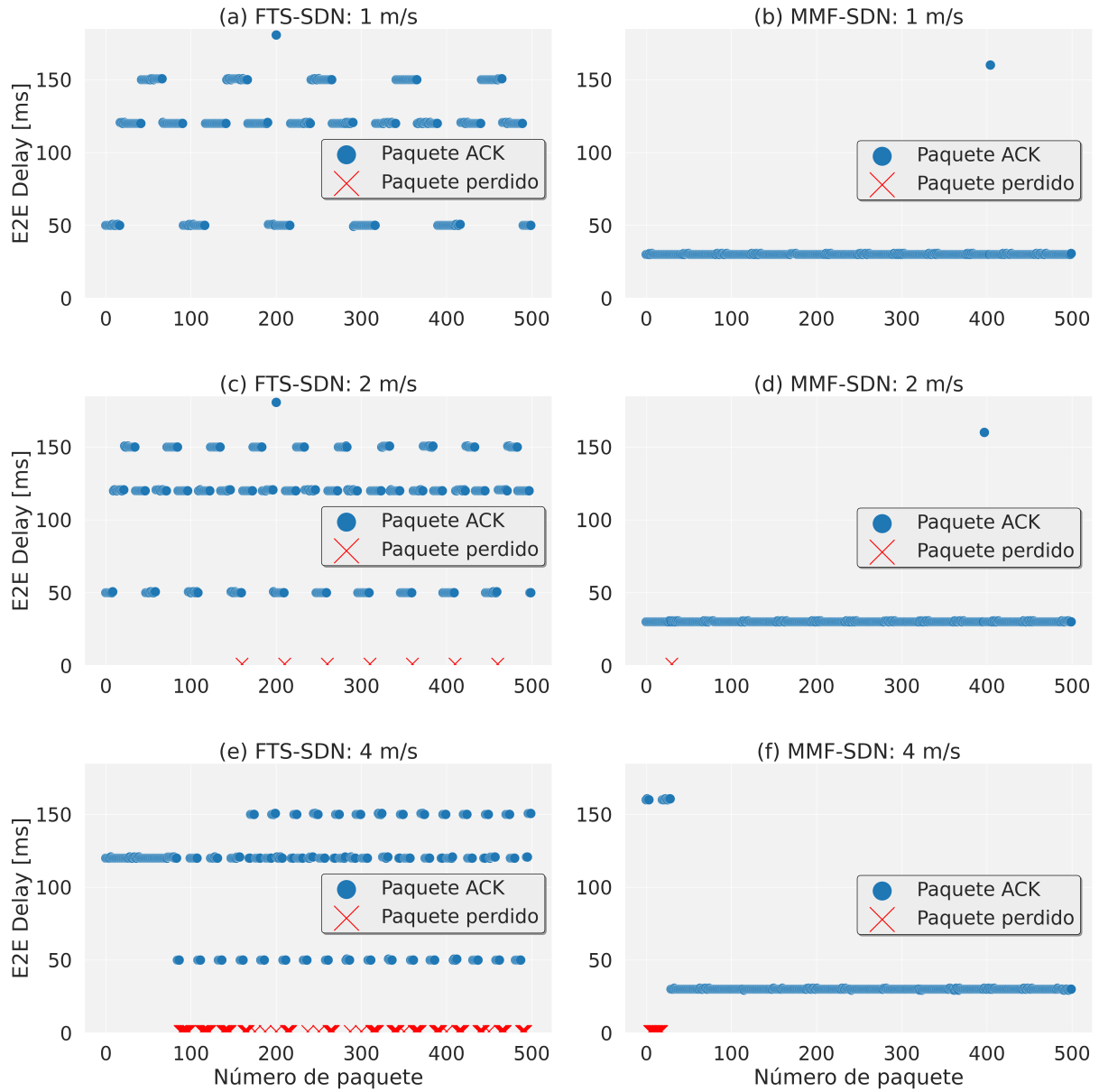


Figura 5-9: Comparación de MMF-SDN y FTS-SDN a distintas velocidades de los nodos móviles.

Al aumentar la velocidad a 2 m/s, como se muestra en la figura 5-9 (c) y (d), el protocolo FTS-SDN presenta un alto rendimiento en la transmisión de los primeros paquetes, pero después de 100 paquetes empieza a tener pérdidas a diferencia de MMF-SDN que las tiene en el primer cambio de padre y posteriormente permanece estable. Con una velocidad de desplazamiento alta (4 m/s), el controlador no logra actualizar las rutas con la suficiente rapidez y en el caso de FTS-SDN en la figura 5-9(e), se producen pérdidas constantes, concretamente un 20%. Por el contrario, con MMF-SDN la configuración de las rutas solo

afecta en los primeros paquetes, como se muestra en la figura 5-9(f). Después de esto, el comportamiento es totalmente estable, ya que no se requieren configuraciones adicionales, lo que permite un mejor rendimiento en régimen permanente.

La figura 5-10, resume el comportamiento de los dos protocolos para las tres velocidades de desplazamiento del nodo móvil. En la figura 5-10(a) el PDR para FTS-SDN se mantiene en 100 % para una velocidad de 1 m/s. Para las velocidades de desplazamiento de 2 m/s y 4 m/s se tiene 98.6 % y 76.6 %. Sin embargo, el PDR empieza a descender de forma constante. Debido a que el controlador no puede adaptarse a los cambios rápidos en la topología, no es posible conseguir un sistema estable. Por esto, las velocidades de 2 m/s y 4 m/s tienen una tendencia negativa, a mayor velocidad más notorio es este efecto. La figura 5-10(b) muestra el PDR para el protocolo MMF-SDN, al finalizar el envío de los 500 paquetes se tiene un PDR de 100 %, 99.8 % y 97 % respectivamente. Con la asignación de recursos acumulada se tienen pérdidas y un aumento del E2E delay en la fase inicial de descubrimiento mientras el controlador configura las entradas, pero después de esta asignación mantiene estable el rendimiento. Por esto en todos los casos la tendencia es positiva.

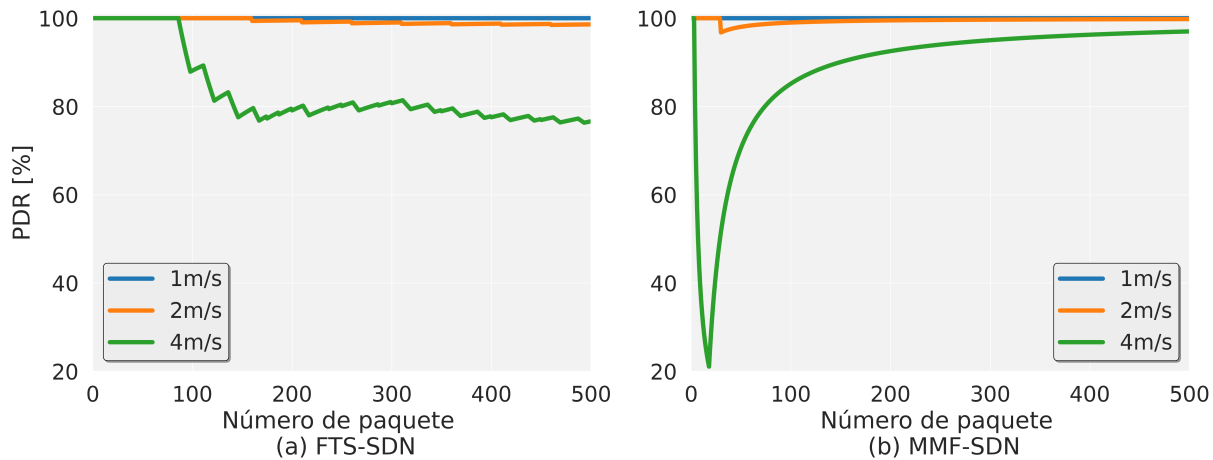


Figura 5-10: PDR con diferentes velocidades del nodo móvil.

5.3.2. Escenario 2

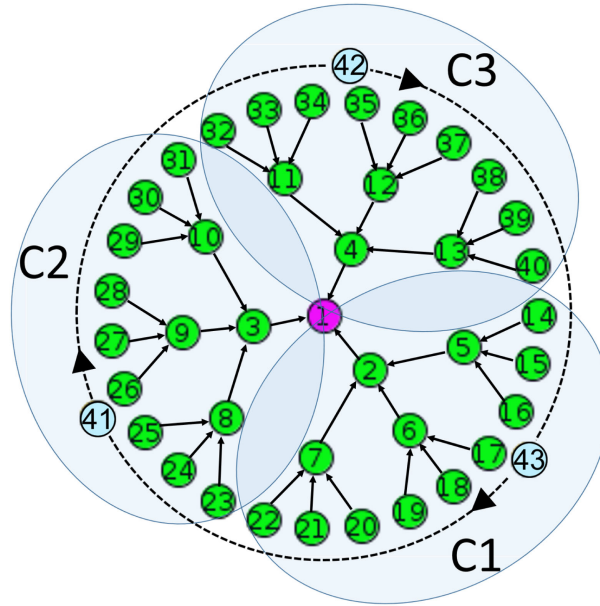


Figura 5-11: Topología con 43 nodos, 3 nodos móviles [62]

En esta sección se muestran los resultados obtenidos tras evaluar el escenario de la figura 5-11 [62]. Está compuesto por 43 nodos donde 3 de ellos son móviles (Nodos 41, 42 y 43) y envían paquetes al sink cada 3 segundos, se desplazan por la trayectoria circular a una velocidad de 1 m/s. A diferencia del escenario 1, donde el nodo móvil tenía una distancia fija a uno de sus padres (B), este escenario tiene cambios constantes de nodo padre y una mayor distancia al sink (Nodo 1). Por lo tanto, el impacto del tráfico de control será más significativo.

Parámetro	Valor
Número de paquetes	500
Frecuencia de envío	1 paquete/3 s
Velocidad del móvil	1 m/s
Cantidad de nodos	40 fijos y 3 móviles
Periodo de Beacon (B_p)	3 s, 5 s
Periodo de Reporte (R_p)	6 s, 10 s

Tabla 5-3: Parámetros utilizados en el escenario 2

El efecto de los periodos de beacon y reporte sobre la cantidad de tráfico de control se observa en la figura 5-12. Se ha utilizado una topología con n nodos y un máximo de 4 saltos al sink. La cantidad de nodos en cada salto se distribuye de forma similar a la que se muestra en la figura 5-11. La línea discontinua representa el límite del ancho de banda, que se logra

al tener todos los slots como shared de acuerdo a la ecuación (5-2), la cual para un $T = 1$ s es de 100 paquetes. Por esto, mientras más cerca esté la recta de tráfico de control del límite del ancho de banda, mayor será el porcentaje de ocupación del slotframe con slots compartidos y menor será la capacidad de la red para enviar flujos de datos.

La solución FTS-SDN necesita una alta frecuencia de actualización, por lo que se ha usado $B_p = 1$ s y $R_p = 2$ s. Estos parámetros se corresponden con la línea (Azul) que se acerca más rápido al límite del ancho de banda. La propuesta planteada, MMF-SDN, se basa en una asignación acumulada de recursos, por lo que no es necesario actualizar con un periodo tan bajo para determinar el padre actual del nodo móvil. Esto permite reducir el tráfico de control al utilizar periodos de beacon y reporte mayores, donde el tráfico de control se reduce desde el 50 % al duplicar estos periodos, hasta el 80 % para una configuración de $B_p = 5$ s y $R_p = 10$ s. Esta reducción en la cantidad de tráfico de control también permite que la red sea más escalable. Por eso, se observa como el límite para los valores de B_p y R_p usados por FTS-SDN está cerca de los 40 nodos, este puede ser extendido hasta los 80 nodos al duplicar los periodos de actualización donde se observa el límite para $B_p = 2$ s y $R_p = 4$ s.

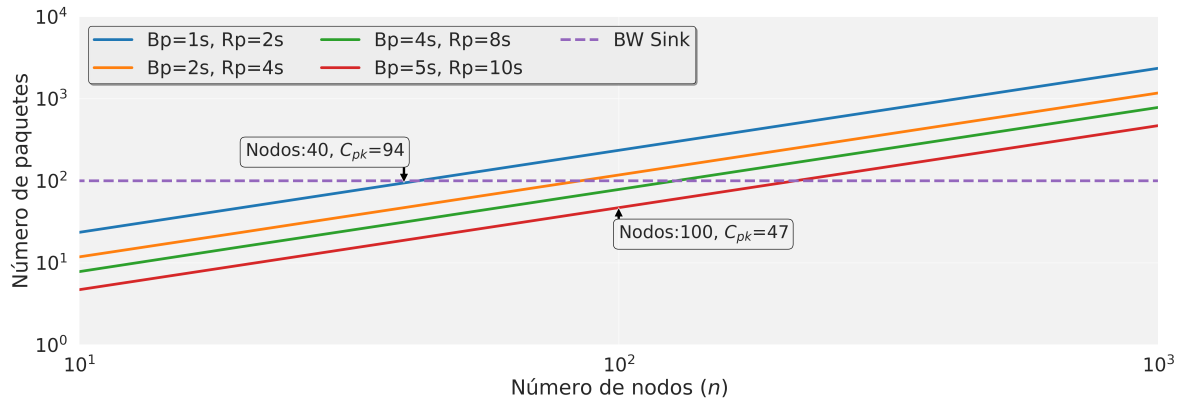


Figura 5-12: Total de paquetes de control en función del número de nodos para $T = 10$ s, con diferentes periodos de beacon (B_p) y periodos de reporte (R_p).

En la figura 5-13 se compara la cantidad de slots compartidos utilizados por FTS-SDN en función del número de nodos, con la cantidad óptima de slots compartidos usada por MMF-SDN para 4 configuraciones diferentes de B_p y R_p . En todos los casos, la cantidad de slots compartidos es muy inferior a la regla de un slot compartido por nodo usado en FTS-SDN, ya que utilizar un número reducido de slots compartidos permite reducir el tamaño del slotframe.

Por ejemplo, para 40 nodos según se muestra en la figura 5-13, FTS-SDN utiliza 40 slots compartidos, mientras MMF-SDN puede utilizar 13 slots. Esto implica un aumento en el ancho de banda, incluso si se usarán slotframes de diferentes tamaños para igualar la cantidad de slots disponibles, ya que por el funcionamiento cíclico de los slotframes, la distancia entre

los slots compartidos será menor. Por eso, en MMF-SDN se obtienen planificaciones con un mejor aprovechamiento de los recursos de la red, incluso en condiciones con una alta frecuencia de actualización.

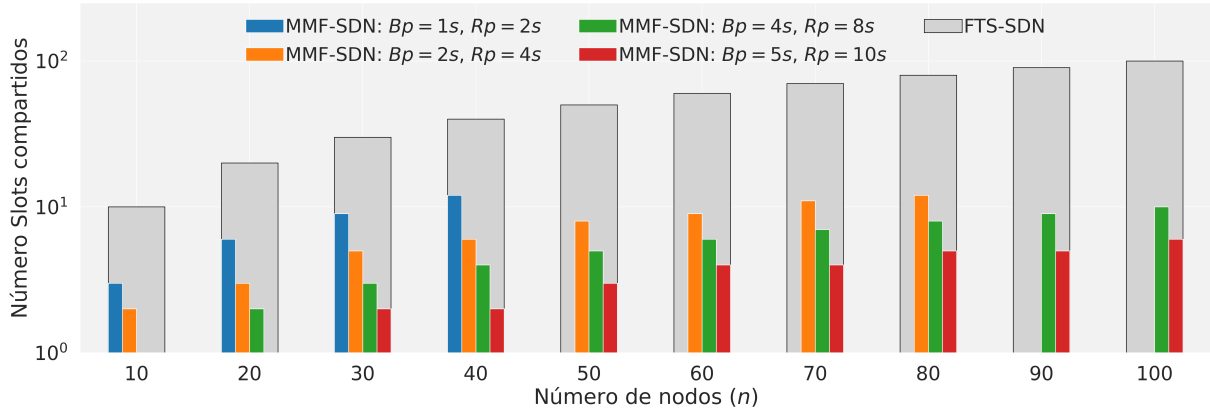


Figura 5-13: Número de slots compartidos a utilizar en función del tráfico de control.

De aquí que, al aumentar el número de nodos se observe también como aumenta la diferencia de rendimiento, ya que la cantidad de slots compartidos asignada en FTS-SDN no se adapta a las necesidades de ancho de banda de la red. Por ejemplo, la topología mostrada En la figura 5-11 [62], está compuesta por 40 nodos fijos (**N**), 3 móviles (**M**) y un máximo de 4 saltos. En el 4^o salto hay 3 nodos móviles, en el 3^o salto hay 27 nodos fijos, en el 2^o salto 9 nodos fijos y en el 1^o salto 3 nodos fijos. Por esto, para los nodos fijos se necesitan $N-1$ celdas de uplink y $N-1$ celdas de downlink, lo que equivale a 39 en ambos casos. Además, los nodos móviles necesitan un enlace con cada nodo exterior. Esto son 27 celdas por cada nodo móvil para uplink, más otras 27 celdas por nodo móvil para downlink, es decir, 54 celdas por nodo móvil. Por lo tanto, el slotframe debe tener asignadas 162 celdas para los nodos móviles y 78 celdas para los nodos fijos, es decir, un total de 240 celdas. Este valor está definido en [62] como el número mínimo de enlaces unicast, que equivale a dos veces la cantidad de nodos fijos por la cantidad de nodos móviles ($2 \cdot N \cdot M$). Para FTS-SDN, se utilizó un slotframe de 91 slots y 4 canales. Al retirar los 43 slots que se asignan a los slots compartidos, quedan 48 slots disponibles. Usando 4 canales hay en total 192 celdas disponibles, donde no es posible ubicar las 240 celdas necesarias. Al utilizar el enfoque multicast de MMF-SDN, no es necesario utilizar una celda diferente para cada uno de los 27 receptores del nodo móvil. De aquí que las 162 celdas utilizadas para los enlaces unicast de los nodos móviles, puedan ser reducidas a solo 6, 2 por nodo móvil. Por lo tanto, usando los mismos parámetros, el slotframe pasaría de una ocupación del 100% al 70%. Al evaluar la cantidad de tráfico de control que genera la topología de la figura 5-11, usando la ecuación (5-1) con $Bp = 1$ s, $Rp = 2$ s se tienen 1000 paquetes de control en un periodo T igual a 10 segundos:

$$C_{pk} = 43 \frac{T}{Bp} + \frac{T}{Rp} ((3 \cdot 4) + (27 \cdot 3) + (9 \cdot 2) + (3 \cdot 1)) = 1000$$

El ancho de banda necesario para enviar esta cantidad de paquetes de control, solo se consigue al usar todos los slots del slotframe como slots compartidos, es decir, según la ecuación (5-2) $\frac{Sh_{SF}}{SF_{sz}} = 1$. Por lo que el periodo de beacon y reporte debe aumentarse. En este escenario, se utiliza un slotframe con el mismo tamaño que en las pruebas de FTS-SDN [62], 91 slots y 4 canales. La cantidad óptima de slots compartidos para esta longitud del slotframe es de 30 slots, utilizando $B_p = 3$ s y $R_p = 6$ s.

Bajo estas condiciones MMF-SDN tiene un mejor comportamiento en E2E delay y determinismo, como se aprecia en la figura **5-14(a)**. MMF-SDN puede garantizar un delay por debajo de los 200 ms en el 99% de los paquetes enviados durante la fase inicial de descubrimiento. Esta coincide con una rotación completa del nodo móvil, que para la velocidad y longitud de trayectoria en este escenario está cerca de los 200 paquetes. En el mismo escenario para FTS-SDN en la figura **5-14(b)** se tiene una alta variabilidad en los E2E delays y pérdidas durante toda la simulación. Los resultados obtenidos en [62], presentan delays mayores a los 1.5 s, y cuando se segmenta esta red en clusters los autores logran reducir estos delays cerca de los 500 ms. Esta diferencia de rendimiento tan significativa se debe a que en FTS-SDN se produce una saturación de los nodos centrales, ya que estos deben reenviar el tráfico de control de los nodos hojas, y al tener un solo slot compartido por cada slotframe el ancho de banda se reduce al aumentar la longitud del slotframe. Durante la simulación hay un promedio de 6 paquetes en el buffer de los nodos centrales (2, 3 y 4) para la implementación de FTS-SDN con $B_p = 1$ s y $R_p = 2$ s. Se tiene por ello, un desfase entre la topología que tiene el controlador y la real, lo que afecta a la fiabilidad de las rutas. Por esto, se utiliza $B_p = 3$ s y $R_p = 6$ s donde se reduce a 2 paquetes promedio en buffer para FTS-SDN y 1 paquete para MMF-SDN. Este último logra tener una menor cantidad de paquetes en buffer, debido a que los nodos tienen una mayor cantidad de slots compartidos disponibles por slotframe.

5.3.3. Escenario 3

En este caso se simula un escenario característico del sector industrial, donde se tienen dos AGV que cubren diferentes rutas dentro de la planta. En total son 27 nodos, 25 fijos y 2 móviles, todos ellos con un rango de cobertura de 20 m. Los nodos móviles están situados en los AGV que permanecen en continuo movimiento a una velocidad de 1.5 m/s. Estos nodos envían un paquete de datos al sink cada segundo. Tal y como muestra la figura **5-15**, los nodos móviles se desplazan por los dos pasillos centrales, uno en dirección horizontal (violeta) a lo largo de 64 m, y el otro en dirección vertical (verde), con un desplazamiento de 49 m. El sink se ha situado en la parte superior del pasillo central vertical. De acuerdo al

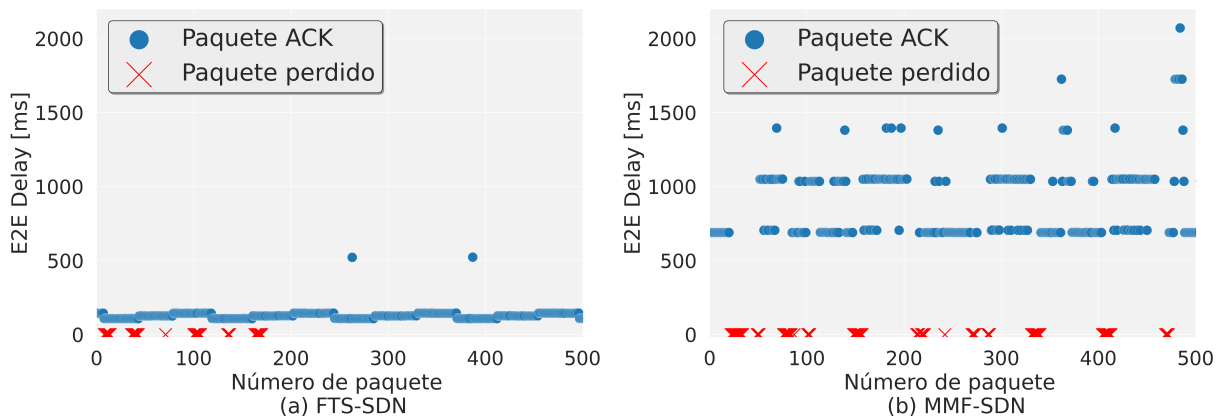


Figura 5-14: E2E delay en topología con 43 nodos

radio de cobertura, el tráfico de toda la WSN llega a este a través de los 4 nodos ubicados en la parte superior del pasillo central.

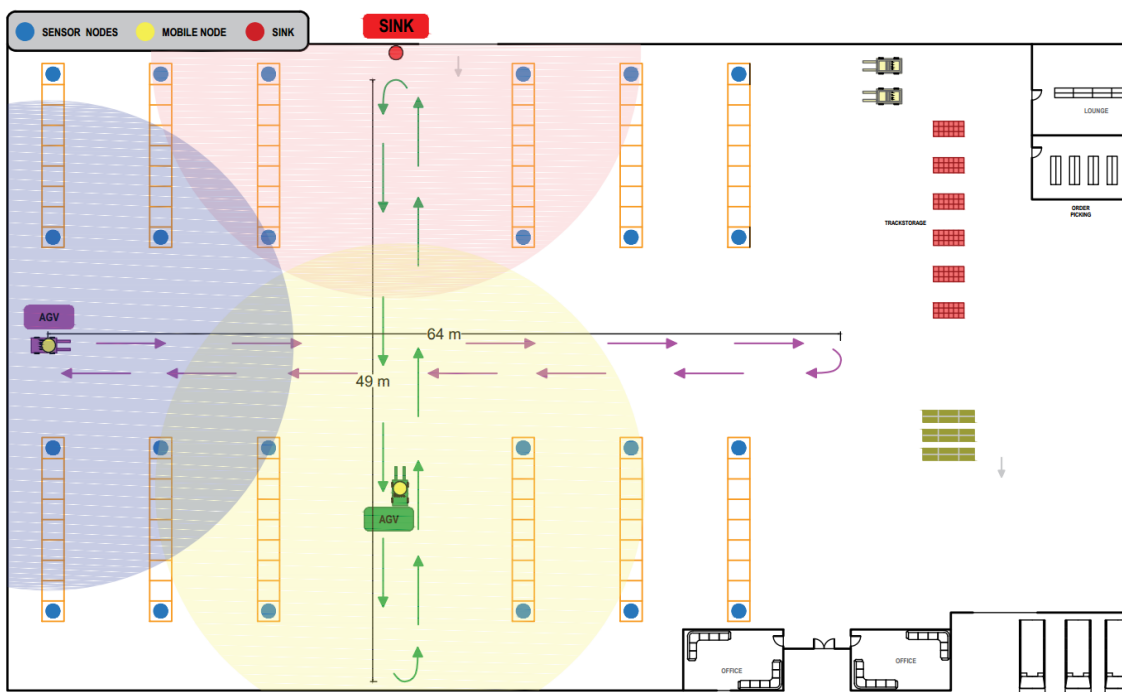


Figura 5-15: Escenario en un entorno industrial con 25 nodos fijos y 2 móviles.

Respecto a los dos escenarios anteriores, se aumenta la complejidad debido al desplazamiento de los nodos móviles por el centro de la nave industrial utilizada como escenario, lo que produce cambios en una mayor cantidad de nodos de forma simultánea.

Parámetro	Valor
Número de paquetes	3600
Frecuencia de envío	1 paquete/s
Velocidad del móvil	1.5 m/s
Cantidad de nodos	25 fijos y 2 móviles
Periodo de Beacon (B_p)	1 s
Periodo de Reporte (R_p)	2 s

Tabla 5-4: Parámetros utilizados en el escenario 3

La figura 5-16 muestra los E2E delay obtenidos utilizando los dos enfoques durante la fase inicial. En la figura 5-16(a) se aprecia como MMF-SDN logra tener el E2E delay que apenas supera los 300 ms tanto para el movimiento vertical como para el horizontal. Entre los 50 y 150 paquetes se tiene un delay mayor. Esto se debe a que en los nodos fijos solo hay un enlace dedicado para enviar el tráfico de datos al sink. Por esto, el E2E delay se aumenta cuando las rutas de los dos nodos móviles coinciden y uno de los paquetes debe permanecer en cola. Para MMF-SDN la pérdida de paquetes se produce solo durante la fase inicial de descubrimiento, cuando no están asignados los recursos. En la figura 5-16(b) se muestran los resultados utilizando la implementación de FTS-SDN. Para el movimiento horizontal el E2E delay puede mantenerse por debajo de los 200 ms, sin embargo, para el desplazamiento vertical existen diferentes niveles en el E2E delay hasta un máximo de 800 ms. En FTS-SDN la pérdida de paquetes puede producirse en cualquier instante de la prueba. Los delays y las pérdidas son efecto del encolamiento que se produce.

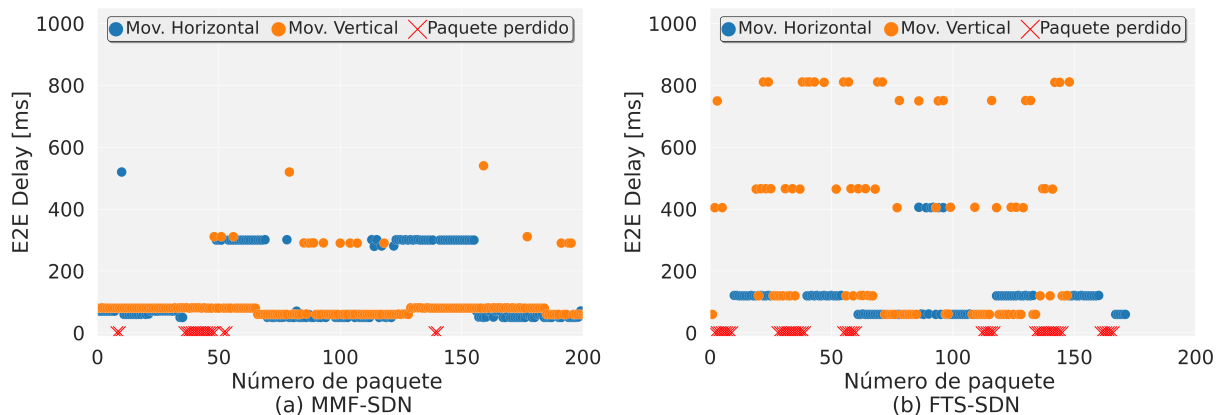


Figura 5-16: E2E delay para el movimiento horizontal y vertical del escenario 3.

La pérdida de paquetes en todo el recorrido se representa en la figura 5-17(a). MMF-SDN no tiene pérdidas en el movimiento vertical, es en el horizontal donde se producen pérdidas en la fase inicial de descubrimiento, concretamente durante el envío de los primeros 50 paquetes. Durante este tiempo, es cuando se produce el primer recorrido de la trayectoria y las pérdidas se deben al retraso en la asignación de los recursos. Después de esto, se observa un crecimiento constante en el PDR.

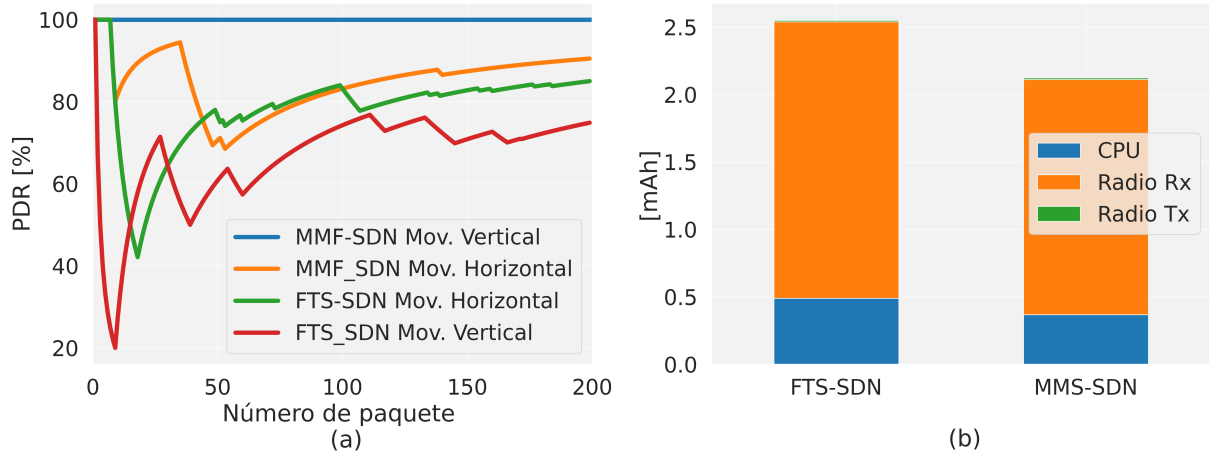


Figura 5-17: Comparación entre MMF-SDN y FTS-SDN Escenario 3.

Para FTS-SDN la probabilidad de pérdidas es igual en cualquier instante de tiempo, esto se debe a la saturación de los nodos más cercanos al sink, que envían el tráfico de control de todos los nodos hoja. Cuando se supera el ancho de banda los paquetes son almacenados en el buffer y son enviados cuando vuelve a haber recursos disponibles. Esto incrementa el desfase entre la topología real y la topología que utiliza el controlador para tomar las decisiones. Por lo tanto, las reglas que envíe este a los nodos móviles producirán pérdidas. En la figura 5-17(b) se observa el consumo de energía de los dos enfoques, en este caso la diferencia entre ambas es menor que la obtenida en los escenarios 1 y 2. Esto se debe a que en MMF-SDN por la naturaleza multicast se envía tráfico duplicado cuando dos receptores multicast reciben el paquete del nodo móvil. Es posible limitar la cantidad de paquetes duplicados con condiciones de RSSI sobre los paquetes multicast para acotar la zona de recepción, utilizando los números de secuencia para que un nodo descarte los paquetes repetidos que reciba o implementando aplicaciones en el controlador que permitan depurar las rutas sobre las que se reciben paquetes duplicados. Aun así, el incremento en el consumo de energía producido por estos paquetes, no es suficiente para superar el consumo que se produce en FTS-SDN.

Para comprobar los efectos de la estabilidad obtenida en MMF-SDN, se realiza una prueba durante una hora, donde se envía un total de 3.600 paquetes. La figura 5-18(a) muestra el E2E delay para los tipos de trayectorias del escenario 3. Se observan claramente dos

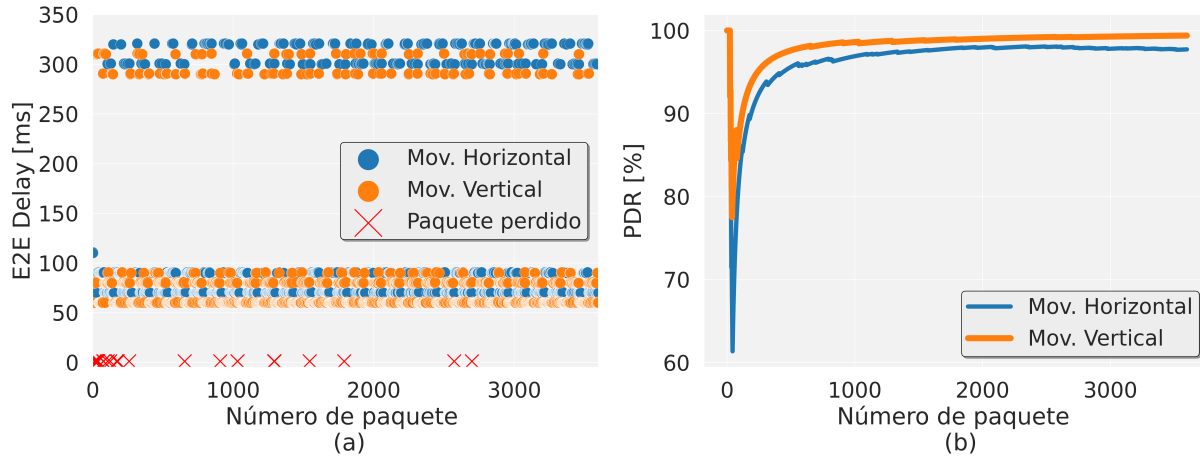


Figura 5-18: Estabilidad de MMF-SDN en el escenario 3.

agrupaciones, una por debajo de 100 ms y otra debajo de los 330 ms con una menor densidad. Estos son paquetes que se almacenaron en cola debido a que las rutas de los dos nodos móviles coincidieron, por lo tanto, el retardo corresponde con un slotframe más. En cada una de las agrupaciones se observan 4 bandas de puntos separadas entre ellas un slot (10 ms). Estas corresponden a cada uno de los nodos con conexión directa al sink. En la figura 5-18(b) se observan en ambos casos pérdidas iniciales de paquetes. Después se observa una estabilidad que permite obtener un PDR superior al 95 %. El movimiento vertical tiene una cantidad menor de pérdidas debido a que en su desplazamiento puede tener dos rutas independientes al sink. Por el envío en multicast el tráfico llega duplicado al sink, lo que permite tener un PDR cercano al 100 %.

5.4. Conclusiones

Durante este capítulo se ha demostrado como, en el contexto de los entornos industriales, las redes IWSN deben ser capaces de adaptarse rápidamente a los cambios físicos y lógicos, manteniendo al mismo tiempo un alto nivel de determinismo y calidad de servicio. Sin embargo, aumentar de forma masiva los dispositivos de digitalización lleva a situaciones donde los elementos se desplazan y producen cambios constantes en la topología. Por ello, para satisfacer estas necesidades, se ha desarrollado MMF-SDN, un protocolo de gestión de recursos en IWSN definido por software que permite incorporar eficientemente nodos en movimiento. Este protocolo supera las dificultades causadas por la falta de convergencia en el protocolo de enrutamiento cuando se produce un cambio de nodo padre, por medio de la asignación de recursos multicast, donde el controlador aprende los desplazamientos y al mismo tiempo garantiza parámetros estrictos de calidad de servicio.

Comparado con otro protocolo SDN, MMF-SDN presenta una menor dependencia del tráfico de control, lo que permite reducir este tipo de tráfico en un 50 % en comparación con FTS-SDN. Además, el uso optimizado de slots compartidos reduce el consumo de energía en recepción hasta en un 50 %. La asignación de recursos multicast acumulados utilizados por MMF-SDN permite estabilizar la red con nodos móviles, a diferencia de los enfoques basados en convergencia como FTS-SDN, que requieren una respuesta del controlador ante cada cambio en la topología.

La transmisión multicast en el nodo móvil elimina la dificultad de cambiar de padre que ocurre en FTS-SDN, de modo que el cambio de padre no produce un retardo adicional. Esto permite reducir el retardo de extremo a extremo en un 70 % en los escenarios descritos. Además, reduce la ocupación y longitud del slotframe en más del 30 %, ya que para cada nodo móvil se utilizan las mismas celdas de transmisión y recepción en todos los nodos padre.

Aunque el uso convencional de slots compartidos aumenta la pérdida de paquetes de control y la cantidad total de tráfico debido a las retransmisiones, no tiene un impacto significativo en el rendimiento de MMF-SDN. Esto se debe a que este protocolo no requiere una actualización rápida de la posición móvil, dado que los recursos permanecen asignados en cada una de las posibles entradas.

Capítulo 6

Escalabilidad en SD-IWSN

6.1. Introducción

Como se ha demostrado durante los capítulos anteriores, el paradigma de las redes definidas por software en las IWSN (SD-IWSN) [50] [61] [82] permite ir más allá de la centralización de protocolos, al extraer completamente el plano de control de los equipos de red y centralizar todos los procesos en un dispositivo común que tiene un conocimiento global de la red. Esta centralización completa permite la ejecución de procesos más exigentes en dispositivos externos a la propia red IWSN, empleando una mayor cantidad de recursos y permitiendo tener una administración de la IWSN con un alto nivel de detalle, mejorando la calidad de servicio y la flexibilidad [6]. Para conseguir este conocimiento global es necesario el tráfico de control, donde cada nodo debe enviar información de su entorno al controlador SDN.

Sin embargo, este flujo de tráfico de control es el mayor inconveniente de las SD-IWSN, ya que, combinado con las limitaciones de ancho de banda, produce una acumulación de tráfico de control en los nodos más cercanos al sink. A medida que aumenta la cantidad de nodos de la red, este efecto es más notable y tendrá un impacto en el rendimiento de la red, debido a que los nodos, además del tráfico de control, deben reenviar el tráfico de datos. Estos dos tipos de tráfico se agregarán a través de los diferentes saltos de la IWSN hasta saturar las colas y producir pérdida de paquetes. Dado que el sink es el único punto que conecta la IWSN con el controlador SDN, será el primer nodo en experimentar esta congestión, lo que limita la cantidad total de nodos en la red.

Por ello, en este capítulo se proponen e implementan mecanismos que permiten a las SD-IWSN escalar manteniendo las ventajas de flexibilidad y garantías de calidad de servicio que ofrece la combinación de SDN y TSCH. Entre estos mecanismos se encuentra la implementación de sinks con múltiples interfaces radio, denominados *sink virtuales*, que gracias al control que ejerce el controlador SDN, las múltiples interfaces radio se comportan como

un único dispositivo sink, consiguiendo aumentar considerablemente el ancho de banda de la red IWSN. Finalmente, para conseguir aprovechar completamente el espectro radioeléctrico orientado a mejorar la escalabilidad de estas redes, se han desplegado múltiples IWSN basadas en el concepto de sink virtual, integradas a través de la red SDN cableada y orquestadas desde el controlador SDN.

6.1.1. Análisis de escalabilidad en WSN

La escalabilidad en las IWSN es un tema ampliamente estudiado, debido a su naturaleza inalámbrica y funcionamiento autónomo, que les permite adaptarse a cambios en la topología física. Estas redes han evolucionado para adaptarse a diferentes casos de uso, hasta el punto de ser compatibles con entornos más críticos, en los que es necesario garantizar parámetros de QoS como el retardo, el *Packet Delivery Ratio* (PDR) y el *Deadline Satisfaction Ratio* (DSR). Sin embargo, garantizar estos aspectos de QoS manteniendo la escalabilidad sigue siendo un reto en este tipo de redes [41].

Las redes de sensores más tradicionales utilizan el protocolo CSMA como protocolo de acceso al medio, realizando un acceso aleatorio al medio compartido, lo que conlleva la posibilidad de colisiones e interferencias. En general, la escalabilidad de este tipo de redes *ad-hoc* se aborda con soluciones basadas en el enrutamiento y la clusterización, como proponen algunos trabajos como [83] [84] [85] [86] y [87].

Los clusters son divisiones jerárquicas de la red por grupos de nodos; en cada grupo hay un nodo principal que se encarga de recibir la información de los demás nodos del cluster, agruparla y enviarla al sink [41]. Esto reduce el número de transmisiones necesarias para enviar la información, lo que se traduce en un menor consumo de energía. Algunas soluciones, como [84], utilizan un algoritmo distribuido para elegir el nodo principal del clúster. Este nodo será un elemento crítico en el consumo de energía debido a las retransmisiones de mensajes de su clúster. Para mitigar este problema, el algoritmo incluye un sistema de rotación de los nodos principales. Otros estudios abordan el mismo problema de consumo energético de estos nodos principales del cluster [87], en los que los autores proponen diferentes algoritmos para mejorar el consumo energético en redes más densas y con enlaces más largos.

Otra alternativa para mejorar la escalabilidad de las redes malladas sin que ello repercuta negativamente en la calidad de servicio, es aumentar el número de sinks, [88] en el que se simulan redes con un máximo de 64 nodos. Al aumentar el número de sinks, el throughput de la red se mantiene estable, sin perjudicar el PDR en el conjunto de la red. También se han realizado un análisis sobre la coexistencia de múltiples redes TSCH asíncronas [89], permitiendo el despliegue de un número más significativo de nodos pero requiriendo mecanismos centralizados para mitigar las interferencias entre redes TSCH.

Artículo	Número de nodos	Simulación	CSMA	TSCH	SDN	Temas
EESRA: Energy Efficient Scalable Routing Algorithm for Wireless Sensor Networks [84]	Simulación: 100, 200, 300, 400	Matlab	✓	✓	✗	Algoritmos de enrutamiento y clusterización
An Effective Relay Node Selection Technique for Energy Efficient WSN-Assisted IoT [85]	Simulación: 1000, 1500	Matlab	✓		✗	Algoritmos de enrutamiento y clusterización.
A Platform for Evaluating Clustering Strategies in Mobile IEEE 802.15.4-TSCH networks [86]	Simulación: 15 nodos/cluster	Cooja		✓	✗	Algoritmos de enrutamiento y clusterización
Hybrid energy-efficient multi-path routing for wireless sensor networks [87]	Simulación: 500 nodos	Matlab			✗	Algoritmos de enrutamiento y clusterización
Network Performance Enhancement of Multi-sink Enabled Low Power Lossy Networks in SDN Based Internet of Things [88]	Simulación: 16, 32, 48, 64	Cooja	✓		✓	Posicionamiento en entornos multi-sink
Coexistence analysis of multiple asynchronous IEEE 802.15.4 TSCH-based networks [89]	Simulación: Up to 60 networks	Cooja, Matlab		✓	✗	Redes TSCH Múltiples
The Cost of Software-Defining Things: A Scalability Study of Software-Defined Sensor Networks [90]	Simulación: 289 Testbed: 10	Cooja	✓		✓	Controller positioning and multi-sink
Performance Analysis of Software-Defined Multihop Wireless Sensor Networks [91]	Simulación: 50, 100	Matlab	✓		✓	Consumo de energía en redes SD-WSN
A Review of the Control Plane Scalability Approaches in Software Defined Networking [92]	NA	NA			✓	Escalabilidad en SDN.
An SDN approach to route massive data flows of sensor networks [93]	Simulación: 1000	VMWare VSphere	NA	NA	✓	Algoritmos de clusterización
SYNCOP: An evolutionary multi-objective placement of SDN controllers for optimizing cost and network performance in WSNs [94]	Simulación: 100, 150, 200, 300	Java	NA	NA	✓	Posicionamiento del Controlador SDN con enfoque a la escalabilidad
A distributed density optimized scheduling function for IEEE 802.15.4e TSCH networks [95]	Simulación: 10 : 1500	6TiSCH Simulator		✓	✗	Planificación TSCH en WSN de alta densidad
Decentralized broadcast-based scheduling for dense multi-hop TSCH networks [96]	Simulación: 1 : 1500 nodos	6TiSCH Simulator		✓	✗	Optimización de la planificación TSCH
A distributed algorithm to schedule TSCH links under the SINR model [97]	Simulación: 300 nodos	Sinalgo		✓	✗	Optimización de la planificación TSCH
DIGEST: a decentralized divergecast scheduling algorithm for IEEE 802.15.4e TSCH in the internet of things [60]	Simulación: 30,50,70	Matlab		✓	✗	Optimización de la planificación TSCH
Esta tesis	Simulación: 320, 1152* Testbed: 18	Cooja		✓	✓	Multi-sink, Multi-radio y Múltiples TSCH IWSN

NA: No disponible * : Máximo permitido con QoS

Tabla 6-1: Comparativa de estudios sobre escalabilidad en IWSN

Sin embargo, es difícil garantizar los parámetros de calidad de servicio, ya que la probabilidad de colisiones e interferencias aumenta directamente con el número de nodos. Por ello, para entornos más críticos, en los que deben garantizarse las transmisiones y los parámetros de QoS, el protocolo CSMA ha sido sustituido por protocolos basados en TDMA, como TSCH. Este protocolo permite la sincronización de todos los nodos de la red y la asignación de timeslots a cada nodo para la transmisión o recepción de datos [9]. De este modo, se evitan las colisiones que se producen en CSMA cuando varios nodos transmiten en instantes de tiempo similares. Además, TSCH utiliza múltiples canales, lo que permite el salto de frecuencia para cada transmisión, reduciendo las interferencias y el desvanecimiento del camino [98]. La escalabilidad de TSCH es un tema que ha recibido una atención limitada en la literatura actual [99]. Esto se debe a que TSCH se ha desarrollado principalmente para aplicaciones industriales, donde tradicionalmente el número de nodos ha sido más limitado. Sin embargo, las necesidades actuales de digitalización requieren el despliegue de redes de sensores masivas en estos entornos, lo que plantea nuevos desafíos para la escalabilidad de TSCH, ya que además, se deben mantener altos niveles de calidad de servicio (QoS). Una comparativa de estos enfoques de escalabilidad puede observarse en la tabla 6.1, donde se muestran los diferentes protocolos utilizados y la cantidad de nodos con las que se llevan a cabo las pruebas.

6.1.2. Análisis del tráfico de control

Como se ha mencionado en los diferentes capítulos, las redes SDN dependen en gran medida de la información de control que reciben de los nodos, debido a que estos recogen información de su entorno y la envían al controlador SDN. Esta información es utilizada por el controlador SDN para construir una representación global de la topología, combinando la información de múltiples fuentes. Sin embargo, debido a las limitaciones de ancho de banda en las IWSN, la cantidad total de paquetes de control debe estar optimizada, para así minimizar el impacto sobre throughput y el ancho de banda disponible para el tráfico de datos. La tabla 6-2, contiene la notación utilizada en este capítulo.

La cantidad total de paquetes por unidad de tiempo o throughput de un nodo en el protocolo TSCH depende fundamentalmente de la duración del timeslot (T_s), debido a que cada nodo de la red solo puede realizar una única acción por timeslot: recibir o transmitir. Por lo tanto, con una duración típica de 10 ms por T_s , un nodo está limitado a realizar un máximo de 100 acciones por segundo ($1/T_s$), esto es 100 recepciones, 100 transmisiones o una combinación de ambas, dependiendo del flujo de paquetes en la red.

Por ello, en la ecuación (5.1) del capítulo anterior, se ha definido la cantidad total de paquetes de control (C_{pk}) en SDN WISE-TSCH en función del número total de nodos en la red, y los periodos de beacon y reporte. El aislamiento del tráfico de control y el de datos se consigue transmitiendo los paquetes de control en los slots compartidos, mientras que los paquetes

de datos se envían por las celdas dedicadas. Para garantizar que el total de paquetes de control se transmiten de forma correcta en la red, la cantidad de slots compartidos debe ser asignada basándose en B_p , R_p y la cantidad de nodos de la red (n), lo que ha dado lugar a la ecuación (5.3) ($Sh_{SF} = \lceil C_{pk} \cdot SF_{sz} \cdot \frac{T_s}{T} \rceil$). De este modo se tienen las celdas suficientes para que todos los paquetes de beacon de los nodos del primer nivel (n_{L1}) lleguen al sink, además, de los paquetes de reporte de toda la red. Aunque la cantidad de recursos asignados son suficientes para transmitir los datos al sink sin que se acumulen paquetes en las colas, existe probabilidad de colisiones debido al funcionamiento de los slots compartidos.

Notación	Definición	Unidades
n	Número de nodos excluyendo el sink	nodos
C_{pk}	Paquetes de control	paquetes/s
Th_D	Throughput de datos	paquetes/s
Sh_{SF}	Timeslots compartidos	timeslots
SF_{sz}	Longitud de Slotframe	timeslots
H_i	Cantidad de saltos entre el nodo i y el sink	saltos
T_s	Duración del timeslot	ms
T	Período de evaluación	s
B_p	Periodo de Beacon	s
R_p	Periodo de Reporte	s
P_f	Frecuencia de paquete	paquetes/s
D_p	Periodo de datos ($1/P_f$)	s
n_{L1}	Número de nodos a un salto del sink	nodos
r	Número de interfaces radio	interfaces
P_s	Tamaño del paquete	Bytes
Ind	Indicadores	Bytes
P_h	Encabezado del paquete	Bytes
NR	Número de repeticiones	path/slotframe

Tabla 6-2: Resumen de las notaciones utilizadas en este capítulo

Basándose en estas ecuaciones, para el caso particular del nodo sink, el flujo en paquetes de control por segundo que debe procesar se obtiene en la ecuación (6-1). El primer término corresponde a los paquetes de beacon que debe recibir, que depende de los nodos en el área de cobertura, en este caso n_{L1} , más el envío de su propio beacon. Además, debe recibir los paquetes de reporte de los n nodos en la red.

$$C_{pk}^s = \frac{n_{L1} + 1}{B_p} + \frac{n}{R_p} \quad (6-1)$$

De aquí que aumentar la cantidad de slots compartidos reduzca directamente la cantidad de celdas disponibles para datos. Por lo tanto, el throughput real disponible para datos en el sink en un segundo se calcula de acuerdo a la ecuación (6-2). La cual equivale a la relación de ocupación del slotframe con slots compartidos: $(SF_{sz} - Sh_{SF})/SF_{sz}$ por el throughput máximo (T/T_s) o lo que es lo mismo, el throughput máximo de paquetes por segundo $(1/T_s)$ menos la cantidad de paquetes de control por segundo obtenidos en la ecuación (6-1).

$$Th_D^s = Th_D(T = 1) = \frac{SF_{sz} - Sh_{SF}}{SF_{sz} \cdot T_s} = \frac{1}{T_s} - C_{pk}^s \quad (6-2)$$

6.1.3. Propuesta de escalabilidad en SD-IWSN

La figura 6-1 muestra un ejemplo de red SD-IWSN, en la que el sink está enlazado con el controlador SDN a través de una red cableada. La red SD-IWSN utiliza 4 de los 16 canales disponibles en TSCH. En cada timeslot se utiliza un canal físico diferente, debido a la rotación de canales que realiza el mecanismo TSCH empleando el parámetro ASN, que se va incrementando en cada timeslot. En este caso el sink únicamente dispone de una interfaz radio independientemente de la cantidad de canales, por lo que únicamente puede realizar una acción (Rx o Tx) en cada timeslot. Por lo tanto, al tener dos hijos, tendrá que usar dos timeslots diferentes para la recepción de cada uno.

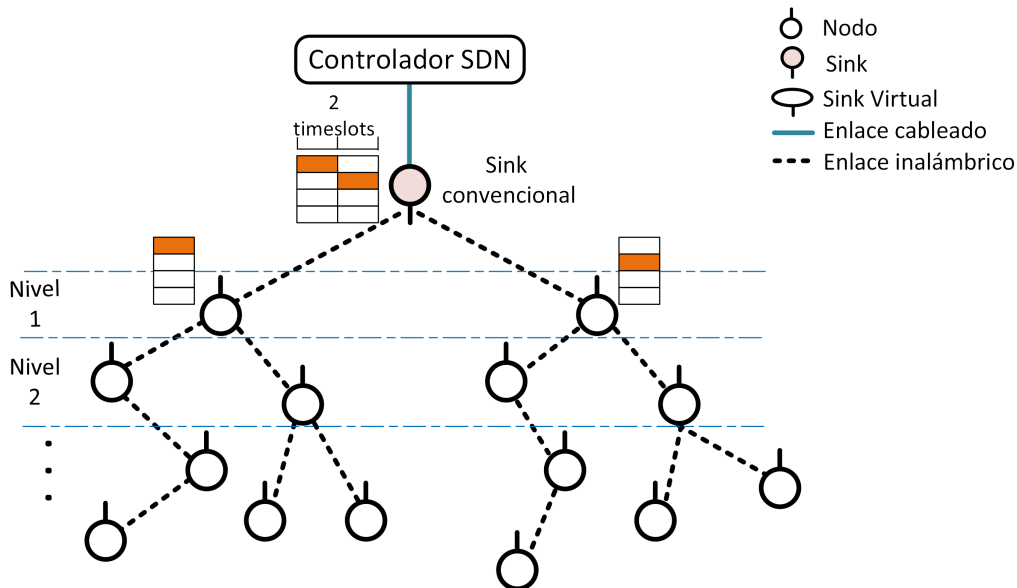


Figura 6-1: SD-IWSN sink convencional con una única interfaz de radio

Al agregar una nueva interfaz radio al sink, como se muestra en la figura 6-2, es posible recibir de forma simultánea en el mismo timeslot utilizando dos canales diferentes. En este caso el sink estará compuesto por dos nodos convencionales que siguen planificaciones diferentes,

pero que el controlador SDN orquesta para que se comporten como un único sink. Para implementar este tipo de nodos, no se utiliza un único dispositivo con capacidad multi-radio, sino una agregación lógica de múltiples dispositivos simples, orquestados de forma centralizada desde el controlador SDN. Esto, reduce de forma significativa la complejidad de implementar sistemas con capacidad multi-radio simultánea, permitiendo su implementación con equipos convencionales.

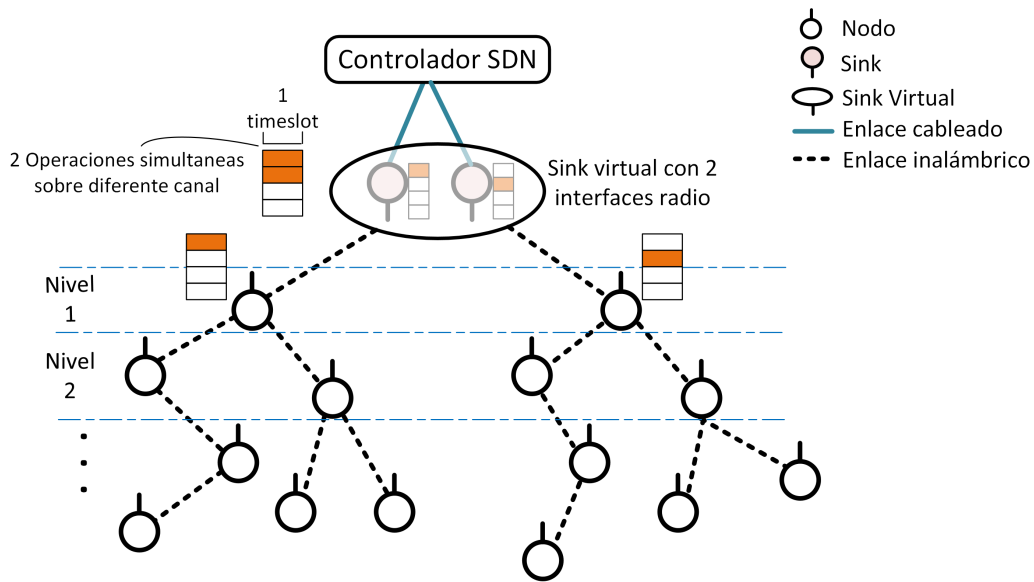


Figura 6-2: SD-IWSN sink virtual con 2 interfaces radio usando 4 canales.

En ejemplo de la figura 6-2, se usan dos interfaces radio para una red de cuatro canales TSCH. Sin embargo, es posible crear sink virtuales que agreguen un nodo para cada uno de los 16 canales disponibles en TSCH, permitiendo utilizar la radio de manera simultánea en todos los canales disponibles.

Este aumento de interfaces radio en el sink tiene un impacto directo sobre la escalabilidad, ya que, el sink convencional está limitado a realizar un máximo de 100 acciones TSCH por segundo ($1/T_s$), limitando la cantidad de nodos que puede tener la red y el throughput disponible para datos. Cuando se agregan r interfaces de radio, el sink puede realizar r acciones en cada timeslot, lo que aumenta directamente la capacidad del sink a r/T_s , reduciendo así el cuello de botella que se produce en el sink. Sin embargo, como los demás nodos de la red no tienen capacidad multi-radio, la saturación se producirá en los nodos a un salto del sink (n_{L1}).

Estos nodos deben recibir el tráfico de datos de los demás nodos de la red, lo cual se representa como: $(n - n_{L1}) \cdot D_p$. Además de recibirlos, deben generar sus propios paquetes y enviar el total al sink, por lo que la cantidad de acciones en transmisión se expresa como: $n \cdot D_p$. Por otro lado, la recepción de los paquetes de datos desde otros nodos también con-

sume recursos TSCH, reduciendo el throughput máximo para transmisión. Para estos nodos convencionales, la cantidad de acciones en recepción y transmisión siempre debe ser menor o igual al throughput disponible para datos (Th_D). Por lo tanto, en la ecuación (6-3), el primer término representa la cantidad de acciones en recepción que deben realizar los nodos en n_{L1} y el segundo término la cantidad de acciones de transmisión. Como para cada acción se requiere un slot, el total de acciones debe ser menor o igual que el throughput de datos disponible entre todos los nodos del primer nivel ($Th_D \cdot n_{L1}$).

$$(n - n_{L1}) \cdot D_p + n \cdot D_p \leq Th_D \cdot n_{L1} \quad (6-3)$$

La cantidad máxima de nodos en la red, puede obtenerse al despejar n en la ecuación (6-3) e igualando al valor máximo del throughput como se muestra en la ecuación (6-4).

$$n \cdot \frac{2 \cdot D_p}{n_{L1}} = Th_D + D_p \quad (6-4)$$

Al reemplazar en esta ecuación los valores correspondientes para el throughput de datos y la cantidad de paquetes de control por segundo en el sink se obtienen las siguientes expresiones:

$$\begin{aligned} n \cdot \frac{2 \cdot D_p}{n_{L1}} &= \frac{1}{T_s} - C_{pk}^s + D_p \\ n \cdot \frac{2 \cdot D_p}{n_{L1}} &= \frac{1}{T_s} - \frac{n_{L1} + 1}{B_p} - \frac{n}{R_p} + D_p \\ n \left(\frac{2 \cdot D_p}{n_{L1}} - \frac{1}{R_p} \right) &= \frac{1}{T_s} - \frac{n_{L1} + 1}{B_p} + D_p \end{aligned}$$

Por lo tanto, despejando n de la ecuación (6-4) la cantidad máxima de nodos que puede tener la red depende de la cantidad de nodos en el primer salto, la frecuencia de generación de paquetes de datos (P_f o $1/D_p$) y el throughput disponible, de acuerdo a la ecuación (6-5).

$$n = \frac{n_{L1} \cdot R_p}{2 \cdot D_p \cdot R_p + n_{L1}} \left(\frac{1}{T_s} - \frac{n_{L1} + 1}{B_p} + D_p \right) \quad (6-5)$$

La ecuación anterior obtiene el número máximo de nodos que permiten utilizar todo el throughput disponible de cada uno de los nodos del primer salto. De acuerdo a este planteamiento, el número óptimo de interfaces radio (r_{opt}) que debe tener el sink virtual depende de la cantidad de datos que envíen el conjunto de nodos de la red SD-IWSN, de acuerdo a la ecuación (6-6).

$$r_{opt} = \frac{n \cdot D_p}{Th_D} \quad (6-6)$$

En la figura 6-3, se muestra r_{opt} en función de la frecuencia de paquetes y el número de nodos de la red. El número de nodos se obtiene a partir de la ecuación (6-5) utilizando $B_p = 3$, $R_p = 6$ y $n_{L1} = 8$. Se observa que el número óptimo de interfaces de radio aumenta a medida que aumenta el throughput de la red, es decir, cuando aumenta la frecuencia de paquetes o la cantidad de nodos en la red. Para cada uno de estos parámetros, el valor máximo de r_{opt} es diferente, ya que depende de la saturación de los nodos en n_{L1} . A partir de este punto, aumentar el número de radios no supone una mejora en la red, ya que el sink consigue un throughput que excede la capacidad de todos los nodos en n_{L1} .

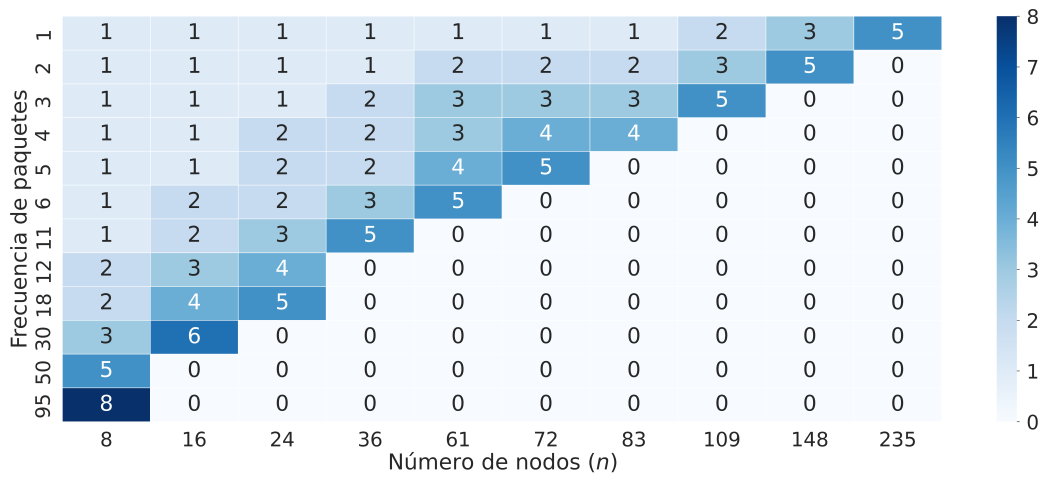


Figura 6-3: Número de interfaces de radio en función del número de nodos y de la frecuencia de paquetes con $n_{L1} = 8$

Este valor máximo de las interfaces de radio varía entre n_{L1} y $n_{L1}/2$. Tiende hacia el mínimo cuando el throughput de la red aumenta debido a un aumento en la cantidad de nodos n , ya que cuantos más nodos haya en la red, más acciones de recepción deben realizar los nodos de primer nivel. Esto reduce la cantidad de transmisiones al sink virtual. Por otro lado, cuando el throughput de la red aumenta debido a la frecuencia de paquetes, los nodos en n_{L1} se utilizan menos en acciones de recepción, y es posible realizar un número más significativo de transmisiones, por lo que se requiere un mayor número de interfaces de radio.

La figura 6-3 también permite observar las limitaciones de las redes SD-IWSN que utilizan una sola interfaz radio. Se observa como pueden tener hasta un máximo de 83 nodos generando un paquete de datos por segundo, mientras que al utilizar un sink virtual con la misma frecuencia de generación de paquetes, se pueden tener redes de hasta 235 nodos con 5 interfaces radio. El límite de 235 nodos, se debe al agotamiento del throughput de los nodos a un salto del sink, por eso en este caso, aumentar la cantidad de interfaces radio en el sink virtual no genera un aumento en el throughput. Sin embargo, si se mantiene la cantidad de nodos es posible aumentar la frecuencia de generación de paquetes de datos en un factor igual

a la cantidad de interfaces radio. Por ejemplo, para 83 nodos con una única interfaz radio solo es posible enviar un paquete por segundo en cada nodo, al aumentar a 4 las interfaces radio, es posible aumentar la frecuencia de generación de paquetes a 4 paquetes/s.

En la figura 6-4 se ha aumentado la cantidad de nodos en el primer nivel a 24. Esto permite aumentar la cantidad total de nodos hasta 371, al enviar un paquete de datos por segundo, sería necesario 13 interfaces radio en el sink virtual. En este caso, cuando se utilizan la cantidad mínima de nodos, no es posible conseguir que el número de radios sea igual a los nodos de primer nivel ($r = n_{L1}$), porque la cantidad disponible de canales físicos en el estándar IEEE 802.15.4e es de 16. En estas figuras también se observa como aumentar la cantidad de radios no incrementa de forma proporcional la cantidad de nodos, esto se debe a que al aumentar los nodos aumenta de forma más significativa el tráfico de control y la ocupación del slotframe. Además, la cantidad máxima de radios a utilizar depende completamente de la topología de red, específicamente de los nodos que estén a un salto del sink.

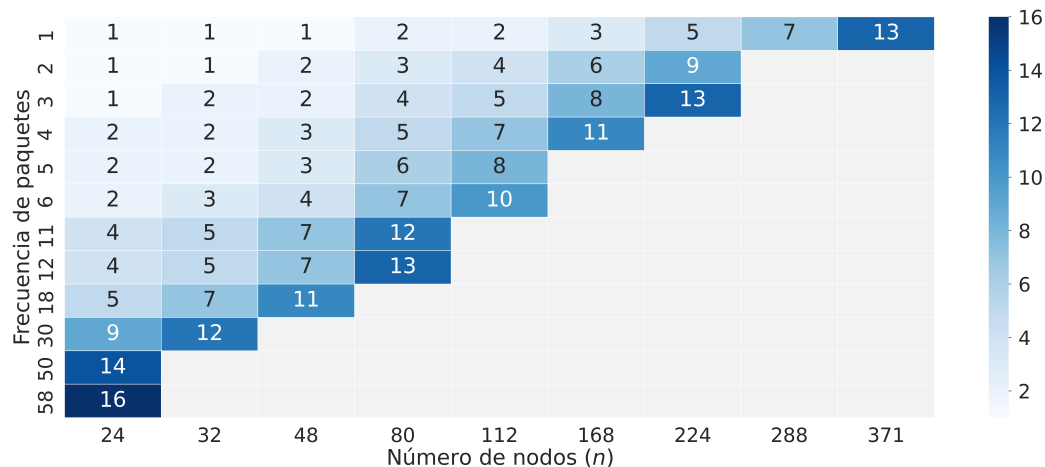


Figura 6-4: Número de interfaces de radio en función del número de nodos y de la frecuencia de paquetes con $n_{L1} = 24$

Por esto, es más eficiente utilizar los canales físicos restantes en otras SD-IWSN que operen de manera concurrente. Siguiendo el ejemplo de la figura 6-2, que muestra una SD-IWSN con 2 interfaces radio y 4 canales, quedarían 12 canales libres del espectro de frecuencias, lo que permitiría crear otras tres SD-IWSN, tal y como se muestra en la figura 6-5. En este otro caso se consiguen desplegar hasta 4 SD-IWSN, integradas por medio de la red cableada y orquestadas por el mismo controlador SDN. Como cada una de las SD-IWSN está utilizando un patrón de salto en frecuencia sobre canales diferentes, no se produce ninguna interferencia entre ellas. Esto permite la coexistencia de estas redes en el mismo entorno industrial, aumentando la densidad de nodos sin incrementar el tráfico de control generado por el controlador SDN.

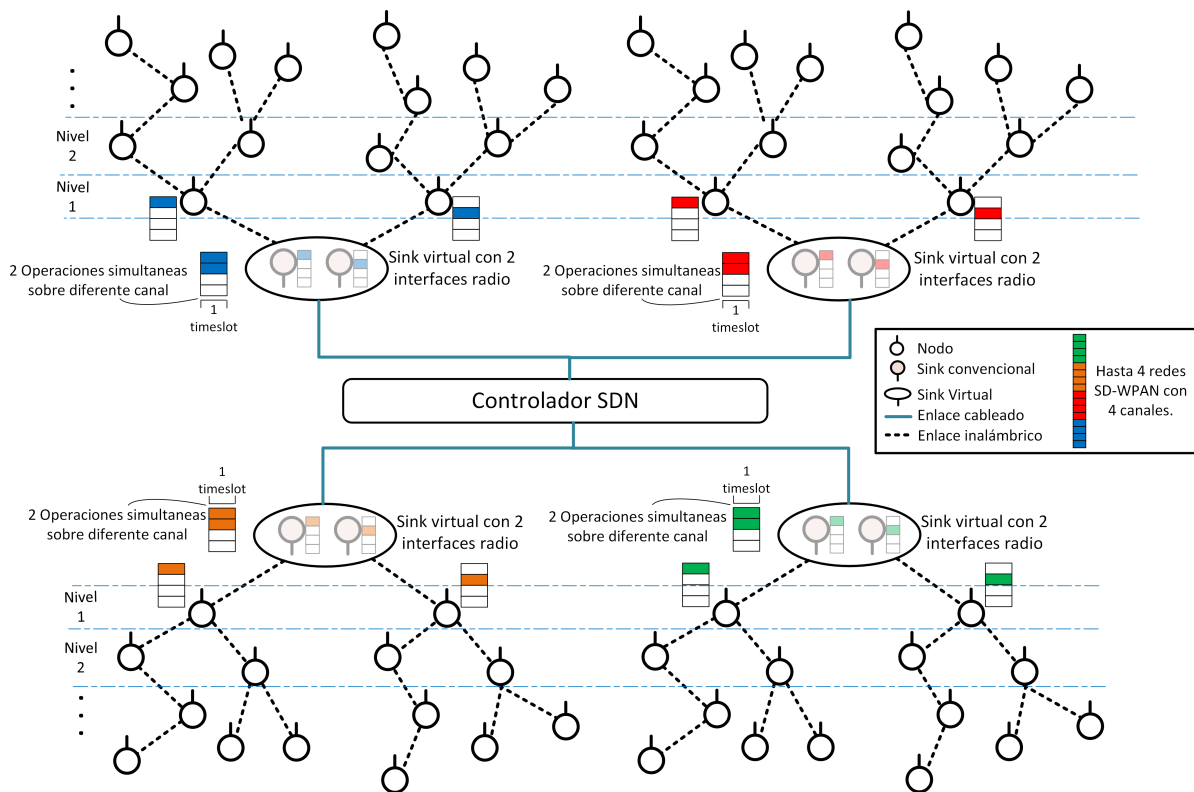


Figura 6-5: Múltiples SD-IWSN separadas por frecuencia y orquestadas por el mismo controlador SDN

6.1.4. Topología

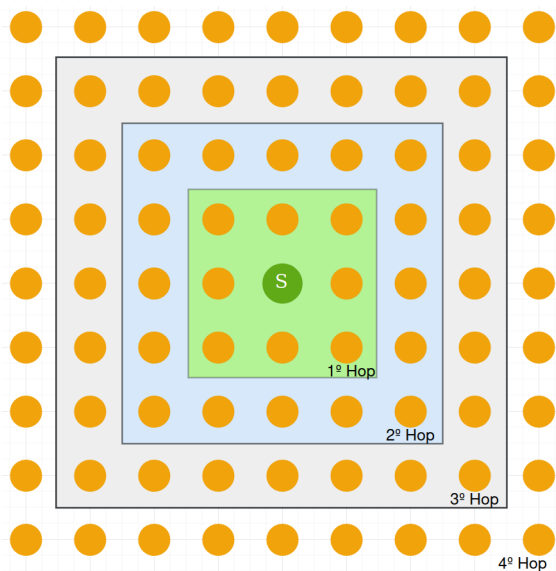


Figura 6-6: Topología de malla cuadrada con sink en el centro

Para generalizar los resultados obtenidos se utiliza una topología de malla cuadrada como en la figura 6-6 de tamaño $m \cdot m$. Los nodos están separados una distancia de 16 metros, y la red aumenta de tamaño de forma regular con m impar, para mantener el sink en el centro de la red.

La cantidad de nodos en cada salto para la topología anterior se muestra en la figura 6-7 donde se tiene el crecimiento de una red de malla cuadrada conformada por 1000 nodos con dos áreas de cobertura que definen los nodos en el primer nivel. Para n_{L1} se tendría un máximo de 16 saltos para una red de 1000 nodos, mientras que una red de 100 nodos no supera los 5 saltos. Aumentando la cobertura de los nodos o disminuyendo la distancia entre ellos, se aumentan los nodos del primer nivel, reduciendo la distancia al sink de todos los nodos. Para una red de 1000 nodos con $n_{L1} = 24$ los nodos más alejados están a 7 saltos del sink.

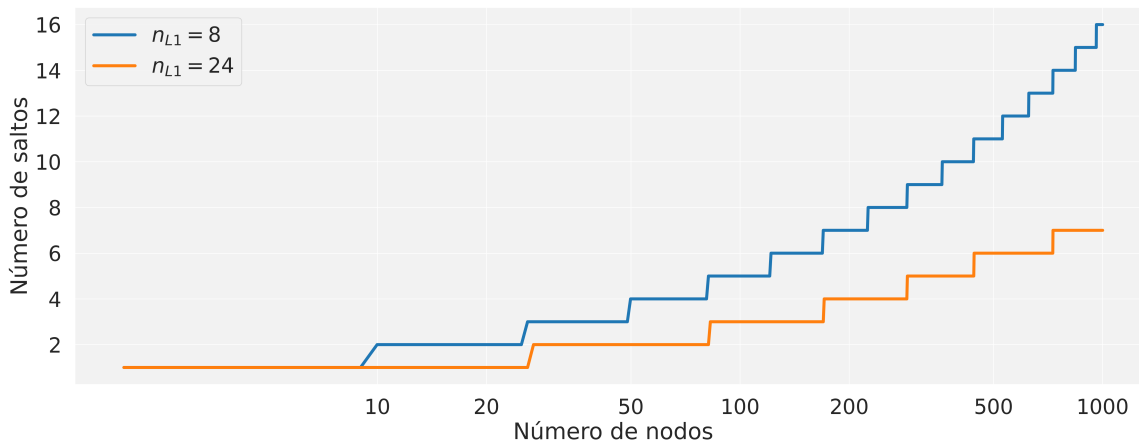


Figura 6-7: Aumento de los saltos al sink en topología de malla cuadrada en función del número de nodos

6.1.5. Configuración de rutas

En una red SDN, el controlador debe configurar las acciones en cada uno de los nodos, por lo tanto, los paquetes de configuración también deben poder ser escalables y generar el menor aumento posible del tráfico de control. Por esto, para enviar la configuración de rutas y la asignación de celdas, el controlador SDN usa el paquete OpenPathTSCH, que contiene las reglas para un flujo específico, la ruta y la configuración *hop-by-hop* de las celdas asignadas para la transmisión TSCH. Para evitar el envío masivo de esta información desde el controlador, este tipo de paquete tiene un mecanismo de auto-propagación, donde los nodos receptores procesan y reenvían el paquete al siguiente salto de la ruta, reduciendo la cantidad de paquetes que se deben enviar desde el controlador SDN. Sin embargo, el tamaño de este paquete está limitado a 116 Bytes, que son asignados de acuerdo a la ecuación (6-7)

obtenida en el capítulo 3. La cabecera o *packet header* (P_h) es común en todos los paquetes SDN-WISE y tiene una longitud de 10 Bytes, en este se especifican los parámetros necesarios para el procesamiento y reenvío de los paquetes. El campo *indicator* (Ind) es específico para el paquete OpenPathTSCH, y contiene los valores de NR , SF_{sz} y el número de nodos en la ruta. Estos mensajes son utilizados para configurar las celdas TSCH correspondientes en cada nodo.

$$P_s = P_h + 5 \cdot NR + Ind + 2 \cdot n + 2 \cdot NR \cdot (n - 1) \tag{6-7}$$

El tamaño de este paquete limita la cantidad de saltos y NR que puede tener un flujo. En la figura 6-8 se muestra una línea discontinua, que representa la capacidad de este tipo de paquete de configuración (116 Bytes). En escenarios con una alta cantidad de nodos, usando múltiples envíos de un flujo dentro del slotframe (NR), el valor máximo en saltos que puede configurar este paquete es de 24 (curva azul), donde se realiza una única transmisión del flujo por cada slotframe ($NR = 1$). Sin embargo, como se muestra en la figura 6-7, esta cantidad de saltos no se consigue en topologías de malla cuadrada, donde con 1000 nodos, la distancia máxima al sink es de 16 saltos. Si, por el contrario, los flujos requieren una mayor frecuencia de envío de datos, es necesario aumentar la cantidad de repeticiones. En este caso es posible configurar hasta 10 transmisiones por slotframe para un flujo con una distancia de 4 saltos al sink en un único paquete OpenPathTSCH.

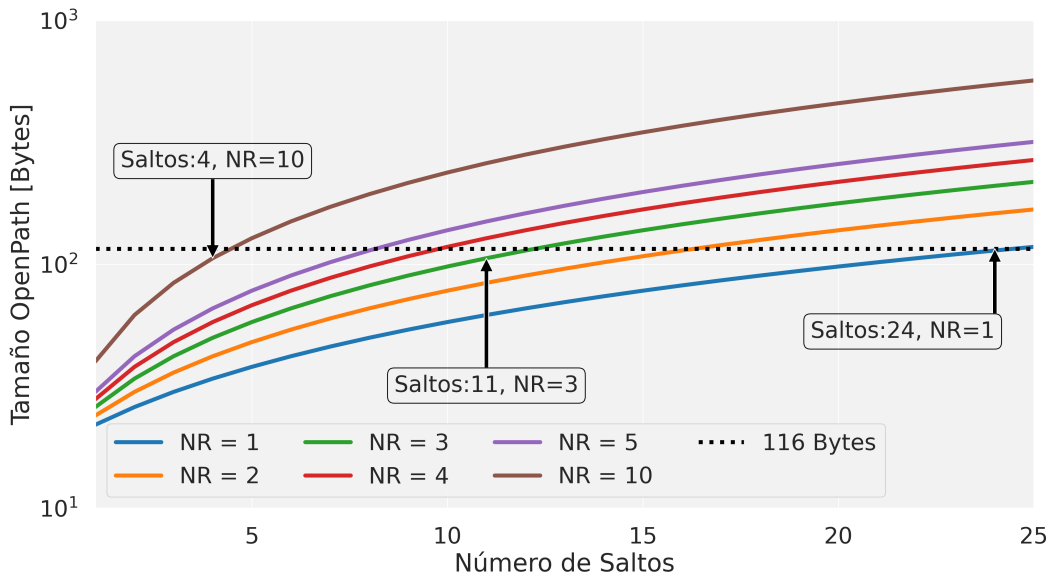


Figura 6-8: Tamaño del OpenPathTSCH en función del número de nodos y repeticiones dentro del Slotframe

6.2. Evaluación del rendimiento

Esta sección muestra los resultados usando la topología de malla cuadrada con sink en el centro, mostrado en la figura 6-6. Se han realizado diferentes tipos de simulaciones utilizando la herramienta Cooja, desplegando diferentes cantidades de nodos. El objetivo de estas pruebas consiste en analizar el rendimiento y las limitaciones de escalabilidad de un sink convencional en comparación con el sink virtual propuesto en esta tesis, utilizando diferentes configuraciones con 4, 8 y 16 interfaces radio. Las simulaciones se han llevado a cabo utilizando como base la pila de comunicaciones propuesta en Contiki-NG, en la que se han utilizado los nodos con la plataforma *sky* para emular el comportamiento real de los equipos. Un resumen de los parámetros generales utilizados en las simulaciones se muestra en la tabla 6-3.

Parámetro	Valor
Tiempo de simulación	180 s
Número de ejecuciones	10
Periodo de datos (D_p)	500 ms
Frecuencia de paquetes (P_f)	2 p/s
n (Topología)	8(3x3), 24(5x5), 48(7x7), 80(9x9)
Número de canales	4
Timeslot (T_s)	10 ms
Tamaño de Slotframe (SF_{sz})	101 timeslots
Timeslots compartidos (Sh_{SF})	6, 8, 12, 18 timeslots
Periodo de Beacon (B_p)	3 s
Periodo de Reporte (R_p)	6 s

Tabla 6-3: Parámetros utilizados en la simulación

El controlador SDN utilizado es el SDN WISE-TSCH el cual incorpora un servicio TCP para conectarse con el sink a través de la red cableada. Este controlador incluye aplicaciones para la administración de los flujos, el proceso de enrutamiento y el planificador TSCH. Además, para probar el enfoque multisink se utilizó ONOS, un controlador de carácter industrial, que tiene una mayor robustez y un diseño totalmente modular. Esto permite cargar protocolos de *southbound* no estándar, como el que utiliza SDN WISE.

6.2.1. Sink convencional

Partiendo de la red de malla cuadrada y de la ecuación (5.1) para el tráfico de control total, se evalúa la cantidad media de acciones (Rx o Tx) que debe realizar cada nodo para recibir y retransmitir el tráfico de control, dependiendo de su distancia al sink y la cantidad de nodos en la red. Como se muestra en la figura 6-9, los nodos a 2 saltos del sink tienen poca

saturación, solo alcanzan el nivel límite de TSCH cuando es una red de más de 700 nodos, debido a la agregación del tráfico de los nodos superiores. Para un solo salto este límite se alcanza a los 370 nodos y para el sink a los 100, por lo tanto, es el sink el que limita la cantidad de nodos de la red. Sin embargo, estos valores máximos se obtienen en un entorno ideal, donde no hay colisiones en los slots compartidos y el tráfico es balanceado por igual entre todos los nodos de la red. Por lo tanto, el throughput disponible para datos se verá reducido por los slots compartidos asignados al tráfico de control.

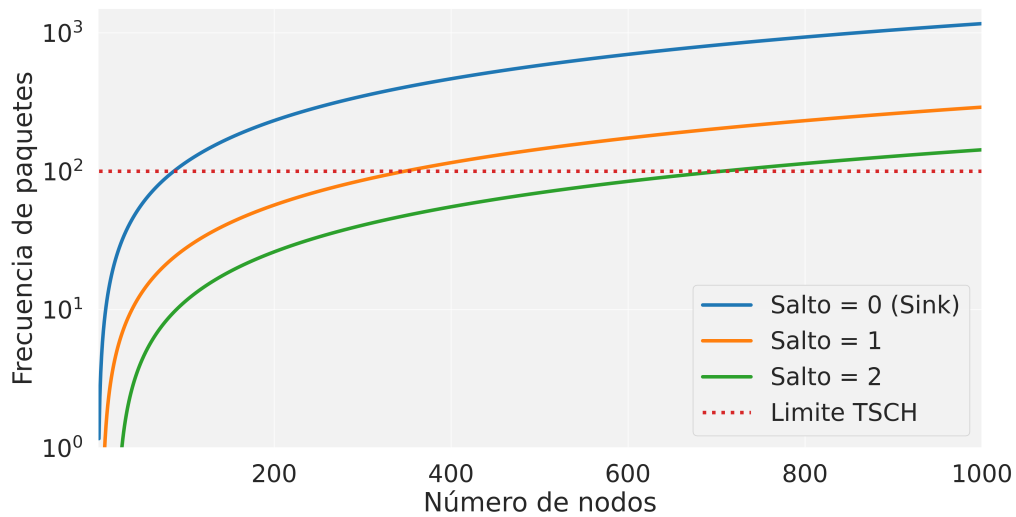


Figura 6-9: Promedio de paquetes recibidos/transmitidos por cada nodo en función de la distancia al sink

Para tener un mayor throughput disponible para datos se ha utilizado $B_p = 3s$ y $R_p = 6s$, donde para 81 nodos el tráfico de control requiere un 20% del ancho de banda total. Se puede observar como en la figura **6-10(a)** la cantidad total de paquetes datos aumenta progresivamente con el número de nodos. Sin embargo, con 49 y 81 nodos la cantidad total de paquetes de datos no varía de forma significativa. Esto se debe a que se ha alcanzado el throughput máximo de la red, y a partir de ese instante los paquetes empiezan a almacenarse en las colas hasta que desbordan y son descartados por el nodo. Esta pérdida de paquetes se observa en la figura **6-10(b)**, donde para las primeras dos redes el PDR es del 100%, para la red de 49 nodos ya desciende hasta el 80% y con los 81 nodos desciende hasta el 45%.

En la figura **6-11**, se observa la proporción de timeslots que están siendo utilizados en transmisión o recepción por el sink en función de la cantidad de nodos. El primer valor muestra los resultados para la topología de 3x3, en la que el sink se ubica en el centro y los 8 nodos a un salto de este. La ocupación del slotframe no es significativa, y la mayor parte de esta ocupación se debe a la cantidad de shared slots usados para enviar el tráfico de control. Cuando se aumenta la cantidad de nodos hasta 24 (5x5), la ocupación tiene un

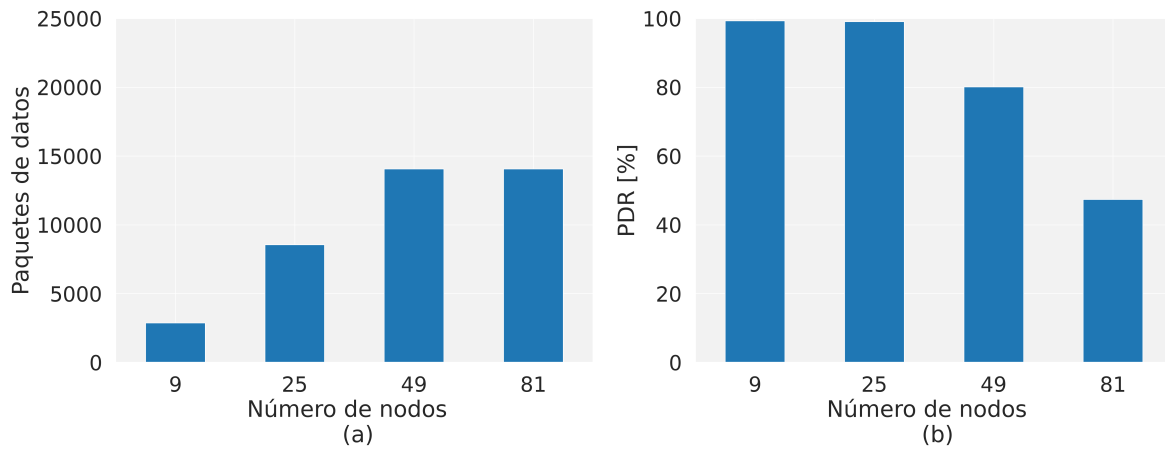


Figura 6-10: Total paquete de datos en simulación con sink convencional

impacto significativo en el sink en comparación al resultado anterior, ya que la ocupación está sobre el 50 %, mientras que la de los nodos en el primer salto (n_{L1}) es inferior al 20 %. En las topologías siguientes se observa como el sink alcanza la ocupación máxima del slotframe, mientras que la ocupación de los demás nodos está por debajo del 40 % y 60 %, para las topologías de 7x7 y 9x9 respectivamente. Por lo tanto, la red genera más paquetes de los que puede recibir el nodo sink, donde se genera un cuello de botella que termina saturando la cola de los nodos cuando el throughput de la red supera los 80 paquetes por segundo, limitando igualmente las capacidades de escalabilidad de la red SD-IWSN.

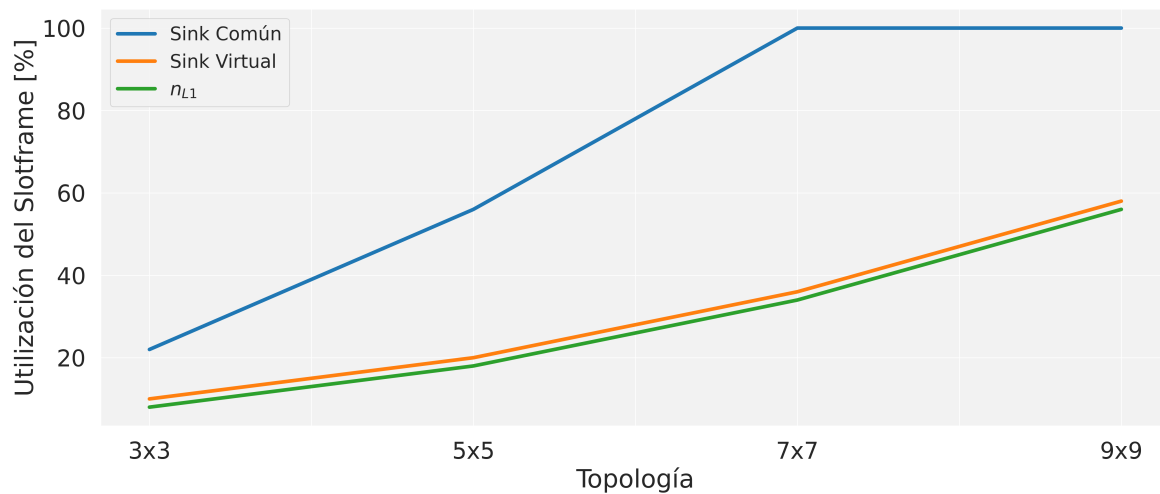


Figura 6-11: Ocupación del slotframe en función de la cantidad de nodos

6.2.2. Sink virtual

De acuerdo al apartado anterior, el uso de un único sink limita las capacidades de escalabilidad de las SD-IWSN, debido a que es el único punto de entrada de los n nodos de la red. El nivel de saturación de la red, depende principalmente de dos factores: la cantidad de nodos y la frecuencia de envío de paquetes (datos y control) desde cada nodo, ambos relacionados con la ocupación del slotframe TSCH y el throughput total de la red SD-IWSN.

Aumentar cualquiera de estas variables tiene un impacto directo en la cantidad de recursos TSCH que deben asignarse, especialmente en el sink. Debido a que estos recursos son limitados, el sink pierde paquetes cuando el producto $n \cdot D_p$ supera el throughput impuesto por el protocolo TSCH. Para reducir esta limitación, se han utilizado las implementaciones de sink virtuales propuestas, en las que el uso de múltiples interfaces radio (r) permite que cada uno de estos dispositivos pueda realizar múltiples acciones en canales diferentes y en el mismo instante de tiempo. Esto permite cambiar la limitación en términos de escalabilidad del mecanismo TSCH, incrementándolo en un factor que dependerá de la cantidad de interfaces radio utilizadas por el sink virtual, pasando de $1/T_s$ a r/T_s .

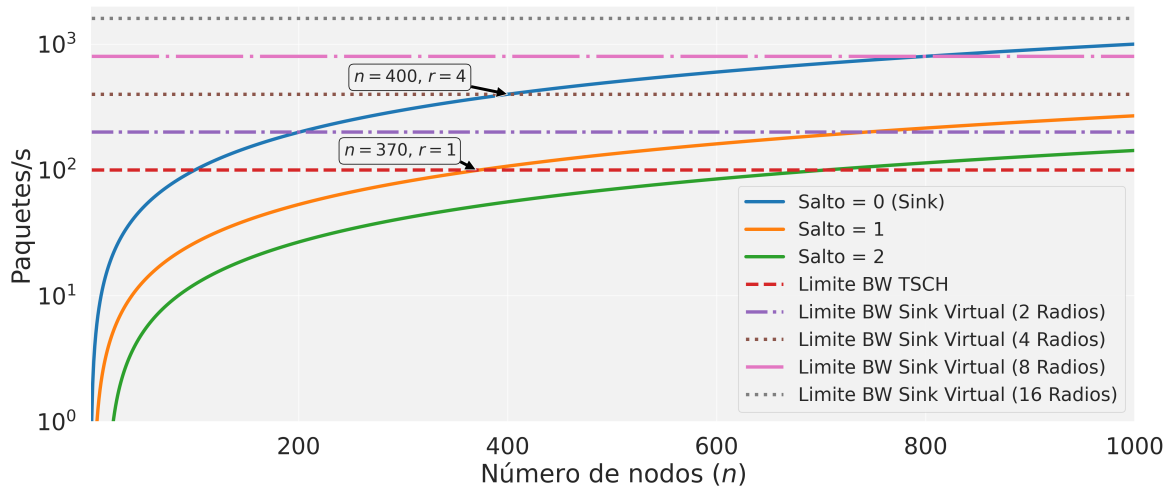


Figura 6-12: Promedio de paquetes recibidos/transmitidos por los nodos en función de la distancia al sink

La figura 6-12 muestra la distribución de la carga de control en cada nodo, en función de la distancia al sink virtual, y el número de nodos de la red. Además, en este caso se han agregado sinks virtuales con 2, 4, 8 y 16 interfaces radio. Para el caso del sink virtual con 16 interfaces radio, el máximo número de canales en 2.4 GHz definidos en el estándar IEEE 802.15.4e, se puede soportar el tráfico de control de más de 1000 nodos, ya que para este valor no alcanza el límite del ancho de banda. Sin embargo, en una red de 1000 nodos, los nodos en el primer nivel estarían saturados, impidiendo que el tráfico llegue hasta sink virtual. Este

comportamiento es similar para el sink virtual con 4 y 8 interfaces radio, donde el cuello de botella son los nodos del primer salto. Tener un alto número de interfaces radio incrementa de forma significativa el throughput en el sink virtual, lo que tiene una implicación directa sobre las capacidades de escalabilidad de la red SD-IWSN, aunque estará limitado por el throughput máximo que pueden generar los nodos del primer salto (n_{L1}).

Los resultados de agregación de interfaces radio se pueden observar en la figura **6-13(a)** donde, bajo las mismas condiciones que el sink convencional, tiene un crecimiento constante en la cantidad total de paquetes. En este caso se observa claramente la diferencia entre la red de 49 y 81 nodos, donde alcanza a recibir 27.000 paquetes de datos, duplicando el throughput anterior. El PDR mostrado en la figura **6-13(b)** se mantiene en todos los casos sobre el 99 %, por lo que no existe ningún tipo de saturación.

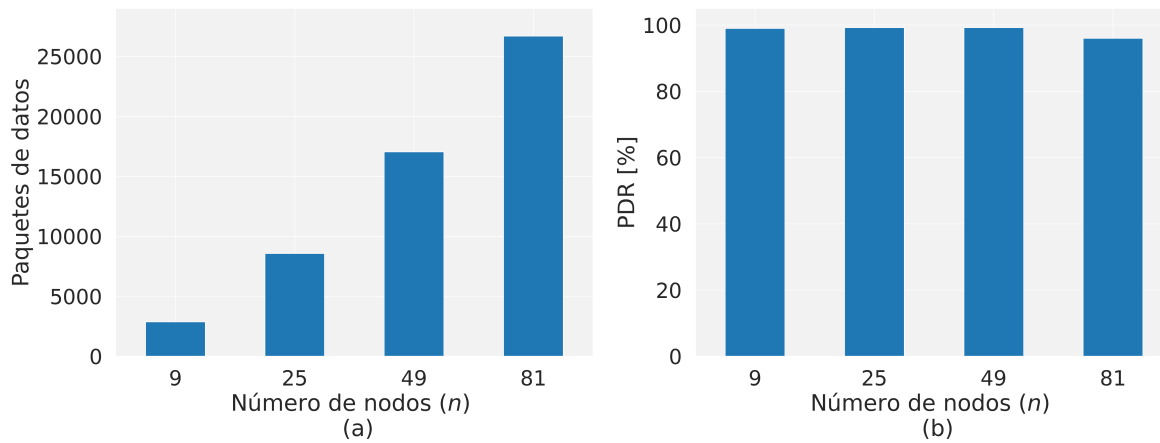


Figura 6-13: Paquetes de datos totales en simulaciones con sink virtual compuesto de 2 interfaces radio

Debido a que ahora se tienen 4 interfaces radio con planificaciones independientes, la ocupación del slotframe en el sink se reduce de forma drástica de acuerdo a la figura **6-11**, donde para el sink virtual se mantiene por debajo del 60 % en todos los casos. Además, se observa como la curva del sink virtual se ajusta completamente con la utilización del slotframe en los nodos del primer nivel. Tal y como demuestran estos resultados, la agregación de interfaces radio en sink virtuales aumenta de forma considerable el ancho de banda, evitando las congestiones y permitiendo utilizar todo el throughput disponible de los nodos del primer salto. Estas mejoras de rendimiento no solo permiten optimizar estas SD-IWSN aprovechando mejor los recursos radio, sino que también ayuda a aumentar la cantidad de nodos desplegados y el periodo de envío de datos.

Finalmente, si analizamos las consideraciones energéticas de utilizar un sink virtual con múltiples interfaces radio, el impacto que supone para el resto de los nodos de la red es transparente, ya que para estos dispositivos únicamente existe un sink con el que establecer

su comunicación. En cuanto al sink virtual que agrupa múltiples interfaces radio, el consumo de energía será mayor, ya que debe contar con varios de estos dispositivos. Sin embargo, dado que debe permanecer conectado al controlador cuenta con una fuente de alimentación permanente. Por lo tanto, este mayor consumo no se traducirá en ningún efecto negativo sobre el tiempo de vida útil de la red, al igual que ocurre con el consumo de la red con un sink convencional.

6.2.3. Múltiples SD-IWSN con sink virtual

Empleando las soluciones basadas en sink virtuales, se ha demostrado como es posible mejorar las capacidades de ancho de banda total de una SD-IWSN, cuyo impacto se traduce en una mejora de la escalabilidad de la red, así como de la frecuencia de transmisión de los diferentes nodos de la red SD-IWSN. Sin embargo, en algunos casos se emplean únicamente 4 de los 16 canales disponibles en la banda de 2.4 GHz. Para mejorar aún más la escalabilidad de la red SD-IWSN y aprovechar completamente el espectro radioeléctrico de TSCH se pueden replicar estas SD-IWSN que utilizan un sink virtual, pero empleando una tupla de 4 canales diferentes, que permanecerán aisladas completamente en dominios de interferencia diferentes.

El controlador SDN está diseñado para agregar múltiples redes, incluso de diferentes tipos, así que la única limitación que tiene el plano de control será el uso en CPU y memoria que tenga el controlador SDN por cada nodo administrado. Por lo tanto, no hay diferencia a nivel de controlador SDN entre aumentar el número de dispositivos en una red, o a través de múltiples redes. Si este crecimiento de la red se llevase a cabo teniendo en cuenta la opción de una sola red, tendría un impacto negativo considerable, ya que el tráfico de control se reenvía a través de slots compartidos. Los slots compartidos son comunes para todos los nodos de la red, por lo tanto, cuando un nodo debe enviar información de control puede usar cualquiera de los slots compartidos configurados para hacer la transmisión. Este comportamiento aumenta la probabilidad de que se produzcan colisiones en el tráfico de control, ya que aumenta la cantidad de nodos en la misma red. Sin embargo, utilizando múltiples SD-IWSN no solo se mitiga este problema, sino que también se aprovecha completamente el espectro disponible. En este sentido, la cantidad de nodos de cada SD-IWSN se puede aumentar siempre que los requisitos de ancho de banda control y datos no superen el límite de TSCH, y a partir de aquí debe aumentarse a través de otras SD-IWSN que coexistan. SDN permite administrar las diferentes SD-IWSN desde el mismo controlador, eliminando las dificultades de administración y configuración que tiene implementar múltiples SD-IWSN.

Para comprobar, la coexistencia de múltiples SD-IWSN se usa el controlador ONOS, desplegando 4 SD-IWSN cada una con 4 canales diferentes, en las que la topología de cada una es una malla cuadrada de 80 nodos. Los resultados obtenidos en cuanto a paquetes totales, PDR y ocupación del slotframe tienen el mismo comportamiento que el obtenido en la fi-

SD-IWSN	0	1	2	3	4
CPU	20 %	20.7 %	21.5 %	22.1 %	23 %
RAM	4 Gb	6.67 Gb	9.21 Gb	11.92 Gb	14.55 Gb

Tabla 6-4: Uso de CPU y RAM en el controlador SDN ONOS con múltiples IWSN

gura 6-13 y figura 6-11 para cada una de las redes, ya que se han utilizado grupos de 4 canales diferentes, aprovechando así todo el espectro del estándar IEEE 802.15.4e. El consumo computacional en términos de CPU y RAM puede observarse en la tabla 6-4, donde se tiene el consumo del controlador sin dispositivos conectados, y con hasta 4 SD-IWSN de 80 nodos cada una, donde los sink virtuales están formados por 4 interfaces radio con canales diferentes, lo que en total permite dar servicio a una red de 320 nodos.

6.2.4. Tesbed

Para comprobar el funcionamiento en hardware real, se ha realizado un testbed en el que se han desplegado dos SD-IWSN aisladas en frecuencia, utilizando grupos de 4 canales diferentes. Las redes tienen una topología de malla cuadrada de 3x3 y la potencia de los nodos se ha limitado a -15 dB para realizar las pruebas en un entorno de laboratorio. En este caso, cada red estará compuesta de un nodo sink y 8 nodos que realizan un envío de datos periódico. El sink está conectado a una Raspberry Pi a través de USB, para darle acceso a la red SDN cableada donde se encuentra el controlador SDN, tal y como se muestra en la figura 6-14. Las pruebas se realizan utilizando los dos tipos de sink, primero las dos SD-IWSN con un sink convencional y finalmente ambas SD-IWSN con un sink virtual con 2 interfaces radio, formados a partir de dos OpenMote-B gestionados por el controlador para que los demás nodos de la red lo perciban como un único dispositivo.

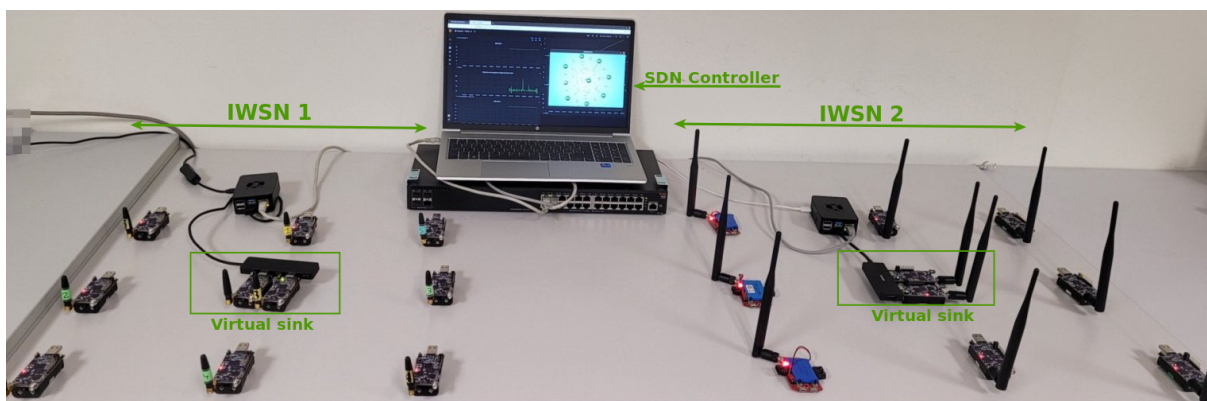


Figura 6-14: Testbed: Dos SD-IWSN con topología de malla cuadrada, sinks virtuales con dos interfaces radio

Con el fin observar el efecto de la saturación con una cantidad limitada de nodos, se utilizan

diferentes frecuencias de generación de paquetes, que van desde los dos paquetes por segundo (500 ms) como en las simulaciones, hasta 20 paquetes por segundo (50 ms). El sistema Operativo de los nodos es el Contiki-NG, sobre el cual se ha desarrollado el firmware para el hardware OpenMote-B y REmote. Los parámetros de configuración utilizados se encuentran en la tabla 6-5.

Parámetro	Valor
Tiempo de simulación	180 s
Número de ejecuciones	5
Período de datos (D_p)	500 ms, 250 ms, 100 ms, 50 ms
Frecuencia de paquetes (P_f)	2 p/s, 4 p/s, 10 p/s, 20 p/s
n (Topología)	8 (3x3)
Número de canales	4
Potencia de transmisión	-15 dB
Timeslot (T_s)	10 ms
Tamaño de Slotframe (SF_{sz})	101 timeslot
Timeslot compartidos (Sh_{SF})	6 timeslot
Período de Beacon (B_p)	3 s
Período de Reporte (R_p)	6 s

Tabla 6-5: Parámetros utilizados en el testbed

Los resultados obtenidos tras inyectar tráfico de datos desde todos los nodos se observan en la figura 6-15, donde se muestra el total de los paquetes recibidos por SD-IWSN para diferentes periodos de envío de datos. Utilizando los periodos de 500 ms y 250 ms, el comportamiento obtenido es similar al de las simulaciones, donde para un número reducido de nodos no hay diferencias entre utilizar un hardware con una interfaz radio o dos. Para los casos con 100 ms y 50 ms se puede observar como la red con el sink convencional llega a su máximo ancho de banda en los 100 ms, ya que las columnas de 50 ms y 100 ms no muestran una diferencia significativa.

Por el contrario, en el sink virtual, donde sus dos interfaces radio permiten tener un mayor flujo de paquetes, se observa una cantidad total de paquetes levemente superior a la del sink convencional en el periodo de 100 ms. Al disminuir el periodo a 50 ms, se observa como el sink virtual duplica la cantidad total de paquetes, mientras el convencional recibe una cantidad de paquetes igual que con el periodo de 100 ms.

Los resultados demuestran que, gracias a la mayor capacidad de ancho de banda en el sink virtual, es posible garantizar un mayor PDR en condiciones con un alto flujo de tráfico, lo que tiene implicaciones directas sobre la escalabilidad de la red y la frecuencia máxima de transferencia de datos. Como se observa en las columnas verde y roja de la figura 6-15, se obtiene un PDR cercano al 100% para los dos primeros casos, tanto con el sink convencional

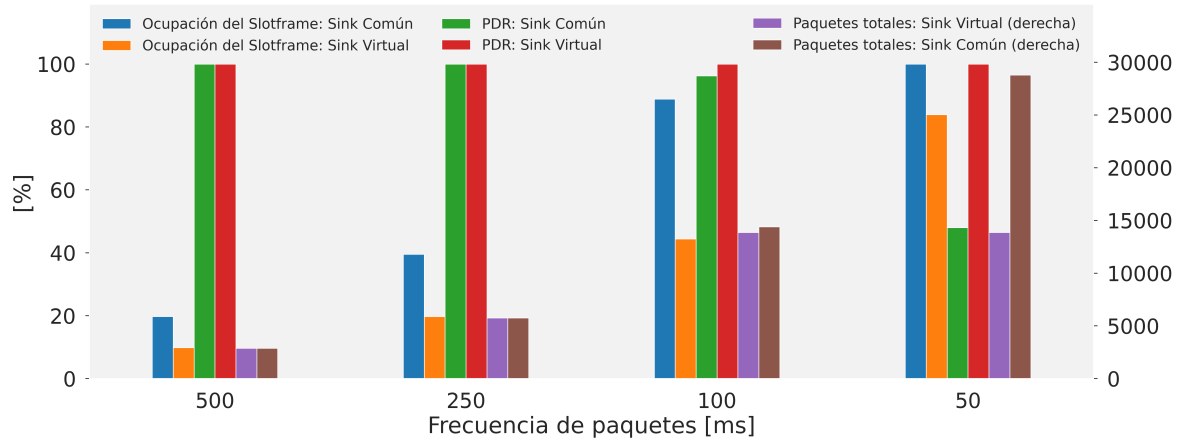


Figura 6-15: PDR y Utilización del Slotframe en el testbed con diferentes frecuencias de paquetes

como con el sink virtual. Sin embargo, a partir de aquí, el sink convencional empieza a tener pérdida de paquetes, hasta perder más del 50 % para las pruebas con un periodo de 50 ms, donde el sink virtual mantiene el PDR sobre el 99 %. Esta pérdida de paquetes, se produce por la falta de recursos TSCH, tal y como se muestra en las columnas azul y naranja de la figura 6-15, que se incrementan de forma progresiva cuando se reduce el periodo, al igual que en la simulación en la que se aumentaba el número de nodos. El sink convencional tiene asignados el 100 % de los timeslots, donde cada uno de los nodos transmisores tiene 30 slots asignados, cerca de un 30 % del slotframe. En el sink virtual no se observa saturación en la utilización del slotframe, ya que, aunque tiene la misma cantidad de timeslots, puede realizar dos recepciones en cada uno, por esto se observa cómo no pierde paquetes ni el uso de slotframe llega al 100 %. Estos resultados pueden extrapolarse a redes con una mayor cantidad de nodos, ya que se han usado frecuencias de generación de paquetes elevadas, para observar el comportamiento de la red con una cantidad alta de flujo de datos. Por ejemplo, con una frecuencia de generación de paquetes de 20 paquetes por segundo ($D_p = 50$ ms) en una red de 9 nodos, genera la misma cantidad de paquetes de datos que una red de 180 nodos con una frecuencia de generación de 1 paquete por segundo.

6.3. Conclusiones

En este capítulo se han abordado los aspectos relevantes de escalabilidad en las IWSN y cómo se han tratado de manera tradicional a través de enfoques de enrutamiento y *clustering*. Sin embargo, estos enfoques no son completamente adecuados para garantizar la calidad del servicio necesaria en los sectores industriales.

Por esta razón, se ha realizado un análisis detallado de las limitaciones que presenta el

protocolo de acceso al medio TSCH en este tipo de redes, especialmente en lo que se refiere al tráfico de control y la distribución del tráfico en los nodos centrales. Para superar estas limitaciones, se propone el uso de sinks virtuales y la optimización del espectro.

La implementación de sink virtuales en las redes SD-IWSN se ha demostrado como una solución efectiva para mejorar la escalabilidad y aumentar la capacidad de nodos en la red. Además, la optimización de la distribución de la red desde un punto centralizado y la combinación de múltiples redes a través del controlador SDN permiten desplegar múltiples redes SD-IWSN que aumentan significativamente la capacidad de nodos en la red y garantizan una alta tasa de entrega de paquetes incluso bajo cargas de tráfico elevadas.

Capítulo 7

Conclusiones y trabajos futuros

En esta tesis se ha realizado un estudio sobre las diferentes arquitecturas que buscaban integrar las redes de sensores inalámbricos y el concepto de las redes definidas por software. En este estudio se ha encontrado que existen grandes dificultades para adaptar este paradigma debido a las limitaciones que tienen este tipo de redes, especialmente el ancho de banda. Estas limitaciones son más notorias cuando se utiliza una arquitectura SDN, ya que esta requiere un intercambio constante de información entre todos los nodos de la IWSN y el controlador SDN.

Sin embargo, en la actualidad las redes IWSN presentan desafíos que son difíciles de abordar mediante enfoques tradicionales, en los que se necesitan protocolos altamente complejos o en los que sus inherentes limitaciones incrementan significativamente la complejidad de las soluciones propuestas. Por ello, la integración con un paradigma más disruptivo como SDN no es solo un requisito, sino también una necesidad que les permitirá seguir siendo competitivas en entornos productivos. Además, reforzará su flexibilidad al poder adaptarse de forma ágil, sencilla y optimizada a los diferentes retos o cambios constantes que requieren.

Por ello, las aportaciones de esta tesis han permitido integrar el paradigma SDN a las redes de sensores inalámbricos industriales, donde se requieren requisitos estrictos para su operación. Esta integración ha sido validada mediante simulaciones y testbeds, evidenciando su eficacia para abordar de manera efectiva y sencilla desafíos tradicionales en las IWSN, tales como la calidad de servicio, la flexibilidad, la movilidad y la escalabilidad. Durante las diversas simulaciones y testbeds se ha demostrado cómo, al combinar SDN con un control de acceso al medio basado en TDMA, se obtiene un control más preciso de las transmisiones, resaltando la gestión centralizada del controlador SDN y su capacidad para orquestrar dinámicamente dispositivos. Así, se ha conseguido garantizar un comportamiento determinista en toda la red IWSN.

Además, este control total de las transmisiones y del acceso al medio ha habilitado la aplicación de slicing, segmentando la red en flujos que tienen una asignación precisa de recursos, garantizando requisitos de calidad de servicio independientes, incluso en redes con múltiples flujos. El enfoque de slicing y la asignación centralizada de recursos, también ha permitido afrontar el reto de la movilidad de los nodos. Al extender su funcionamiento a las transmisiones multicast, se ha conseguido que los cambios de receptor durante el desplazamiento sean completamente transparentes para el nodo móvil. Esto ha eliminado la necesidad de procesos de reconexión y convergencia, asegurando así la calidad de servicio y reduciendo la dependencia del tráfico de control necesario para actualizar la posición de los nodos móviles.

Finalmente, a través de los diferentes capítulos de la tesis se ha observado como las limitaciones de las redes 802.15.4 en general, derivan de la incapacidad para realizar transmisiones o recepciones simultáneas, especialmente en el nodo sink, donde se concentra de forma inevitable el tráfico de la red. Por ello, se ha propuesto la agregación de múltiples interfaces radio en el nodo sink, orquestados por el controlador SDN. Esto ha permitido aumentar el ancho de banda y habilitar operaciones de recepción de manera simultánea a través de la multiplexación del canal. De esta manera, se ha logrado disminuir el impacto del tráfico de control de SDN en la red y, por lo tanto, aumentar el número de nodos en la red sin comprometer las garantías de calidad de servicio.

Los resultados presentados en esta tesis han demostrado la capacidad de las tecnologías emergentes, como SDN y Slicing, para abordar desafíos fundamentales en las IWSN basadas en el estándar IEEE 802.15.4e. A través de la combinación de enfoque centralizado y la aplicación de protocolos de enrutamiento y planificación mejorados, se ha conseguido garantizar una calidad de servicio óptima para diferentes flujos de tráfico, así como una gestión eficaz de los recursos de la red. Además, se ha enfrentado con éxito el reto de la movilidad de los nodos y la escalabilidad, demostrando el potencial de estas innovaciones para transformar y mejorar el rendimiento y la versatilidad de las IWSN. No obstante, existen diversas áreas de mejora que pueden ser abordadas en futuras investigaciones para optimizar aún más estas redes y maximizar su rendimiento. Como es el caso de:

- Aplicación de sistemas de telemetría en banda (Inband-telemetry) para obtener información en tiempo real sobre el estado de la red sin generar tráfico adicional. Con esta información más precisa, se podrían desarrollar estrategias avanzadas, como la identificación y bloqueo de canales con bajo rendimiento o la detección temprana de fallos en la red.
- Explorar el uso de aplicaciones de aprendizaje automático para interactuar con el controlador SDN. Esto permitirá mejorar el rendimiento y la escalabilidad en enfoques de múltiples redes SD-IWSN, mediante el análisis de los canales utilizados por cada red y su rendimiento. Con esta información, se podrán seleccionar los mejores canales para cada red y optimizar la segmentación de las redes, eligiendo la configuración óptima del canal para cada nodo.
- Explorar la implementación de mecanismos que faciliten la detección, el cambio y la selección automática de redes óptimas para cada nodo, lo cual tendría como resultado una mejora en la eficiencia y la escalabilidad de las redes en su conjunto. Este enfoque también incrementaría la flexibilidad de las SD-IWSN al permitir que los nodos se desplacen entre diversas redes en función del rendimiento que alcancen en cada una de ellas.
- Estudiar la posibilidad de desarrollar soluciones de integración y cooperación entre las redes SD-IWSN y otras tecnologías de comunicación inalámbrica de baja potencia, como Bluetooth Low Energy (BLE) y LoRa, para aprovechar al máximo sus ventajas complementarias y ofrecer soluciones de conectividad más versátiles y eficientes.
- La integración de redes deterministas cableadas como Time-Sensitive Networking (TSN), con redes SD-IWSN para lograr una reserva de recursos de extremo a extremo, garantizando la calidad de servicio y la sincronización temporal en entornos heterogéneos que combinen infraestructuras cableadas e inalámbricas

Estos temas adicionales reflejan la diversidad y la amplitud de las oportunidades para futuras investigaciones en el campo de las redes SD-IWSN, enfocándose en la innovación y la mejora continua del rendimiento, la eficiencia y la adaptabilidad de estas redes en entornos industriales.

Bibliografía

- [1] H. Cao, M. Wachowicz, C. Renso, and E. Carlini, “Analytics Everywhere: Generating Insights From the Internet of Things,” *IEEE Access*, vol. 7, pp. 71749–71769, 2019.
- [2] M. Ebrahimi, A. Baboli, and E. Rother, “The evolution of world class manufacturing toward Industry 4.0: A case study in the automotive industry,” *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 188–194, 2019. 13th IFAC Workshop on Intelligent Manufacturing Systems IMS 2019.
- [3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial Internet of Things: Challenges, Opportunities, and Directions,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.
- [4] M. Tabaa, B. Chouri, S. Saadaoui, and K. Alami, “Industrial Communication based on Modbus and Node-RED,” *Procedia Computer Science*, vol. 130, pp. 583–588, 2018. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.
- [5] M. Ehrlich, L. Wisniewski, and J. Jasperneite, “State of the Art and Future Applications of Industrial Wireless Sensor Networks,” in *Kommunikation und Bildverarbeitung in der Automation* (J. Jasperneite and V. Lohweg, eds.), (Berlin, Heidelberg), pp. 28–39, Springer Berlin Heidelberg, 2018.
- [6] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-Defined Networking: A Comprehensive Survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
- [7] T. Luo, H.-P. Tan, and T. Q. S. Quek, “Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks,” *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.
- [8] P. M. Egidius, A. M. Abu-Mahfouz, and G. P. Hancke, “Programmable Node in Software-Defined Wireless Sensor Networks: A Review,” in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4672–4677, 2018.

-
- [9] “IEEE Standard for Low-Rate Wireless Networks,” *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, 2020.
- [10] J. D. Adriano, E. C. d. Rosario, and J. J. Rodrigues, “Wireless Sensor Networks in Industry 4.0: WirelessHART and ISA100.11a,” in *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 924–929, 2018.
- [11] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, “On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3655–3666, 2013.
- [12] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, “A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks,” in *2016 IEEE Wireless Communications and Networking Conference*, pp. 1–6, 2016.
- [13] K. Monteiro, E. Rocha, E. Silva, G. L. Santos, W. Santos, and P. T. Endo, “Developing an e-Health System Based on IoT, Fog and Cloud Computing,” in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, pp. 17–18, 2018.
- [14] H. Ali, U. U. Tariq, X. Zhai, and L. Liu, “Energy Efficient Task Mapping & Scheduling on Heterogeneous NoC-MPSoCs in IoT Based Smart City,” in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 1305–1313, 2018.
- [15] “IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),” *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011.
- [16] “IEEE Standard for Low-Rate Wireless Networks - Redline,” *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015) - Redline*, pp. 1–1294, 2020.
- [17] S. Soijoyo and A. Ashari, “Analysis of Zigbee Data Transmission on Wireless Sensor Network Topology,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 9, 2017.
- [18] P. D. P. Adi, V. Sihombing, V. M. M. Siregar, G. J. Yanris, F. A. Sianturi, W. Purba, S. P. Tamba, J. Simatupang, R. Arifuddin, Subairi, and D. A. Prasetya, “A Performance Evaluation of ZigBee Mesh Communication on the Internet of Things (IoT),” in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pp. 7–13, 2021.
- [19] J. W. Hui and D. E. Culler, “Extending IP to Low-Power, Wireless Personal Area Networks,” *IEEE Internet Computing*, vol. 12, no. 4, pp. 37–45, 2008.

-
- [20] W. Rzepecki, L. Iwanecki, and P. Ryba, “IEEE 802.15.4 Thread Mesh Network – Data Transmission in Harsh Environment,” in *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pp. 42–47, 2018.
- [21] W. Rzepecki and P. Ryba, “IoTSP: Thread Mesh vs Other Widely used Wireless Protocols – Comparison and use Cases Study,” in *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 291–295, 2019.
- [22] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, “WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control,” in *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 377–386, 2008.
- [23] FieldComm Group, *WirelessHART Specification*, 2018.
- [24] I. Natgunanathan, N. Fernando, S. W. Loke, and C. Weerasuriya, “Bluetooth Low Energy Mesh: Applications, Considerations and Current State-of-the-Art,” *Sensors*, vol. 23, no. 4, 2023.
- [25] P. K. D. Pramanik, B. Mukherjee, S. Pal, B. K. Upadhyaya, and S. Dutta, *Ubiquitous Manufacturing in the Age of Industry 4.0: A State-of-the-Art Primer*, pp. 73–112. Cham: Springer International Publishing, 2020.
- [26] D. Aureli, A. Cianfrani, A. Diamanti, J. M. Sanchez Vilchez, and S. Secci, “Going Beyond DiffServ in IP Traffic Classification,” in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6, 2020.
- [27] S. Saadaoui, M. Tabaa, F. Monteiro, M. Chehaitly, A. Dandache, and A. Oukaira, “IWSN under an industrial wireless channel in the context of Industry 4.0,” in *2017 29th International Conference on Microelectronics (ICM)*, pp. 1–4, 2017.
- [28] A. A. Kumar S., K. Ovsthus, and L. M. Kristensen., “An Industrial Perspective on Wireless Sensor Networks — A Survey of Requirements, Protocols, and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1391–1412, 2014.
- [29] S. Kharb and A. Singhrova, “A survey on network formation and scheduling algorithms for time slotted channel hopping in industrial networks,” *Journal of Network and Computer Applications*, vol. 126, pp. 59–87, 2019.
- [30] H.-T. Wu, K.-W. Ke, S.-F. Wang, P.-H. Chen, G.-D. Lee, C.-Y. Tseng, and C.-Y. Ho, “The Implementation of Wireless Industrial Internet of Things (IIoT) Based Upon IEEE 802.15.4-2015 TSCH Access Mode,” in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, pp. 367–369, 2019.

-
- [31] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, “Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks,” in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, pp. 327–332, 2012.
- [32] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, “Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things,” in *2013 IEEE 14th International Symposium on .A World of Wireless, Mobile and Multimedia Networks”(WoWMoM)*, pp. 1–6, 2013.
- [33] R. Soua, P. Minet, and E. Livolant, “Wave: A Distributed Scheduling Algorithm for Convergecast in IEEE 802.15.4e Networks (Extended Version),” *Transactions on Emerging Telecommunications Technologies*, vol. 27, 10 2015.
- [34] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH,” in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys ’15*, (New York, NY, USA), p. 337–350, Association for Computing Machinery, 2015.
- [35] S. Oh, D. Hwang, K.-H. Kim, and K. Kim, “Escalator: An Autonomous Scheduling Scheme for Convergecast in TSCH ,” *Sensors*, vol. 18, p. 1209, 04 2018.
- [36] S. Kim, H.-S. Kim, and C. Kim, “ALICE: Autonomous Link-based Cell Scheduling for TSCH,” in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 121–132, 2019.
- [37] M. Karakus and A. Durresi, “Quality of Service (QoS) in Software Defined Networking (SDN): A survey,” *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017.
- [38] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, “QoS-Aware Adaptive Routing in Multi-layer Hierarchical Software Defined Networks: A Reinforcement Learning Approach,” in *2016 IEEE International Conference on Services Computing (SCC)*, pp. 25–33, 2016.
- [39] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, “A roadmap for traffic engineering in SDN-OpenFlow networks,” *Computer Networks*, vol. 71, pp. 1–30, 2014.
- [40] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, “Software Defined Wireless Networks: Unbridling SDNs,” in *2012 European Workshop on Software Defined Networking*, pp. 1–6, 2012.
- [41] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, “A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements,” *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

-
- [42] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. El-Bakoury, “Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.
- [43] I. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah, “A Survey on Cross-Layer Quality-of-Service Approaches in WSNs for Delay and Reliability-Aware Applications,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 525–552, 2016.
- [44] B. B. Letswamotse, R. Malekian, C.-Y. Chen, and K. M. Modieginnyane, “Software Defined Wireless Sensor Networks (SDWSN): A Review on Efficient Resources, Applications and Technologies,” *Journal of Internet Technology*, vol. 19, pp. 1303–1313, 2018.
- [45] B. Trevizan de Oliveira, L. Batista Gabriel, and C. Borges Margi, “TinySDN: Enabling Multiple Controllers for Software-Defined Wireless Sensor Networks,” *IEEE Latin America Transactions*, vol. 13, no. 11, pp. 3690–3696, 2015.
- [46] M. Baddeley, R. Nejabati, G. Oikonomou, S. Gormus, M. Sooriyabandara, and D. Simeonidou, “Isolating SDN control traffic with layer-2 slicing in 6TiSCH industrial IoT networks,” in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 247–251, 2017.
- [47] M. Baddeley, U. Raza, A. Stanoev, G. Oikonomou, R. Nejabati, M. Sooriyabandara, and D. Simeonidou, “Atomic-SDN: Is Synchronous Flooding the Solution to Software-Defined Networking in IoT?,” *IEEE Access*, vol. 7, pp. 96019–96034, 2019.
- [48] T. Theodorou and L. Mamatas, “CORAL-SDN: A software-defined networking solution for the Internet of Things,” in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 1–2, 2017.
- [49] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, “SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for Wireless Sensor networks,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 513–521, 2015.
- [50] A.-C. Anadiotis, L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, “SD-WISE: A Software-Defined Wireless Sensor network,” *Computer Networks*, vol. 159, pp. 84–95, 2019.
- [51] P. Di Dio, S. Faraci, L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, and P. Livreri, “Exploiting state information to support QoS in Software-Defined WSNs,” in *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 1–7, 2016.
- [52] L. L. Bello, A. Lombardo, S. Milardo, G. Patti, and M. Reno, “Software-Defined Networking for Dynamic Control of Mobile Industrial Wireless Sensor Networks,” in *2018*

- IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 290–296, 2018.
- [53] B. Safaei, A. Mohammadsalehi, K. T. Khoosani, S. Zarbaf, A. M. H. Monazzah, F. Samie, L. Bauer, J. Henkel, and A. Ejlali, “Impacts of Mobility Models on RPL-Based Mobile IoT Infrastructures: An Evaluative Comparison and Survey,” *IEEE Access*, vol. 8, pp. 167779–167829, 2020.
- [54] C. Cobârzan, J. Montavont, and T. Noël, “Analysis and performance evaluation of RPL under mobility,” in *2014 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, 2014.
- [55] S. Raza, T. v. d. Lee, G. Exarchakos, and M. Güneş, “A Reliability Analysis of TSCH Protocol in a Mobile Scenario,” in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, 2019.
- [56] Y. Al-Nidawi and A. H. Kemp, “Mobility Aware Framework for Timeslotted Channel Hopping IEEE 802.15.4e Sensor Networks,” *IEEE Sensors Journal*, vol. 15, no. 12, pp. 7112–7125, 2015.
- [57] O. Gaddour, A. Koubäa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, “Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism,” in *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)*, pp. 200–209, 2014.
- [58] M. Barcelo, A. Correa, J. L. Vicario, A. Morell, and X. Vilajosana, “Addressing Mobility in RPL With Position Assisted Metrics,” *IEEE Sensors Journal*, vol. 16, no. 7, pp. 2151–2161, 2016.
- [59] O. Tavallaie, J. Taheri, and A. Y. Zomaya, “Design and Optimization of Traffic-Aware TSCH Scheduling for Mobile 6TiSCH Networks,” in *Proceedings of the International Conference on Internet-of-Things Design and Implementation, IoTDI '21*, (New York, NY, USA), p. 234–246, Association for Computing Machinery, 2021.
- [60] A. Demir and S. Bilgili, “DIGEST: a decentralized divergecast scheduling algorithm for IEEE 802.15.4e TSCH in the internet of things,” *Wireless Networks*, vol. 27, pp. 1–16, 10 2021.
- [61] F. Orozco-Santos, V. Sempere-Payá, T. Albero-Albero, and J. Silvestre-Blanes, “Enhancing SDN WISE with Slicing Over TSCH,” *Sensors*, vol. 21, p. 1075, 02 2021.
- [62] L. L. Bello, A. Lombardo, S. Milardo, G. Patti, and M. Reno, “Experimental Assessments and Analysis of an SDN Framework to Integrate Mobility Management in Industrial Wireless Sensor Networks,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5586–5595, 2020.

-
- [63] A. Karaagac, I. Moerman, and J. Hoebeke, “Hybrid Schedule Management in 6TiSCH Networks: The Coexistence of Determinism and Flexibility,” *IEEE Access*, vol. 6, pp. 33941–33952, 2018.
- [64] A. Karaagac, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Time-critical communication in 6TiSCH networks,” in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 161–166, 2018.
- [65] P. Thubert, M. R. Palattella, and T. Engel, “6TiSCH centralized scheduling: When SDN meet IoT,” in *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 42–47, 2015.
- [66] B. Varga, J. Farkas, R. Cummings, Y. Jiang, and D. Fedyk, “Flow and Service Information Model for Deterministic Networking (DetNet).” RFC 9016, Mar. 2021.
- [67] A. Sarkar and T. Senthil Murugan, “Routing protocols for wireless sensor networks: What the literature says?,” *Alexandria Engineering Journal*, vol. 55, no. 4, pp. 3173–3183, 2016.
- [68] C. Lim, “A Survey on Congestion Control for RPL-Based Wireless Sensor Networks,” *Sensors*, vol. 19, no. 11, 2019.
- [69] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, “The Contiki-NG open source operating system for next generation IoT devices,” *SoftwareX*, vol. 18, p. 101089, 2022.
- [70] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, “Fast join and synchronization schema in the IEEE 802.15.4e MAC,” in *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 85–90, 2015.
- [71] B.-H. Bae and S.-H. Chung, “Fast Synchronization Scheme Using 2-Way Parallel Rendezvous in IEEE 802.15.4 TSCH,” *Sensors*, vol. 20, no. 5, 2020.
- [72] M. Mohamadi, B. Djamaa, and M. R. Senouci, “Performance evaluation of TSCH-minimal and orchestra scheduling in IEEE 802.15.4e networks,” in *2018 International Symposium on Programming and Systems (ISPS)*, pp. 1–6, 2018.
- [73] S. Rekik, N. Baccour, M. Jmaiel, K. Drira, and L. A. Grieco, “Autonomous and traffic-aware scheduling for TSCH networks,” *Computer Networks*, vol. 135, pp. 201–212, 2018.
- [74] R. Teles Hermeto, A. Gallais, and F. Theoleyre, “Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey,” *Computer Communications*, vol. 114, pp. 84–105, 2017.

-
- [75] T. van der Lee, G. Exarchakos, and A. Liotta, “Distributed TSCH scheduling: A comparative analysis,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3517–3522, 2017.
- [76] E. T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. Dujovne, “6TiSCH Minimal Scheduling Function (MSF),” tech. rep., Internet Engineering Task Force (IETF), 2021.
- [77] R.-H. Hwang, C.-C. Wang, and W.-B. Wang, “A Distributed Scheduling Algorithm for IEEE 802.15.4e Wireless Sensor Networks,” *Computer Standards & Interfaces*, vol. 52, pp. 63–70, 2017.
- [78] S. Duquennoy, A. Elsts, B. A. Nahas, and G. Oikonomo, “TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation,” in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 11–18, 2017.
- [79] M. Ojo and S. Giordano, “An efficient centralized scheduling algorithm in IEEE 802.15.4e TSCH networks,” in *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1–6, 2016.
- [80] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. V. Herwegen, and W. Vantomme, “The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2740–2751, 2008.
- [81] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan, “A Critical Analysis of Research Potential, Challenges, and Future Directives in Industrial Wireless Sensor Networks,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 39–95, 2018.
- [82] M. Baddeley, R. Nejabati, G. Oikonomou, M. Sooriyabandara, and D. Simeonidou, “Evolving SDN for Low-Power IoT Networks,” in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 71–79, 2018.
- [83] K. Singh, “WSN LEACH based protocols: A structural analysis,” in *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, pp. 1–7, 2015.
- [84] E. F. Ahmed Elsmay, M. A. Omar, T.-C. Wan, and A. A. Altahir, “EESRA: Energy Efficient Scalable Routing Algorithm for Wireless Sensor Networks,” *IEEE Access*, vol. 7, pp. 96974–96983, 2019.
- [85] A. Shukla and S. Tripathi, “An Effective Relay Node Selection Technique for Energy Efficient WSN-Assisted IoT,” *Wireless Personal Communications*, vol. 112, 06 2020.

-
- [86] L. L. Bello, A. Lombardo, G. Patti, and M. Reno, "A Platform for Evaluating Clustering Strategies in Mobile IEEE 802.15. 4-TSCH networks," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 987–992, IEEE, 2019.
- [87] M. Sajwan, D. Gosain, and A. K. Sharma, "Hybrid energy-efficient multi-path routing for wireless sensor networks," *Computers & Electrical Engineering*, vol. 67, pp. 96–113, 2018.
- [88] G. Shabbir, A. Akram, M. M. Iqbal, S. Jabbar, M. Alfawair, and J. Chaudhry, "Network Performance Enhancement of Multi-sink Enabled Low Power Lossy Networks in SDN Based Internet of Things," *International Journal of Parallel Programming*, vol. 48, pp. 367–398, dec 2018.
- [89] F. Veisi, M. Nabi, and H. Saidi, "Coexistence analysis of multiple asynchronous IEEE 802.15. 4 TSCH-based networks," *IEEE Access*, vol. 8, pp. 150573–150585, 2020.
- [90] R. C. A. Alves, D. A. G. Oliveira, G. A. Nunez Segura, and C. B. Margi, "The Cost of Software-Defining Things: A Scalability Study of Software-Defined Sensor Networks," *IEEE Access*, vol. 7, pp. 115093–115108, 2019.
- [91] F. F. Jurado-Lasso, K. Clarke, and A. Nirmalathas, "Performance Analysis of Software-Defined Multihop Wireless Sensor Networks," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4653–4662, 2020.
- [92] A. Abuarqoub, "A Review of the Control Plane Scalability Approaches in Software Defined Networking," *Future Internet*, vol. 12, no. 3, 2020.
- [93] O. Flauzac, C. Javier Gonzalez Santamaria, F. Nolot, and I. Woungang, "An SDN approach to route massive data flows of sensor networks," *International Journal of Communication Systems*, vol. 33, no. 7, p. e4309, 2020.
- [94] S. Tahmasebi, N. Rasouli, A. H. Kashefi, E. Rezaeyk, and H. R. Faragardi, "SYNCOP: An evolutionary multi-objective placement of SDN controllers for optimizing cost and network performance in WSNs," *Computer Networks*, vol. 185, p. 107727, 2021.
- [95] E. Municio, K. Spaey, and S. Latré, "A distributed density optimized scheduling function for IEEE 802.15.4e TSCH networks," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 7, p. e3420, 2018.
- [96] E. Municio and S. Latré, "Decentralized broadcast-based scheduling for dense multi-hop TSCH networks," in *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*, pp. 19–24, 2016.

-
- [97] J. C. da Silva and F. Assis, “A distributed algorithm to schedule TSCH links under the SINR model,” *Design Automation for Embedded Systems*, vol. 23, no. 1, pp. 21–39, 2019.
- [98] J. Vera-Pérez, D. Todolí-Ferrandis, S. Santonja-Climent, J. Silvestre-Blanes, and V. Sempere-Payá, “A Joining Procedure and Synchronization for TSCH-RPL Wireless Sensor Networks,” *Sensors*, vol. 18, no. 10, 2018.
- [99] A. R. Urke, Ø. Kure, and K. Øvsthus, “A Survey of 802.15.4 TSCH Schedulers for a Standardized Industrial Internet of Things,” *Sensors*, vol. 22, p. 15, 12 2021.