



Ph.D. Program in Information and
Communications Technologies

**Contribución a tecnologías
habilitantes en redes programables y
definidas por software**

Ph.D. Dissertation
David Carrascal Acebron

2025



Universidad
de Alcalá

Ph.D. Program in Information and
Communications Technologies

Contribución a tecnologías
habilitantes en redes programables y
definidas por software

Author

David Carrascal Acebron

Advisors

Dr. Elisa Rojas Sánchez

Dr. Diego López Pajares

Alcalá de Henares

2025

*“No os preguntarán por mí,
que en estos tiempos a nadie le da lustre
haber nacido segundón en casa grande;
pero si pregunta alguno,
bueno será contestarle
que, español, a toda vena,
amé, reñí, di mi sangre,
pensé poco, recé mucho,
jugué bien, perdí bastante.*

*Y, porque esa empresa loca que nunca debió tentarme,
que, perdiendo ofende a todos,
que, triunfando alcanza a nadie,
no quise salir del mundo
sin poner mi pica en Flandes.*

¡Por España!

*Y el que quiera defenderla honrado muera.
Y el traidor que la abandone,
no tenga quien le perdone,
ni en Tierra Santa cobijo,
ni una cruz en sus despojos,
ni las manos de un buen hijo,
para cerrarle los ojos.”*

HERNANDO DE ACUÑA

Este trabajo ha sido parcialmente financiado por la Universidad de Alcalá a través del programa FPI-UAH 2022; por la Comunidad de Madrid mediante los proyectos TAPIR-CM (S2018/TCS-4496), MistLETOE-CM (CM/JIN/2021-006), TUCAN6-CM (TEC-2024/COM-460) y VERANO (CM/DEMG/2024-038); por el MICIU y la Unión Europea NextGenerationEU/PRTR a través del proyecto ADMINISTER (TED2021-131301B-I00); por el Ministerio de Ciencia e Innovación mediante el proyecto ONENESS (PID2020-116361RA-I00); y por el Ministerio de Asuntos Económicos y Transformación Digital y la Unión Europea–NextGenerationEU a través del proyecto UNICO 5G I+D 6G-DATADRIVEN-02, coordinado por la Universidad Carlos III de Madrid.

Agradecimientos

Es increíble cómo pasa el tiempo, y da vértigo echar la vista atrás. Pensar en cómo he llegado hasta aquí, las conferencias a las que he tenido la suerte de asistir, los viajes que he podido realizar, y, sobre todo, en las personas que he conocido por el camino. A menudo se describe el doctorado como un proceso arduo y exigente, y no diré lo contrario, pero, también merece la pena poner en valor todo lo aprendido, lo vivido y las experiencias que me han hecho crecer tanto a nivel personal como profesional.

Escribo estas líneas desde Milán, durante mi estancia doctoral junto a Marco Savi, a quien quiero agradecer sinceramente todo el tiempo, dedicación y esfuerzo que me ha brindado. Su apoyo ha hecho que esta etapa sea verdaderamente inolvidable. Esta experiencia no habría sido la misma sin todas las personas maravillosas que he conocido en Italia, que me han hecho sentir como en casa, siempre con una sonrisa y dispuestas a ayudar.

Volviendo la mirada a casa, no puedo dejar de agradecer a mis amigos, que han estado siempre a mi lado, apoyándome en cada paso. A mi familia, por su paciencia infinita, por su ayuda constante y por soportarme en los momentos de estrés y agobio de esta “empresa loca” que es el doctorado. A mis amigos del LE-34, los que están y los que ya no están, con quienes he compartido tantos cafés, confidencias, comidas y tardes de “trabajo”. Y, por supuesto, a todos mis compañeros del grupo NetIS, que tras tantos años, más que colegas se han convertido en una segunda familia. Gracias por creer en mí y por acompañarme hasta aquí.

Por último, a Elisa y Diego, mis directores de tesis. Gracias por vuestra guía, vuestro apoyo incondicional y por confiar en mí desde el primer momento. En especial a Elisa: si no fuera por aquel correo que me enviaste hace ya más de siete años, ofreciéndome una mini beca de investigación, hoy no estaría escribiendo estas líneas. Gracias por todo lo que me habéis enseñado, por lo que he aprendido con vosotros y por todo lo que aún me queda por aprender.

Sinceramente, mil gracias a todos.

Resumen

La era contemporánea en la cual vivimos se caracteriza en mayor medida por una profunda transformación tecnológica y social, impulsada por la globalización y por el desarrollo de infraestructuras digitales interconectadas que configuran lo que se conoce como el Internet of Everything (IoE). En este nuevo paradigma, emergen redes densas y altamente heterogéneas en las que convergen dispositivos, servicios y plataformas con requerimientos funcionales muy variados, integrando no solo redes de comunicaciones, sino también infraestructuras energéticas, industriales y logísticas. Esta complejidad creciente demanda nuevas metodologías para un control y gestión, que sea en la medida de lo posible, flexible y escalable. En este contexto, las redes softwarizadas y programables se postulan como un elemento tecnológico clave, al permitir una abstracción funcional de la infraestructura subyacente, facilitar su automatización y promover la integración de capacidades de control, así como, la adaptación dinámica de la red, a las necesidades intrínsecas de los nodos de la misma.

Esta Tesis contribuye al desarrollo de las redes softwarizadas y programables mediante la propuesta de soluciones orientadas a mejorar la gestión, la resiliencia y la cooperación entre nodos de dichas redes. En primer lugar, se diseñan y evalúan algoritmos de toma de decisiones inteligentes en entornos altamente dinámicos y heterogéneos, con aplicación en dominios como el Internet de las Cosas Industrial (IIoT), las redes eléctricas inteligentes (Smart Grids) y las redes de comunicaciones de nueva generación. Estas soluciones permiten una asignación dinámica de recursos, una adaptación proactiva a las condiciones del entorno y una reducción sustancial en la complejidad de gestión de la red. Además, se ha abordado la integración de modelos de Inteligencia Artificial (AI) con dichos algoritmos, con el fin de potenciar la detección temprana de fallos, lo que se traduce en una mejora significativa de la resiliencia, y la alta disponibilidad de los servicios. En segundo lugar, se propone una arquitectura software modular, orientada a servicios y alineada con estándares actuales, que permite la incorporación de herramientas emergentes, así como mecanismos automatizados con AI, ofreciendo capacidades de computación tanto en la nube y como en el edge. Esta arquitectura está concebida para proporcionar una infraestructura lógica robusta, interoperable y escalable, capaz de facilitar la orquestación autónoma de servicios distribuidos, orientado a contextos de elevada heterogeneidad tecnológica, como entornos IIoT u otros.

Palabras clave: Redes densas y heterogeneas, Redes programables y softwarizadas, Algoritmos, Infraestructura Cloud, IoT industrial, Smart grids.

Abstract

The contemporary era is marked by a profound technological and social transformation, driven by globalization and the pervasive deployment of interconnected digital infrastructures that define the Internet of Everything (IoE). Within this emerging paradigm, dense and highly heterogeneous networks are formed, comprising a wide variety of devices, services, and platforms with diverse and demanding functional requirements. These systems integrate not only communication networks, but also energy, industrial, and logistics infrastructures. The increasing complexity of such environments necessitates the adoption of novel methodologies capable of ensuring the flexible and scalable control and management of distributed resources and services. In this context, softwarized and programmable networks have emerged as a pivotal technological solution, enabling functional abstraction of the underlying infrastructure, supporting automation, and fostering the integration of advanced control mechanisms, as well as the dynamic adaptation of network behavior to the intrinsic requirements of its nodes.

This Thesis contributes to the advancement of softwarized and programmable networking by proposing solutions designed to enhance the management, resilience, and cooperation between nodes within these infrastructures. Firstly, it presents and evaluates intelligent decision-making algorithms tailored for highly dynamic and heterogeneous environments, with applications in domains such as the Industrial Internet of Things (IIoT), smart grids, and next-generation communication networks. These algorithms facilitate dynamic resource allocation, proactive environmental adaptation, and a significant reduction in network management complexity. Furthermore, the integration of Artificial Intelligence (AI) models into these solutions is explored to enable early fault detection, thus improving system resilience and ensuring high service availability. Secondly, the Thesis proposes a modular, service-oriented software architecture, aligned with current standards and capable of incorporating emerging technologies and AI-driven automation mechanisms. This architecture offers computing capabilities across both cloud and edge infrastructures and is designed to deliver a robust, interoperable, and scalable logical platform that supports the autonomous orchestration of distributed services in technologically heterogeneous scenarios such as IIoT environments.

Keywords: Dense and heterogeneous networks, Programmable and softwarized networks, Algorithms, Cloud infrastructure, Industrial IoT, Smart Grids.

Índice general

Resumen	VII
Abstract	IX
Índice general	XI
Índice de figuras	XIII
Índice de tablas	XV
Índice de algoritmos	XVII
Lista de acrónimos	XIX
1. Introducción y objetivos	1
1.1. Introducción	1
1.2. Redes programables y definidas por software	2
1.3. Planteamiento del problema y objetivos de la tesis	4
1.4. Estructura de la tesis	6
1.5. Contribuciones	6
2. Estado del Arte	9
2.1. Las redes SDN	9
2.1.1. Arquitectura lógica de las redes SDN	10
2.1.2. Arquitectura física de las redes SDN	16
2.1.3. Propuestas de despliegue con control in-band	19
2.1.4. Conclusiones y alineación con los objetivos de la Tesis	24
2.2. Servicios y Tecnologías habilitantes en redes softwarizadas y heterogéneas	25
2.2.1. Servicios básicos: arranque y provisión de canales de control en entornos densos y heterogéneos	26
2.2.1.1. Terminología básica	28
2.2.1.2. Propuestas de etiquetado jerárquico	32
2.2.1.3. Conclusiones y alineación con los objetivos de la Tesis . .	35
2.2.2. Servicios avanzados: gestión y planeamiento de recursos en entornos densos y heterogéneos	36

2.2.3. Servicios avanzados: optimización y reconfiguración proactiva del control en entornos densos y heterogéneos	36
2.3. Casos de uso	36
3. Capítulo de prueba	37
3.1. Uso de acrónimos y referencias	37
3.2. Figura de prueba	41
3.3. Tabla de prueba	41
3.4. Algoritmo de prueba	41

Índice de figuras

1.1. Diagrama general del marco general de la Tesis	5
1.2. Línea de tiempo de contribuciones científicas (2023–2025): publicaciones en revistas y conferencias internacionales.	8
2.1. Paradigma en las redes SDN	10
2.2. Arquitectura lógica de las redes SDN	11
2.3. Arquitectura básica de switch OpenFlow	13
2.4. Arquitectura básica de switch P4	15
2.5. Paradigmas de control en las redes SDN	17
2.6. Paradigmas de control <i>Management–Control Continuum</i> (MCC)	25
2.7. Elección de una topología lógica (árbol enraizado) partiendo una topología física	30
2.8. Ejemplo de difusión empleando el etiquetado jerárquico	31
3.1. Esquema de arquitectura softwarizada.	41

Índice de tablas

2.1. Evolución de versiones del protocolo Openflow y el número de campos de cabecera soportados	14
2.2. Características del control <i>in-band</i> y <i>out-of-band</i>	19
3.1. Comparativa de arquitecturas de red.	41

Índice de algoritmos

1.	Selección del nodo con mayor capacidad	42
----	--------------------------------------------------	----

Lista de acrónimos

5G	fifth-generation mobile networks.
6G	sixth-generation mobile networks.
AI	Artificial Intelligence.
API	Application Programming Interface.
ARPANET	Advanced Research Projects Agency Network.
CapEx	Capital Expenditure.
DAG	Directed Acyclic Graph.
DC	Data Center.
DHCP	Dynamic Host Configuration Protocol.
GA3	Generalized Automatic Address Assignment.
gRPC	google Remote Procedure Call.
HLMAC	Hierarchical Local MAC.
IIoT	Industrial Internet of Things.
IoT	Internet of Things.
LLDP	Link Layer Discovery Protocol.
LLNs	Low-Power and Lossy Networks.
MCC	Management–Control Continuum.
ML	Machine Learning.
MPTCP	MultiPath TCP.
NFV	Network Functions Virtualization.
NTN	Non-Terrestrial Network.
ONF	Open Networking Foundation.
OpEx	Operational Expenditure.
OVS	Open vSwitch.
OVSDB	Open vSwitch Database.
P4	Programming Protocol-Independent Packet Processors.
QoS	Quality of Service.
RPL	Low-Power and Lossy Networks.
RSTP	Rapid Spanning Tree Protocol.
SDN	Software-Defined Networking.
SG	Smart grid.
TCAM	Ternary Content-Addressable Memory.
VLAN	Virtual Local Area Network.

WAN Wide Area Network.

Capítulo 1

Introducción y objetivos

En este primer capítulo, se presenta una introducción al tema principal de la tesis, además de dar un contexto y un marco general del problema que se va a abordar. Asimismo, se establecen los objetivos de la investigación, se describe la estructura general del documento y se enumeran las contribuciones principales que ha generado esta tesis doctoral.

1.1. Introducción

Esta Tesis se enmarca dentro de las redes definidas por software (Software-Defined Networking (SDN), por sus siglas en inglés), y las redes programables, las cuales permiten la creación de redes flexibles y adaptables a las necesidades cambiantes de los usuarios y las aplicaciones finales. Este tipo de redes están ganando cada vez más importancia en la sociedad actual, dado que, con la creciente digitalización de la mayoría de los sectores, así como, tejido industrial y social, se están conformando redes cada vez más densas y heterogéneas, que requieren de nuevas tecnologías o herramientas que permitan su gestión y control. La tipología final de estas redes puede ser muy variopinta, pudiendo estar presentes en las redes de comunicaciones, en redes de sensores, en redes de distribución de energía, logística, entre otras.

Es por ello, que es difícil acotar el campo de estudio y aplicación de esta Tesis, y no solo por su naturaleza multidisciplinar, sino por la complejidad de que una herramienta o tecnología, sea completamente extrapolable a otro tipo de red. Por ejemplo, se pueden encontrar similitudes entre las necesidades de los distintos tipos de redes, en las redes de comunicaciones móviles fifth-generation mobile networks (5G) y sixth-generation mobile networks (6G), donde existen múltiples dispositivos, sensores y nodos de acceso que deben coordinarse dinámicamente para ofrecer conectividad. En el ámbito energético, donde las redes eléctricas inteligentes o Smart grids (SGs) destacan por la coordinación dinámica de la integración de fuentes de energía distribuidas, almacenamiento local y consumidores activos. También se puede ver en el campo de la logística, donde se puede considerar el caso de las redes de transporte y distribución, donde los vehículos, almacenes y sistemas de seguimiento deben coordinarse para optimizar rutas, minimizar tiempos de entrega

y reducir costes operativos. En todos estos escenarios, el uso de redes programables y softwarizadas ofrecen una base sólida para la gestión flexible y dinámica, sobre la cual se pueden desarrollar herramientas o tecnologías que optimicen cada caso de uso, consiguiendo escenarios más eficientes y adaptables a las necesidades de la red. De ahí que, la idea de esta Tesis sea la de ahondar en las tecnologías habilitantes de las redes programables y definidas por software, y cómo se pueden aplicar en redes densas y con nodos y necesidades heterogéneas.

1.2. Redes programables y definidas por software

Las redes programables tienen sus raíces en la historia de las redes de comunicación [1]. Desde su inicio, con la llegada de Advanced Research Projects Agency Network (ARPANET) en 1969, creada en Estados Unidos en un contexto de la guerra fría para que los investigadores pudieran intercambiar información, las redes de comunicación han evolucionado desde sistemas simples y estáticos, hacia arquitecturas más complejas y dinámicas. Durante este proceso, la necesidad de controlar y adaptar el comportamiento de la red ha sido un punto recurrente. Sin embargo, durante décadas, las redes tradicionales se caracterizaron por ser muy estáticas, estrechamente ligadas al hardware y al fabricante, lo que dificultaba su evolución y adaptación dinámica a nuevos protocolos o nuevas ideas. Es por ello que, la idea del SDN comenzó a gestarse en la Universidad de Stanford en 2003, cuando el profesor asociado de ese entonces, Nick McKeown, planteó las limitaciones de las redes convencionales y la necesidad de replantear cómo operaban los *backbones* [2]. En 2011, se acuñó el término SDN, al mismo tiempo que se lanzó la organización Open Networking Foundation (ONF) [3], encargada de establecer estándares y promover la difusión del SDN, la cual, a finales de 2023 fue incluida en la Linux Foundation para salvaguardar y reafirmar los proyectos y propuestas *open-source* en materia de *Networking*.

El paradigma SDN [4] radica en un concepto de arquitectura de red, en la que, se separa el plano de control (mantenimiento, gestión y control de la red) del plano de datos (lógica de *forwarding*) de la red, para centralizar toda la lógica de control en un único ente, el cual se denomina como controlador. Esta estructura permite lograr una administración de red más centralizada y flexible [4], facilitando la programabilidad de la red y la implementación de herramientas auxiliares que operan directamente sobre la Application Programming Interface (API) que expone el controlador. Esta API de alto nivel es clave en la integración de cualquier tipo de herramienta, desde monitorización, a Quality of Service (QoS), o incluso de modelos de Artificial Intelligence (AI)/Machine Learning (ML) para predecir o reconfigurar la red de forma automática.

Este paradigma se empezó a popularizar entre las grandes operadoras de telecomunicaciones, que junto a la tecnología de virtualización de funciones de red (del inglés, Network Functions Virtualization (NFV)) [5], podían desplegar, mantener y gestionar los servicios de red de forma dinámica y escalable, permitiendo al administrador de red operar desde un único ente de control, toda la infraestructura. Grandes empresas como Google [6], NTT [7], IBM [8] o Telefónica [9] han contribuido activamente al desarrollo y adopción

de estas tecnologías. Este apoyo del sector tecnológico al despliegue de redes SDN con NFV se ve impulsado por la reutilización de hardware para el despliegue ágil de nuevos servicios y aplicaciones, lo que permite reducir significativamente el gasto en capital (del inglés, Capital Expenditure (CapEx)), así como la posibilidad de operar y gestionar la infraestructura de forma centralizada y programable, lo que se traduce en una disminución del gasto operativo (del inglés, Operational Expenditure (OpEx)).

Sin embargo, las redes softwarizadas y programables no se limitan únicamente a las redes de telecomunicaciones. En los últimos años, se ha podido ver cómo este paradigma se ha extendido a otros campos, como, por ejemplo, en las redes de sensores, donde la flexibilidad y la versatilidad son claves para optimizar el rendimiento de los equipos, que por lo general suelen tener recursos limitados. También se han visto integraciones de ecosistemas SDN en el ámbito de las redes de sensores Internet of Things (IoT) [10], donde se provee de una pila de protocolos que permite la interoperabilidad entre los equipos y controladores convencionales, pudiendo traer todos los avances de las redes de telecomunicación a entornos más agresivos, como por ejemplo el Industrial Internet of Things (IIoT) [11].

Incluso, se ha llegado a ver el uso de las redes programables en el ámbito de la distribución/encaminamiento de energía, donde la integración de fuentes de energía renovables y la gestión de la demanda dinámica requieren una coordinación meticulosa [12]. Históricamente las redes eléctricas de todos los países se han ido conformando en un modelo de top-to-down, donde las grandes centrales eléctricas generaban la energía, y esta se distribuía a los usuarios finales. Con el creciente aumento de la población, la red eléctrica se iba expandiendo y ramificando, creando redes en modo árbol cada vez más densas desde los puntos de interconexión. Sin embargo, con la llegada de las energías renovables y el cambio normativo promovido por la Unión Europea, este modelo tradicional ha comenzado a transformarse. La Directiva 2018/2001/UE sobre energías renovables [13], establece un marco regulador que permite a los ciudadanos y empresas convertirse en productores de energía (prosumidores), facilitando no solo el autoconsumo, sino también la posibilidad de inyectar el excedente energético a la red eléctrica. En España, esta directiva se materializó a través del Real Decreto 244/2019 [14], que regula el autoconsumo eléctrico y permite compensar económicamente la energía excedentaria. Por lo tanto, se ha pasado a un modelo de red eléctrica, top-to-down, a uno más distribuido y con múltiples puntos de generación, donde los usuarios finales pueden ser tanto consumidores como productores de la energía, haciendo que la red requiera de una mayor flexibilidad y adaptabilidad para gestionar este intercambio de dinámico de energía. Esta necesidad ha llevado a la creación de redes eléctricas inteligentes (SG), y a la incorporación de tecnologías de redes programables, que permitan la gestión dinámica de la energía. Estándares como el IEC 61850 [15] han sido claves en el desarrollo para facilitar la interoperabilidad y la comunicación entre diferentes subestaciones y dispositivos dentro de una SG, permitiendo integraciones con soluciones SDN [16, 17].

De forma similar, las redes de logística y transporte también han comenzado a adoptar modelos de redes softwarizadas [18], para optimizar la gestión de flotas, mejorando la eficiencia operativa. Por lo tanto, de forma analoga y sistemática se podría ir sector por

sector, viendo como en cada campo de aplicación se van dando redes densas y heterogéneas, que requieren de un enfoque programables, dando lugar a esta línea de investigación y desarrollo que se aborda en esta Tesis. A continuación, se presenta el planteamiento del problema y los objetivos de la Tesis, donde se aterriza cómo se va a abordar el estudio de las redes programables y softwarizadas, y en qué campos de aplicación se va a trabajar.

1.3. Planteamiento del problema y objetivos de la tesis

Los objetivos que se plantean en el desarrollo de la Tesis se pueden dividir en dos grandes bloques. Por un lado, se busca profundizar en el estudio de las redes programables, partiendo del escenario base de las redes SDN, y con la idea de extender este paradigma a otros campos de aplicación, como son las redes de sensores IIoT y las redes de distribución de energía. En particular, se busca profundizar en los mecanismos de control empleados en este tipo de redes, los cuales suelen organizarse en torno a dos enfoques principales: el control *out-of-band* y el control *in-band* [19]. En el modelo *out-of-band*, cada nodo de red tiene un enlace dedicado con el controlador, permitiendo una separación clara entre el plano de control y el plano de datos. Por el contrario, el modelo *in-band* asume que solo algunos nodos poseen un enlace directo con el controlador, y el resto de los dispositivos reutilizan dicho canal para transmitir información de control.

La idea de empezar profundizando este concepto radica en que, después de haber estado trabajando con ello en estudios anteriores, se ha podido ver que, en función del tipo de red y de la topología, uno u otro paradigma puede ser más adecuado, además de no haber una implementación estandarizada en el modelo *in-band*. Esto deja espacio de mejora y optimización tanto para las redes de comunicaciones, como para las redes de sensores y distribución de energía, donde este enfoque puede tener un papel importante. En redes de sensores, por ejemplo, donde los nodos suelen tener capacidades de cómputo, memoria y conectividad limitadas, implementar un enfoque *out-of-band* resulta poco viable. Asimismo, en las redes de distribución eléctrica, en las que la infraestructura sigue habitualmente una topología jerárquica de tipo árbol, no todos los nodos tienen un acceso directo al núcleo de la red, lo que hace necesario explorar soluciones basadas en el control *in-band*. Es por ello, que en el primer bloque de objetivos de la Tesis se busca profundizar en el estudio de mecanismos de control en redes densas y heterogéneas, donde el mecanismo o algoritmo pueda ser adaptado a las necesidades de la red, no solo a encaминamiento sino también a la toma de decisiones de la reconfiguración de la red en aras del intercambio de recursos, como pueda ser capacidad de cómputo, o energía. Además, se contempla el uso de herramientas de AI/ML como elemento auxiliar en este proceso de control y optimización, con el fin de dotar a la red de capacidades de adaptación proactiva, pudiendo contemplar la predicción de eventos o fallos, y la mejora de las decisiones de reconfiguración y balanceo de carga en tiempo real.

Por otro lado, en el segundo bloque de objetivos de la Tesis, se busca analizar en la infraestructura que habilitan las redes softwarizadas, desde un punto de vista de la gestión y en control de la red, así como de la seguridad. En este sentido, se busca profundizar

en el uso de herramientas de despliegue, monitorización y gestión de red, que permitan al administrador de red tener una visión global del estado de la infraestructura, así como poder tomar decisiones automáticas sobre la reconfiguración y optimización de la red. Asimismo, se contempla el estudio de las implicaciones de rendimiento que conlleva el uso de redes programables, donde se busca identificar posibles cuellos de botella y proponer soluciones para mitigarlos. En este sentido, se contempla el uso de técnicas de AI/ML para la reconfiguración de la red, tomando métricas en la comunicación entre los nodos y el controlador. A continuación, en la Figura 1.1 se presenta un diagrama general del marco de la Tesis, donde se puede ver cómo se relacionan las distintas áreas de estudio y aplicación, así como los objetivos que se persiguen en cada una de ellas.

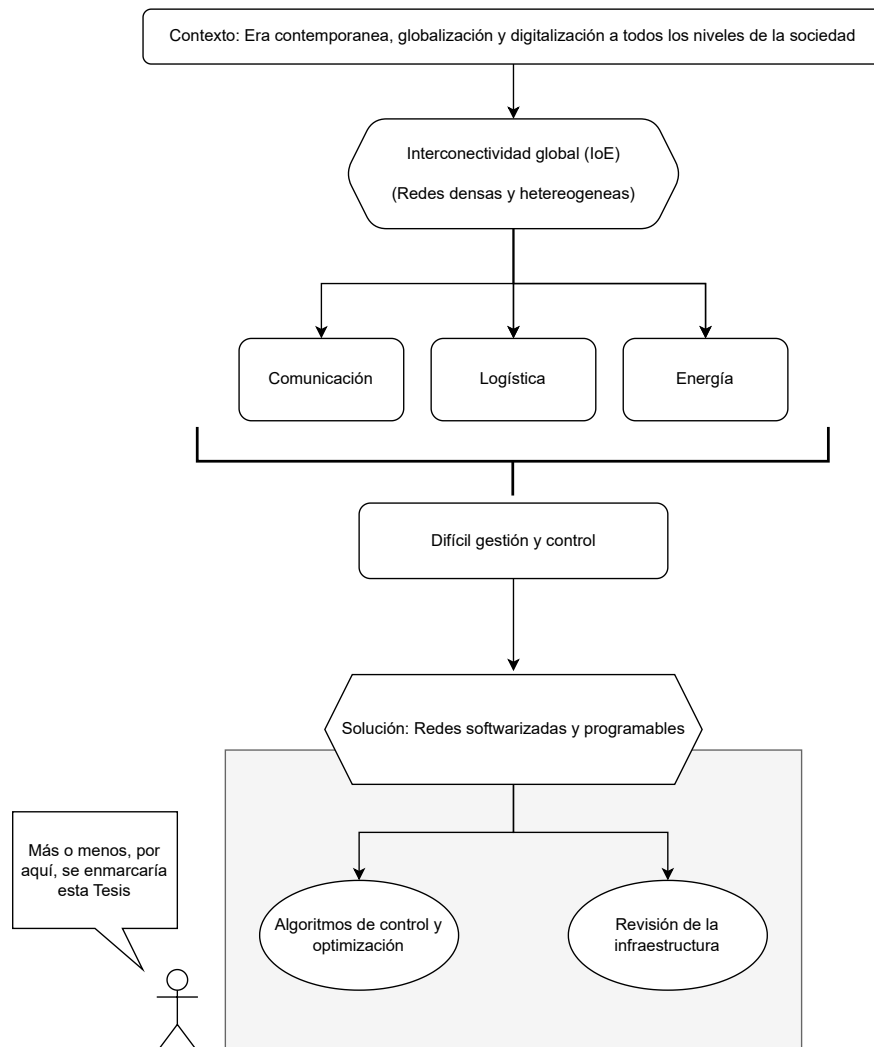


Figura 1.1: Diagrama general del marco general de la Tesis

1.4. Estructura de la tesis

A continuación, se presenta la estructura general de esta memoria, describiendo brevemente el contenido de cada uno de sus capítulos. El objetivo es ofrecer una visión global del desarrollo de la Tesis, que sirva como guía para el lector y facilite la comprensión del marco completo del trabajo realizado.

El primer capítulo ha contextualizado el ámbito de las redes programables y softwareizadas, destacando su relevancia como base tecnológica para la gestión flexible y dinámica de redes heterogéneas. Asimismo, se ha introducido la problemática asociada a la creciente complejidad de las redes actuales, tanto en las redes de comunicaciones, como las de sensores o las redes de distribución energética, y se han formulado los objetivos principales que guían el desarrollo de esta Tesis. Finalmente, el capítulo concluye con una recopilación de las principales contribuciones científicas generadas a lo largo del trabajo.

En el capítulo X,

Por último, el capítulo X recoge las conclusiones principales de la Tesis, así como un bloque que describe futuras líneas de investigación en las que se podrá seguir indagando.

1.5. Contribucciones

El trabajo desarrollado en esta Tesis Doctoral ha generado una contribución notable a la comunidad científica, tanto en términos de generación de conocimiento como en su difusión y transferencia. En concreto, se han publicado cuatro artículos en revistas indexadas en JCR, incluyendo una publicación en una revista de alto impacto Q1 y tres en Q2, y otra dos más que está en revisión (Q2). Además, se han presentado cuatro trabajos en conferencias internacionales organizadas por el IEEE, lo que demuestra la solidez y el interés internacional del trabajo. Como parte del compromiso con la divulgación científica, los avances de esta Tesis también han sido compartidos en eventos como las X Jornadas de Jóvenes Investigadores de la Universidad de Alcalá y la 5th EUGLOH Annual Student Research Conference 2024. Cabe destacar, como elemento diferenciador, el reconocimiento al potencial de transferencia tecnológica de los resultados de esta investigación, materializado en la obtención del Primer Premio en el Concurso de Ideas para la Creación de Empresas de Base Tecnológica de la UAH en 2024. Este premio pone de relieve la capacidad de esta Tesis no solo para generar conocimiento científico de calidad, sino también para transformarlo en soluciones con impacto real en la sociedad, alineadas con los principios de innovación y transferencia del sistema universitario.

Artículos de revista indexadas de alto impacto :

1. Carrascal, D., Rojas, E., Arco, J. M., Lopez-Pajares, D., Alvarez-Horcajo, J., & Carral, J. A. (2023). A comprehensive survey of in-band control in sdn: Challenges and opportunities. *Electronics*, 12(6), 1265. (JCR Q2)

2. Rojas, E., Carrascal, D., Lopez-Pajares, D., Alvarez-Horcajo, J., Carral, J. A., Arco, J. M., & Martinez-Yelmo, I. (2024). A Survey on AI-Empowered Softwarized Industrial IoT Networks. *Electronics*, 13(10), 1979. (JCR Q2)
3. Carrascal, D., Rojas, E., Carral, J. A., Martinez-Yelmo, I., & Alvarez-Horcajo, J. (2024). Topology-aware scalable resource management in multi-hop dense networks. *Heliyon*, 10(18). (JCR Q1)
4. Carrascal, D., Bartolomé, P., Rojas, E., Lopez-Pajares, D., Manso, N., & Diaz-Fuentes, J. (2024). Fault Prediction and Reconfiguration Optimization in Smart Grids: AI-Driven Approach. *Future Internet*, 16(11), 428. (JCR Q2)
5. Carrascal, D., Santos, C., Rojas, E., Arco, J. M., Lopez-Pajares, D. & Rodriguez-Sanchez F. J. (2025). Dynamic Energy Routing Using Tree-Based Topologies with Fast Convergence applied to Meshed Microgrids. *IEEE Access* (under review). (JCR Q2)
6. Carrascal, D., Díaz-Fuentes, J., Manso, N., Lopez-Pajares, D., Rojas, E., Savi, M. & Arco, J. M. (2025). Softwarized Edge Intelligence for Advanced IIoT Ecosystems: A Data-Driven Architecture Across the Cloud/Edge Continuum. *Applied Sciences* (under review). (JCR Q2)

Conferencias internacionales:

1. Carrascal, D., Rojas, E., Lopez-Pajares, D., Manso, N., & Gutierrez, E. (2023, December). A scalable SDN in-band control protocol for IoT networks in 6G environments. In *2023 6th International Conference on Advanced Communication Technologies and Networking (CommNet)* (pp. 1-7). IEEE.
2. Rojas, E., Carrascal, D., Lopez-Pajares, D., Manso, N., & Arco, J. M. (2024, February). Towards ai-enabled cloud continuum for iiot: Challenges and opportunities. In *2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)* (pp. 1-6). IEEE.
3. Comeron, R., Rojas, E., Carrascal, D., Alvarez-Horcajo, J., & Arco, J. M. (2024, October). Multi-hop collaborative edge computing involving constrained IoT devices at the far edge. In *2024 15th International Conference on Network of the Future (NoF)* (pp. 22-24). IEEE.
4. Carrascal, D., Rojas, E., Lopez-Pajares, D., Manso, N., Alvarez-Horcajo, J., & Martinez-Yelmo, I. (2025, March). Softwarized Data-Driven Architecture for Edge Computing IIoT Environments: A Proof of Concept. In *2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN)* (pp. 64-68). IEEE.

Actividades de divulgación:

1. Ponentes a las X Jornadas de Jóvenes Investigadores de la UAH, presentando el trabajo titulado “DEN2NE: origen, presente, ¿y futuro?”.

2. Participación en la 5th EUGLOH Annual Student Research Conference 2024, con el trabajo titulado “Advancements in Enabling Technologies for Programmable and Software-Defined Networks: Paving the Way to 6G”.

Premios:

1. Primer Premio - Concurso de ideas para la creación de empresas de base tecnológica - UAH (Programa propio de investigación y transferencia de la UAH 2024).

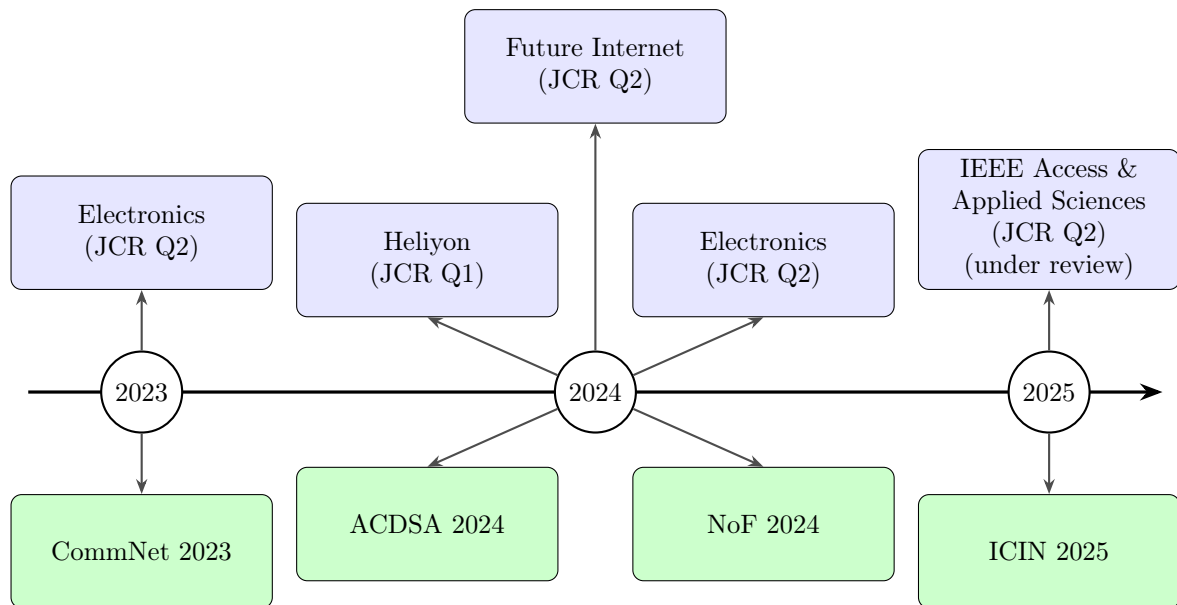


Figura 1.2: Línea de tiempo de contribuciones científicas (2023–2025): publicaciones en revistas y conferencias internacionales.

Capítulo 2

Estado del Arte

Este capítulo tiene como objetivo principal revisar los conceptos clave y el estado del arte que constituyen la base de esta tesis. Para ello, se ha estructurado el capítulo en tres grandes bloques. En primer lugar, se explorarán las redes programables y softwarizadas partiendo de las redes SDN, se seguirá con los algoritmos de red y el uso de la AI como tecnologías habilitadoras, y por último, se analizarán diversos casos de uso relevantes que ejemplifican la aplicación práctica de estas tecnologías, como son las SG y las redes de sensores IIoT.

2.1. Las redes SDN

En este primer bloque se revisan las redes SDN, que son la base de las redes programables y softwarizadas. Se explorarán sus características, ventajas y desventajas, así como sus paradigmas de modos de control, según se indicó anteriormente. Además, se analizarán los protocolos y así como los aspectos clave de las vertientes de trabajo del modo de control *in-band*, y cómo podemos explorar dicho modo para favorecer la flexibilidad y control en redes densas y heterogéneas.

Las redes definidas por software (SDN) representan un nuevo paradigma que rompe con las arquitecturas tradicionales de red. Antes de que apareciera el concepto de SDN, como se puede apreciar en la Figura 2.1, las redes tradicionales solían tener un plano de control unificado en los propios dispositivos, llamado generalmente *Control plane*, en el que se definía la lógica que dictaba cómo se debía llevar a cabo el forwarding de los paquetes, y un plano de datos, conocido como *Data plane*, que se implementaba definiendo su datapath, compuesto por varios bloques de procesamiento para reenviar los paquetes. Ambos planos estarían unificados en un sentido lógico, en un mismo dispositivo. Sin embargo, con la aparición del paradigma de las redes SDN, como se muestra en la Figura, los nodos tradicionales de la red verían cómo su plano de control sería delegado a una entidad externa llamada controlador, preservando su capacidad para manejar los paquetes. En contraste con las arquitecturas tradicionales de la red, donde había que ir configurando equipo a equipo, y donde cada uno de ellos iba a desempeñar una función de red, en las redes SDN, el controlador permite configurar y supervisar de manera inteligente el comportamiento de

la red a través de aplicaciones software, facilitando una programación flexible y dinámica del entorno de red. Por lo que, aunque se sigan llamando “switches” o nodos SDN, estos se comportarán según las reglas que le instale el controlador, pudiendo gestionar paquetes como un switch, un router, un firewall, etc.

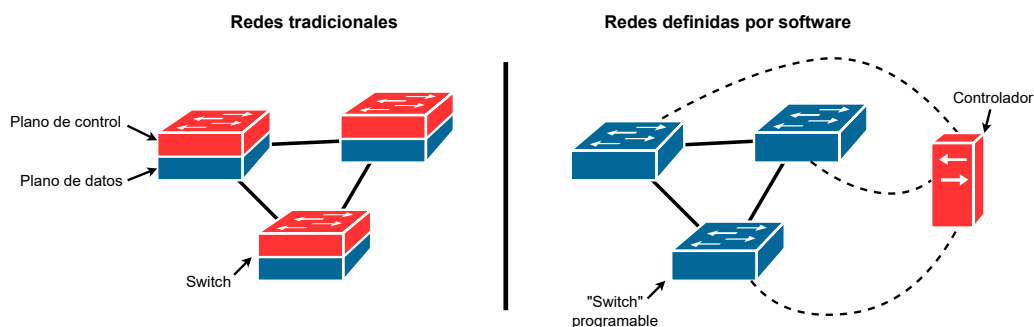


Figura 2.1: Paradigma en las redes SDN

La centralización de la gestión simplifica notablemente las tareas del administrador, al proporcionar una visión global del estado de la red y un punto único desde el cual definir su funcionamiento. A través del controlador, las complejas instrucciones de bajo nivel requeridas por los dispositivos de red tradicionales, como switches y routers, las cuales podían variar en función del fabricante, se abstraen mediante interfaces con sintaxis intuitiva, reduciendo la complejidad operativa. Estas capacidades dotan a la red de una gran agilidad y capacidad de adaptación ante cambios o nuevas necesidades, pudiendo conmutar entre distintos perfiles de funcionamiento de forma automática. El simple despliegue de una nueva aplicación sobre el controlador permite modificar de forma coherente el comportamiento de toda la infraestructura, disminuyendo así los costes asociados al mantenimiento, la operación y el despliegue. Además, SDN promueve activamente el uso de soluciones abiertas tanto a nivel de software como de hardware, fomentando ecosistemas interoperables, reduciendo la dependencia de tecnologías propietarias y eliminando barreras de entrada para nuevos actores en el sector.

2.1.1. Arquitectura lógica de las redes SDN

La arquitectura lógica de las redes SDN se puede dividir en dos planos, el plano de control y el plano de datos, y además, en tres capas: capa de aplicación, capa de control y capa de infraestructura. En la Figura 2.2 se muestra la arquitectura lógica de las redes SDN, así como sus interfaces principales de comunicación que más adelante se explicarán.

El plano de control, se estructura internamente en dos capas funcionales: la capa de control y la capa de aplicación. Estas se comunican mediante la interfaz norte (northbound interface), que permite a las aplicaciones definir políticas de alto nivel que serán interpretadas y gestionadas por el controlador. Estas capas a menudo se pueden encontrar corriendo en la misma máquina, donde conviven el controlador y las aplicaciones que interactúan con él. Sin embargo, también se puede tener un enfoque distribuido, donde el controlador está

en una, máquina, y las aplicaciones en otra, haciendo uso de la interfaz northbound. Por su parte, el plano de datos está conformado por la capa de infraestructura, que engloba los dispositivos físicos de red, principalmente switches SDN, responsables del reenvío de paquetes. La interacción entre el plano de control y el plano de datos se realiza a través de la interfaz sur (southbound interface), cuya función es traducir las decisiones del plano de control en instrucciones ejecutables por los dispositivos de red. En este contexto, el controlador actúa como una pieza clave del sistema, asumiendo responsabilidades esenciales como la instalación de reglas de encaminamiento, la monitorización continua del estado de la red y la recopilación de métricas operativas, las cuales serán aprovechadas por todas las aplicaciones que se ejecuten sobre el controlador.

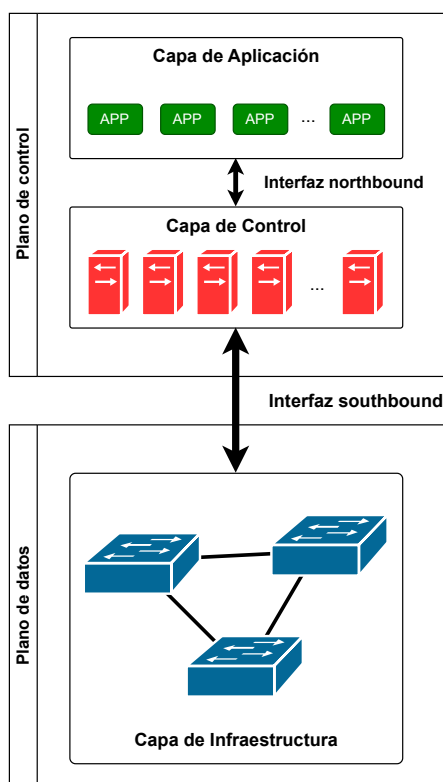


Figura 2.2: Arquitectura lógica de las redes SDN

El plano de datos, por el contrario, no posee lógica de control propia, limitándose a ejecutar las reglas recibidas, como por ejemplo, hacer un reenvío, o descartar paquetes según las reglas establecidas, además de enviar estadísticas de tráfico al controlador. Esta separación de funciones establece una división clara entre la inteligencia de la red, localizada en el plano de control, y su ejecución, delegada como se ha explicado, al plano de datos. De esta manera, se rompe con el modelo tradicional en el que ambos planos

coexistían en un mismo dispositivo de red. Este enfoque modular no solo mejora la escalabilidad y la flexibilidad del sistema, sino que también reduce significativamente los costes de despliegue (CapEx) y operación (OpEx), al concentrar los recursos de cómputo en un nodo centralizado, y simplificar el hardware requerido en los dispositivos de reenvío.

Según se ha visto en la Figura 2.2, la arquitectura SDN se apoya en una estructura jerárquica formada por tres capas principales: aplicación, control e infraestructura. La capa de aplicación representa el nivel de mayor abstracción dentro del ecosistema SDN. Esta capa integra un conjunto de aplicaciones que, apoyándose en los servicios ofrecidos por la capa de control, permiten definir políticas de gestión, QoS, optimizar el rendimiento de la red y adaptarla dinámicamente a diferentes contextos operativos. Un ejemplo típico de uso es la utilización los servicios de descubrimiento topológico proporcionados por la capa de control, que permiten a las aplicaciones calcular rutas óptimas entre dispositivos de red. Estas aplicaciones suelen desarrollarse empleando lenguajes de alto nivel como Python, Go o C++, con el objetivo de facilitar su portabilidad entre plataformas y maximizar la reutilización del código. No obstante, en la práctica, la existencia de APIs y entornos de desarrollo específicos para cada plataforma de control, como ONOS, OpenDaylight, Ryu o el nuevo controlador del ecosistema SDN, TeraflowSDN, introduce ciertos desafíos en la interoperabilidad y portabilidad del software entre distintas implementaciones. En este sentido, uno de los principales retos actuales de SDN sigue siendo la estandarización de interfaces northbound que permitan una integración más fluida y flexible entre aplicaciones y controladores heterogéneos.

Descendiendo, la siguiente capa es la capa de control, la cual constituye el núcleo funcional del paradigma SDN, albergando la inteligencia centralizada de la red. Actúa como intermediario entre las aplicaciones de alto nivel y los dispositivos físicos de la capa de infraestructura, orquestando tareas críticas como el encaminamiento de flujos, la detección y resolución de fallos, la supervisión continua del estado de la red y la gestión de políticas de seguridad y QoS. Su papel como middleware se traduce en la capacidad de transformar políticas abstractas generadas en la capa de aplicación en instrucciones simples y concretas que pueden ser entendidas por los nodos SDN. Esta capacidad de traducir y escalar la lógica de red permite que un único controlador gobierne cientos o miles de switches de forma eficiente, garantizando escalabilidad y consistencia en entornos distribuidos. Por último, la capa de infraestructura, por su parte, está compuesta por los elementos físicos de la red, fundamentalmente nodos SDN, que ejecutan las decisiones tomadas por el plano de control. Estos dispositivos, carentes de lógica propia, cuentan con un agente SDN encargado de comunicarse con el controlador a través de la interfaz sur (southbound interface), como por ejemplo, OpenFlow o P4Runtime. Su funcionalidad se reduce al reenvío y descarte de paquetes o la recolección de estadísticas, lo que permite simplificar su diseño y reducir sus requisitos hardware.

En cuanto a las interfaces, hay dos, como se ha mencionado anteriormente, la interfaz northbound y la interfaz southbound. La interfaz northbound constituye el canal de comunicación entre la capa de control y la capa de aplicación. Su principal función es ofrecer un punto de acceso lógico al administrador de red, permitiéndole supervisar, configurar y

gestionar el comportamiento de la red sin necesidad de interactuar directamente con los mecanismos de bajo nivel que gobiernan los dispositivos físicos. A través de esta interfaz, las aplicaciones pueden programar políticas o solicitudes que serán traducidas por el controlador en instrucciones comprensibles para los elementos de la infraestructura. No obstante, a diferencia de la interfaz southbound, la interfaz northbound carece de una estandarización formal. En consecuencia, la naturaleza y funcionalidad de esta interfaz varían considerablemente en función del controlador SDN empleado, cada uno de los cuales suele ofrecer su propia API con diferentes modelos de datos, protocolos y lenguajes de interacción.

La interfaz southbound constituye el enlace entre la capa de control y la capa de infraestructura dentro de una arquitectura SDN. A diferencia de la interfaz northbound, esta sí cuenta con protocolos estandarizados ampliamente adoptados, que permiten la interoperabilidad entre los controladores y los dispositivos de red. Históricamente, el protocolo más representativo ha sido Openflow [20]. En la implementación del protocolo Openflow según se indica en la Figura 2.3, el concepto central es el de flujo (del inglés, *flow*), entendido como un conjunto de paquetes que cumplen determinadas condiciones definidas por el controlador. Estas condiciones se almacenan en las denominadas tablas de flujo (del inglés, *flow tables*), y suelen hacer referencia a valores específicos de campos en la cabecera del paquete o al puerto de entrada por el que se ha recibido.

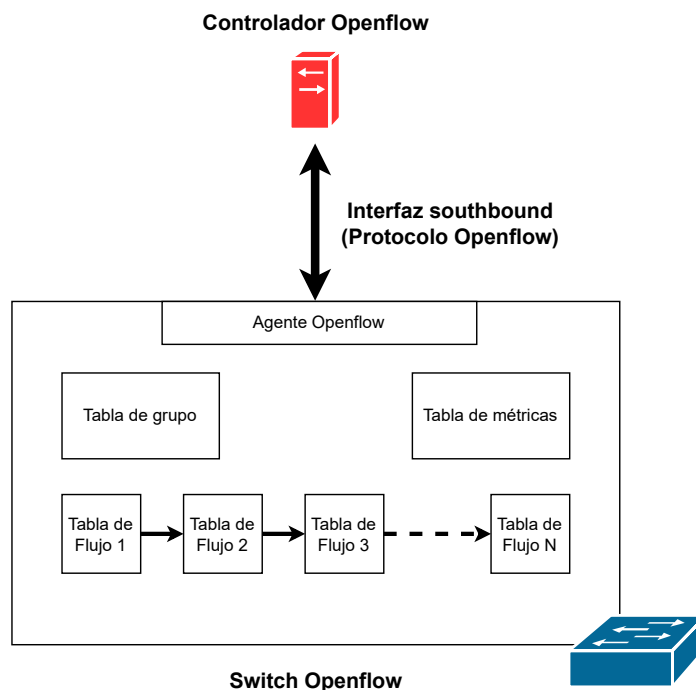


Figura 2.3: Arquitectura básica de switch OpenFlow

Cuando un paquete llega a un switch Openflow, empieza a atravesar de forma iterativa las tablas de flujo y cuando este coincide con los criterios de una regla definida en una tabla, se produce una coincidencia (del inglés, *match*), lo que activa un conjunto de instrucciones asociadas a dicha regla. Estas instrucciones pueden incluir el conteo de paquetes, la aplicación de acciones concretas (como reenviar o descartar el paquete), o bien su reenvío hacia otra tabla para un procesamiento adicional. En caso de no darse una coincidencia, se encapsula y se manda al controlador para que este decida que cómo manejarlo. Así, mediante la instalación de estas reglas por parte del controlador SDN, se determina el comportamiento de reenvío del switch. La comunicación entre el controlador y los dispositivos se realiza a través de un canal estructurado y seguro, que admite mensajes del controlador al switch, mensajes asíncronos generados por los dispositivos, y mensajes simétricos intercambiables por ambas partes, permitiendo una gestión eficiente y dinámica del estado de red.

No obstante, las limitaciones de flexibilidad, extensibilidad y adaptación a nuevas arquitecturas han motivado el surgimiento de alternativas a Openflow. Un ejemplo destacado es el lenguaje Programming Protocol-Independent Packet Processors (P4) [21], diseñado específicamente para superar las restricciones de OpenFlow. Una de las mayores restricciones que tiene OpenFlow es la especificación de forma explícita de los campos de cabecera sobre los que opera. Estos campos de cabecera han pasado de 12 a 41 campos de cabeceras entre sus versiones 1.0 y 1.5 como se puede ver en la Tabla 2.1. Esta evolución ha incrementado la complejidad del protocolo sin proporcionar la flexibilidad necesaria para incorporar nuevas cabeceras o funcionalidades emergentes.

Versión	Fecha	Campos de cabecera
OF 1.0	Dic. 2009	12 campos (Ethernet, TCP/IPv4)
OF 1.1	Feb. 2011	15 campos (MPLS, metadatos entre tablas)
OF 1.2	Dic. 2011	36 campos (ARP, ICMP, IPv6, etc.)
OF 1.3	Jun. 2012	40 campos
OF 1.4	Oct. 2013	41 campos
OF 1.5	Mar. 2015	44 campos

Tabla 2.1: Evolución de versiones del protocolo Openflow y el número de campos de cabecera soportados

En respuesta a ello, P4 nació con tres objetivos principales:

- Permitir la reconfiguración del dispositivo en caliente, es decir, cambiar el comportamiento de los switches una vez desplegados.
- Ofrecer independencia de protocolo, desvinculando el procesamiento de paquetes de protocolos específicos que tengan que estar estandarizados para poder ser gestionados.
- Proporcionar independencia del hardware, permitiendo que las funcionalidades de procesamiento se definan sin depender de los detalles del dispositivo subyacente.

Si bien es cierto que la iniciativa de P4 nació con este objetivo en mente (*open-hardware*), la realidad es que, en la actualidad, se ha visto como cada fabricante ha implementado equipos que si cumplen con algunas de las arquitecturas de P4, pero que cada uno te ofrece unas primitivas de programación diferentes, haciendo que un programa P4 que corre en un dispositivo de un fabricante no sea totalmente compatible en otro [22].

En comparación con Openflow, si nos fijamos en la Figura 2.4, podemos apreciar que empleando P4 se puede definir el cómo el switch va a manejar los paquetes, como los va a procesar y parsear, manteniendo la lógica de las tablas de flujo que teníamos en Openflow, pero ganando en flexibilidad dado que se pueden definir el propio datapath del dispositivo sin depender de un conjunto estandarizado de campos de cabecera. Esto incluso permite que P4 pueda ser implementado en dispositivos de baja capacidad [23], al poder ajustar el datapath a la mínima expresión necesaria para cumplir con las necesidades de la red. Al igual que en Openflow se tenía el protocolo de comunicación para la interfaz southbound, P4 también tiene su propia interfaz de comunicación, llamada P4Runtime [24], que permite a los controladores gestionar y programar dispositivos P4 de forma dinámica.

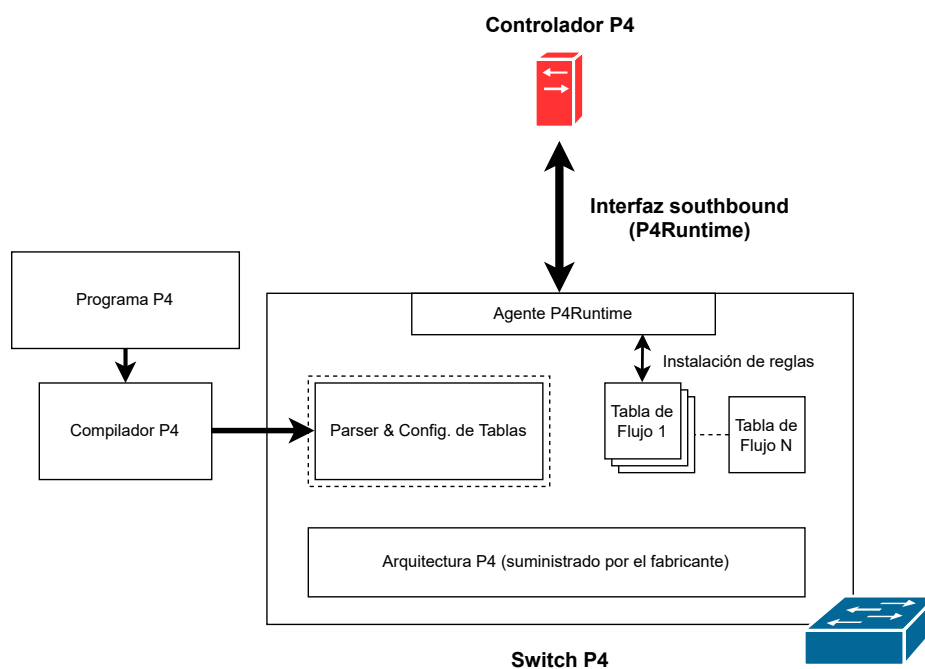


Figura 2.4: Arquitectura básica de switch P4

A diferencia de Openflow, que define un conjunto cerrado de operaciones y estructuras, P4 y su interfaz P4Runtime introducen la posibilidad de reconfigurar dinámicamente el comportamiento del plano de datos mediante descripciones personalizadas del procesamiento de paquetes. Esto se logra mediante una arquitectura basada en google Remote

Procedure Call (gRPC), que ofrece cinco tipos de operaciones principales (Write, Read, Set/GetForwardingPipelineConfig y StreamChannel) para gestionar tanto el estado como la lógica interna de los switches programables. De esta forma, P4 se presenta como una propuesta de evolución de Openflow, orientada a lograr una programabilidad del plano de datos más flexible y escalable, haciéndolo ideal para testing y pruebas de concepto de nuevas soluciones de red.

Paralelamente, ha ido creciendo otra vía complementaria orientada a la gestión y configuración unificada de dispositivos de red llamada OpenConfig. Esta iniciativa, impulsada mayormente por un consorcio de operadores y fabricantes, propone un conjunto de modelos de datos basados en YANG que permiten describir de forma estandarizada y agnóstica el estado operativo y la configuración de dispositivos de red. A diferencia de OpenFlow o P4Runtime, que se centran en el comportamiento del plano de reenvío, OpenConfig aborda la gestión, configuración, el monitoreo y la automatización de tareas de red a través de protocolos como gNMI o NETCONF. Esto convierte a OpenConfig una herramienta clave para aquellas empresas que buscan una gestión softwarizada y programable de sus infraestructuras ya existentes, dado que permite la integración de dispositivos heterogéneos de diferentes fabricantes bajo un modelo común de gestión. Si bien es cierto que OpenConfig no permite definir explícitamente el plano de datos, a diferencia de Openflow o P4, donde se considera que todos los switches o nodos SDN equivalen a un único dispositivo lógico gestionado de forma centralizada, OpenConfig propone un enfoque diferente. Esta iniciativa busca establecer un conjunto común de modelos de datos y configuración, independientes del fabricante, para gestionar redes heterogéneas. A diferencia del enfoque SDN tradicional, en el que los dispositivos se integran como un único plano de control y datos, los dispositivos gestionados mediante OpenConfig siguen operando como entidades independientes. Ambos enfoques persiguen una mayor transparencia y facilidad de gestión de la red, pero difieren en su grado de abstracción y centralización: mientras SDN trata la red como un todo unificado, OpenConfig mantiene la identidad individual de cada dispositivo, facilitando la interoperabilidad en entornos mixtos. Sin embargo, los últimos controladores SDN como TeraflowSDN [25], han comenzado a integrar OpenConfig como una de sus interfaces southbound (además de P4), incluso llegando a no implementar Openflow, lo que sugiere una tendencia hacia un nuevo ecosistema de redes SDN que combina la flexibilidad de la programación del plano de datos con la estandarización y la gestión eficiente de dispositivos heterogéneos.

En este sentido, la evolución de la interfaz southbound no debe entenderse en términos de sustitución de unos protocolos por otros, sino como una diversificación funcional que permite combinar capacidades de reenvío programable, comunicación eficiente y gestión estandarizada según las necesidades específicas de cada red.

2.1.2. Arquitectura física de las redes SDN

Una vez que se ha revisado la arquitectura lógica de las redes SDN, es importante entender cómo se implementa físicamente esta arquitectura, es decir, cómo se conectan los diferentes componentes que se vienen explicando en la sección anterior.

En una red SDN, según se indicó en la Figura 2.2, se compone de un elemento central, el controlador, y un conjunto de switches o nodos SDN distribuidos en la capa de infraestructura los cuales son gestionados por el controlador. Sin embargo, también es posible la implementación de múltiples controladores en una misma red SDN, lo cual aporta funcionalidades adicionales a la red, como mecanismos de respaldo y tolerancia a fallos, que incrementan su fiabilidad, y por ende, la resiliencia de la red. Por ello, se pueden clasificar las conexiones físicas en las redes SDN en dos bloques:

- Las conexiones entre los switches de la capa de infraestructura.
- Las conexiones entre el controlador y los switches de la capa de infraestructura.

Las primeras constituyen la topología física de la red, cuya estructura depende del entorno en el que se despliegue y de los objetivos funcionales de la red. Por ejemplo, en redes SDN diseñadas para centros de datos, es común adoptar una arquitectura jerárquica y regular, ya que esta facilita la escalabilidad y permite absorber incrementos en la demanda de tráfico de forma eficiente [26]. En cambio, en entornos de redes de sensores SDN, es frecuente emplear topologías en malla parcial (tendiendo hacia a una malla completa) [10], que permiten una mayor resiliencia frente a fallos y una reducción en la latencia gracias a la existencia de múltiples caminos entre nodos de la topología.

En cuanto a las conexiones entre el controlador y los switches de la capa de infraestructura, estas se pueden clasificar en principalmente en dos categorías, si bien es cierto que se puede encontrar una tercera categoría que combina ambas. Observando la Figura 2.5, se pueden distinguir dos paradigmas de control: el modo de control *in-band* y el modo de control *out-of-band*, y por último, el modo de control *hybrid-band*, el cual es una combinación de los dos anteriores.

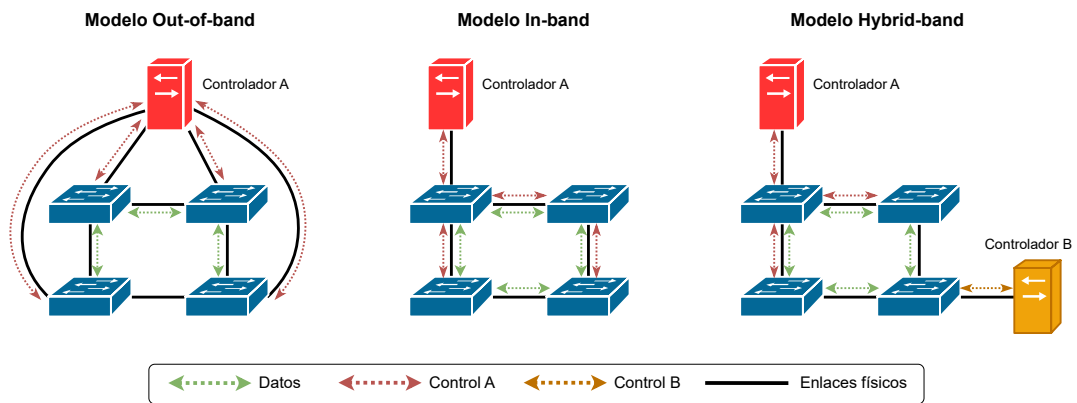


Figura 2.5: Paradigmas de control en las redes SDN

Al desplegar el canal de control en una red SDN, es posible optar por un enfoque out-of-band o in-band, como se ilustra en la Figura 2.5. En el primer caso, denominado

out-of-band, cada nodo SDN dispone de un enlace físico dedicado que lo conecta directamente con el controlador. De este modo, la información de control se transmite a través de una red independiente, exclusiva para dicho propósito, lo cual incrementa la seguridad y el aislamiento del canal, aunque implica un mayor coste de infraestructura al requerir al menos un enlace adicional por nodo. Por el contrario, en el enfoque in-band, solo algunos nodos SDN mantienen un enlace directo con el controlador, mientras que el resto accede a él a través de la propia red de datos, reutilizando los enlaces existentes para transportar la información de control. En este caso, los mensajes de control comparten la infraestructura del plano de datos, lo que puede comprometer su seguridad e integridad, al estar más expuestos a posibles interferencias o interceptaciones.

Finalmente, el enfoque hybrid-band contempla una solución intermedia, en la que coexisten enlaces dedicados y compartidos para la comunicación con el plano de control [27], como se muestra en la Figura 2.5. Este modelo busca equilibrar los costes operativos con los requisitos de fiabilidad y seguridad.

Cada uno de estos esquemas de despliegue presenta ventajas e inconvenientes [27], y la elección entre uno u otro depende fundamentalmente del escenario de red y del caso de uso considerado [28, 29]. No existe un paradigma mejor que otro, sino, que cada enfoque ofrece características particulares que pueden resultar más o menos adecuadas según los requisitos del entorno. Por ejemplo, el modelo out-of-band requiere un enlace físico adicional dedicado a la comunicación entre el controlador y cada nodo SDN, lo que incrementa notablemente los costes de despliegue y mantenimiento. No obstante, esta separación garantiza un mayor aislamiento del canal de control, lo que mejora sustancialmente la seguridad de las comunicaciones. En contraposición, el modelo in-band reutiliza los enlaces existentes del plano de datos para transmitir la información de control, lo que reduce significativamente el coste de infraestructura. Sin embargo, esta economía viene a expensas de una menor seguridad, ya que los mensajes de control comparten canal con el tráfico de red, quedando expuestos a posibles interferencias o ataques.

Además, uno de los principales retos del enfoque in-band radica en la configuración inicial, es decir, el nodo debe conocer de antemano la ruta hacia el controlador a través de la red de datos. En contraste, el modelo out-of-band facilita esta tarea, al disponer de una interfaz exclusiva para dicho propósito. Por ello, en in-band, esta información tiene que proporcionarse mediante protocolos específicos que permiten a cada nodo identificar la interfaz adecuada para reenviar los paquetes de control. Estos protocolos son de especial de interés dado que no existe una solución estandarizada en la academia. Debido a lo cual, se quiere explorar en mayor medida qué opciones existen y qué metodologías se han empleado, dado que estas soluciones son fácilmente extrapolables a otros tipos de redes densas y heterogéneas que empleen entornos softwarizados con una tipología de control in-band. Así, por ejemplo, en entornos como el IoT, donde los dispositivos suelen disponer de una única interfaz de comunicaciones y cuentan con recursos energéticos limitados, el modelo in-band se presenta como una alternativa óptima, al evitar la necesidad de enlaces adicionales que aumentarían el consumo energético y reducirían la vida útil del sensor.

La Tabla 2.2 resume comparativamente las principales características de estos modelos. En ella se observa cómo el paradigma out-of-band destaca por su simplicidad de configuración y seguridad, mientras que el in-band sobresale en términos de escalabilidad y costes.

Propiedad	Control <i>out-of-band</i>	Control <i>in-band</i>
Configuración del dispositivo SDN	Sencilla	Compleja
Seguridad del canal de control	Segura, canal aislado	Riesgosa, canal compartido
Costes de mantenimiento y despliegue	Elevados	Reducidos
Escalabilidad	Limitada	Buena
Resiliencia	Costosa	Recuperación rápida

Tabla 2.2: Características del control *in-band* y *out-of-band*

2.1.3. Propuestas de despliegue con control in-band

La tendencia actual indica que el control in-band está ganando protagonismo en los despliegues de redes SDN [30], especialmente en redes de grandes y densas, donde el coste de utilizar un modelo out-of-band puede resultar prohibitivo. Además, el control in-band habilita el desarrollo de una amplia variedad de nuevas aplicaciones, sobretudo en entornos SDN híbridos o con restricciones de recursos [31, 32], donde el despliegue de enlaces dedicados de control puede ser complejo o incluso inviable. Entre los casos de uso más representativos que se benefician del control in-band se encuentran las redes 5G [33] y las Non-Terrestrial Networks (NTNs) [34], así como diversos escenarios del ámbito IoT, como redes submarinas [35], entornos orientados a la eficiencia energética [36] o sistemas con recursos limitados [37]. A pesar de sus numerosas ventajas, los esfuerzos dirigidos al diseño de protocolos comunes e integrales para el control in-band han sido escasos. Una solución efectiva debería considerar la compatibilidad con plataformas ampliamente utilizadas, tanto en los controladores como en los dispositivos SDN, a fin de garantizar una integración completa en los despliegues actuales. En este contexto, diferentes propuestas han explorado mecanismos para habilitar o mejorar el control in-band, con el objetivo de facilitar su adopción en entornos reales y responder a los retos que plantea, este paradigma. A continuación, se presentan algunos de los trabajos más representativos en esta línea.

En este contexto, los trabajos existentes sobre control in-band pueden clasificarse en función de tres aspectos clave que dicho paradigma debería proporcionar: encaminamiento automático, recuperación rápida ante fallos y arranque autónomo de la red. Además, se ha identificado un cuarto aspecto transversal, relacionado con entornos de control distribuido que también merecen la pena revisar. En primer lugar, la necesidad de contar con un mecanismo de encaminamiento automático en el control in-band resulta evidente. Mientras que el modelo out-of-band suele implementarse mediante enlaces directos entre los nodos SDN y el controlador, el control in-band requiere calcular rutas entre los dispositivos de red y uno o varios controladores, tanto en un sentido ascendente, como descendente. Este mecanismo de encaminamiento es esencial para permitir la comunicación de control a

través del plano de datos y suele condicionarse al tipo de despliegue o a las capacidades del entorno. En segundo lugar, una de las principales ventajas del control in-band es su capacidad de recuperación rápida ante fallos. En caso de que un enlace o switch sufra una avería, es posible restaurar la comunicación de control simplemente seleccionando una ruta alternativa dentro del plano de datos. No obstante, para que este proceso resulte eficaz, los mecanismos de restauración o protección de rutas deben estar bien definidos y coordinados con la lógica de encaminamiento previamente establecida. El tercer aspecto fundamental es el denominado arranque autónomo de la red (del inglés, *network bootstrapping*), que hace referencia a la capacidad del sistema para configurar automáticamente los parámetros necesarios antes del inicio de su operación. Estos parámetros incluyen, entre otros, la información de conectividad entre nodos SDN y controladores. Si bien este proceso también es deseable en entornos out-of-band, en el caso de control in-band se convierte en un requisito crítico, ya que las rutas de control pueden variar en función del despliegue y del protocolo de encaminamiento utilizado. Por último, en entornos con control distribuido, donde existen varios controladores implicados en la toma de decisiones, es necesario incorporar mecanismos de coordinación que permitan compartir rutas, realizar recuperación ante fallos de manera conjunta y gestionar el arranque de la red de forma sincronizada. Además, el control in-band puede ofrecer un canal adicional para la comunicación interna entre controladores, lo cual añade una dimensión interesante a su uso en arquitecturas distribuidas.

En primer lugar, varios autores han sentado las bases de un marco genérico de control in-band y han explorado mecanismos de encaminamiento automático con capacidad de recuperación ante fallos simples de la red. Sharma *et al.* [38] presentan un prototipo basado en OpenFlow que evalúa la viabilidad del encaminamiento automático en distintos switches, proponiendo métodos de restauración y protección frente a fallos individuales. De manera similar, Goltsmann *et al.* [39] introducen el protocolo ICS, que construye un árbol de expansión etiquetado para garantizar conectividad resiliente con un bajo sobre coste computacional. Ambos trabajos ponen de manifiesto la factibilidad del control in-band en escenarios sencillos con encaminamiento automático, pero no extienden sus propuestas a la gestión simultánea de múltiples fallos ni proporcionan un estándar interoperable. Por el contrario, Khakhalin *et al.* [40], formulan un algoritmo de encaminamiento automático resistente a múltiples fallos mediante la asignación de etiquetas únicas a cada switch, sin embargo, la gestión de estas etiquetas se complica rápidamente a medida que el tamaño de la red aumenta. Este mismo problema de escalabilidad se encuentra en la propuesta de Mohan *et al.* [41], donde definen un esquema encaminamiento con rutas de control disjuntas por nodo para detectar y aislar switches maliciosos; si bien mejora la seguridad, su formulación matemática no escala bien debido al elevado número de variables de decisión. Otro grupo de autores, Raza *et al.* [42] y González *et al.* [43], investigan el uso de protocolo MultiPath TCP (MPTCP) para implementar el encaminamiento automático y aumentar la tolerancia a fallos con rutas de respaldo. No obstante, Raza *et al.* no llegaron a implementar su solución en un entorno real, y González *et al.* requieren un canal out-of-band para la fase inicial de arranque, lo que limita la autonomía del despliegue in-band. Finalmente, existen propuestas más avanzadas y específicas de encaminamiento automático para entornos heterogéneos. Asadujjaman *et al.* [44] proponen un enfoque reactivo capaz

de recuperar múltiples fallos, aunque su comunicación controlador-switch es *source-routed*, con sobrecarga que penaliza la escalabilidad. López-Pajares *et al.* [45], proponen Amaru, una solución en la cual integran exploración de red y múltiples rutas de protección con bajo coste y tiempos de recuperación reducidos, pero requieren pequeñas modificaciones en los switches para ser plenamente funcionales. Görkemli *et al.* [46] plantean un plano de control dinámico capaz de redistribuir carga entre varios controladores y adaptar el enrutamiento in-band según la topología y las necesidades de las aplicaciones; sin embargo, su evaluación solo contempla despliegues virtualizados. Holzmann *et al.* [47] presentan Izzy, un protocolo distribuido basado en árboles de expansión y direcciones temporales que logra tiempos de recuperación inferiores a 100 ms en simulaciones en redes de área amplia (del inglés, Wide Area Network (WAN)), aunque aún no dispone de validación en entornos reales. Por último, Kumazoe *et al.* [48] diseñan un canal in-band en entornos P4, embebiendo mensajes de control en paquetes de usuario; aunque innovador, su evaluación inicial revela degradaciones en el reenvío de datos que quedan pendientes de resolver.

Después de analizar los trabajos sobre encaminamiento automático y recuperación antes fallos simples de la red para control in-band, se ha visto que comparten una serie de supuestos y bloques funcionales recurrentes. El primero de ellos, es que parten de la premisa de que debe garantizarse la alcanzabilidad del controlador desde cada switch, por lo que incluyen algún mecanismo, ya sea centralizado o distribuido, para calcular rutas de control de forma automática; asumen la existencia de capacidades en el plano de datos (tablas de reenvío, contadores, soporte OpenFlow/P4 o uso de MPTCP) que permiten instalar reglas y supervisar el estado del tráfico; incorporan estrategias de resiliencia (protección o restauración) basadas en rutas de respaldo; emplean técnicas de identificación o etiquetado (*labels*, *locators*, identificadores temporales) para encaminar mensajes de control sin colisiones o bucles y finalmente, la mayoría valida su propuesta mediante simulación, emulación o prototipos parciales en entornos controlados. Estas piezas comunes configuran el esqueleto funcional de las propuestas y explican por qué los retos reiterados (escalabilidad, gestión de fallos múltiples, interoperabilidad y evaluación en entornos reales) aparecen de forma transversal en la literatura.

En segundo lugar, se han identificado varios trabajos que proponen protocolos concretos para el arranque autónomo (*network bootstrapping*) de la conexión de control in-band. Sharma *et al.* [49] fueron pioneros en implementar una configuración de control in-band haciendo uso del canal Openflow, que dependía en gran parte de un arranque autónomo de la red, el cual, combina el protocolo Dynamic Host Configuration Protocol (DHCP) para obtener direcciones IP y ARP para resolver direcciones MAC, mientras el controlador descubre la topología mediante mensajes sondas de tipo Link Layer Discovery Protocol (LLDP). Su prototipo, basado en el controlador NOX, demostró tiempos de arranque reducidos en topologías variadas, pero su enfoque depende de servicios clásicos de red y no aborda aspectos de interoperabilidad entre controladores heterogéneos. En la misma línea, Tu *et al.* [50] integran un control SDN in-band en switches convencionales de un centro de datos (del inglés, Data Center (DC)) y usan servidores de directorio para orquestar el arranque autónomo y el encaminamiento de la red, lo que funciona bien en entornos controlados pero introduce dependencias en componentes adicionales que no siempre están

disponibles en despliegues heterogéneos. Algunas propuestas requieren protocolos o canales externos para completar el arranque autónomo. Su *et al.* [51] utilizan Open vSwitch Database (OVSDb) para configurar el plano de datos y facilitar la gestión de conexiones in-band en Open vSwitch (OVS), lo que permite cierto control bien definido pero acopla la solución a capacidades propietarias de OVS. Sakic *et al.* [52] plantean dos mecanismos: uno basado en pre-configuración secuencial y otro apoyado en Rapid Spanning Tree Protocol (RSTP) para construir árboles de expansión de mínimos costes; ambas opciones funcionan pero resultan lentas (convergencia en segundos) y una depende del comportamiento de protocolos tradicionales de capa 2. Wu *et al.* [53] también emplean RSTP (rXstp) para estabilizar la topología previa al establecimiento de la conexión de control, requisito que obliga a intervenciones manuales (prioridades de switches) y limita el uso a escenarios con un único controlador. El tiempo de arranque y la sobrecarga de mensajes son factores clave, en este caso, Sharma *et al.* muestra buenos tiempos en ensayos, mientras que Suo *et al.* [27] propone reglas pre-instaladas para acelerar el arranque autónomo de la red en un DC pero se observa pérdida de rendimiento y cierto coste en *throughput*. Silva-Freitas *et al.* [54] abordan explícitamente la eficiencia de la red: su protocolo reduce el número de mensajes frente a LLDP, y además incorpora autenticación, mejorando seguridad y eficiencia. Li *et al.* [55] proponen un mecanismo de arranque muy rápido (50 switches en ≈ 2 s) construyendo un árbol en una pasada, aunque deja la recuperación rápida como trabajo futuro. En contraste, Sakic *et al.* obtienen convergencias lentas en sus opciones, lo que evidencia una tensión entre rapidez, coste y robustez.

En conjunto, la literatura sobre *network bootstrapping* para control SDN in-band ofrece un conjunto rico de alternativas y prototipos, pero existen huecos recurrentes: (i) soluciones agnósticas y estandarizables que no dependan de protocolos o extensiones concretas; (ii) procedimientos escalables y de baja latencia para redes grandes; (iii) arranques seguros y autenticados adecuados para entornos adversos; (iv) mecanismos que combinen arranque, recuperación ante fallos (incluyendo fallos de nodo) y soporte multi-controlador; y (v) validación en entornos reales o testbeds industriales. Estos vacíos marcan líneas claras para trabajos futuros y contextualizan por qué muchas propuestas actuales se centran en escenarios acotados o exigen compromisos prácticos entre coste, seguridad y complejidad.

Por último, en lo referente al control distribuido, varios trabajos abordan cómo establecer conectividad in-band tanto entre switches y controladores como entre controladores entre sí. Schiff *et al.* [56] presentan Medieval, que crea y mantiene dos árboles de expansión por controlador, uno restringido a la región del controlador y otro de alcance global, para conseguir tanto la conquista de switches como la interconexión entre controladores. La propuesta incluye un prototipo en emulación pero requiere IPs de controlador preconfiguradas y reglas iniciales en los switches; no detalla por completo la gestión de fallos de controlador y depende de una determinada configuración previa. En trabajos posteriores, los mismos autores refinan la idea con una aproximación *plug & play* que permite añadir y quitar controladores dinámicamente [57], aunque nuevamente la solución se apoya en reglas preinstaladas y en ARP para descubrimiento, por lo que queda espacio para reducir las dependencias iniciales y hacer el arranque completamente autónomo. La coordinación segura entre controladores es clave para evitar inconsistencias en el plano de control. Schiff

et al. proponen un marco de sincronización basado en operaciones atómicas implementadas exclusivamente con OpenFlow (aprovechando la característica de los *bundles*) [58]. El enfoque demuestra que es posible diseñar primitivas de sincronización sin modificar switches, pero su validez práctica requiere controladores y switches que soporten las primitivas de OpenFlow necesarias (p. ej. *bundles* en versiones ≥ 1.4) y su evaluación se limita a entornos emulados. Otro bloque importante de trabajo en entornos de control distribuido se centra en garantizar recuperación y convergencia tras fallos arbitrarios en el plano de control, para asegurar la consistencia de control a lo largo de la red. Canini *et al.* [59, 60], presentan Renaissance, donde introducen mecanismos que aseguran que, tras fallos arbitrarios, cada controlador no-fallido pueda alcanzar switches en tiempo acotado y que cada switch sea gestionado por al menos un controlador no-fallido. Renaissance ofrece pruebas formales de auto-estabilización y evaluaciones extensas en emulación. Estos avances resuelven muchos problemas teóricos, pero la mayoría de las pruebas siguen siendo emuladas y falta validación en testbeds a escala real. La gestión dinámica del balance de carga de control y la activación/desactivación de controladores es abordada por Görkemli *et al.* [61]. Su arquitectura permite activar controladores bajo elevada carga y reconfigurar rutas de control según la demanda; fue validada en Mininet con OVS y Floodlight, mostrando mejora frente a enfoques estáticos. Sin embargo, la evaluación se limita a entornos virtualizados y falta demostrar su resiliencia y costo real en *hardware* y escenarios con tráfico heterogéneo. Algunos trabajos plantean canales livianos para comunicación entre controladores. Hark *et al.* [62] proponen un servicio de inter-controladores de bajo coste que emplea mensajes embebidos en eventos de cambio de puerto y Virtual Local Area Networks (VLANs) para aislamiento; muestran crecimiento lineal de mensajes con el número de controladores. Canini *et al.* [63] proponen procedimientos basados en *timeout* frente a enfoques iterativos para que switches obtengan rutas hacia todos los controladores y para coordinar controladores entre sí. Ambos contribuyen a la coordinación, pero no resuelven plenamente la escalabilidad de la señalización ni la gestión de conflictos en topologías grandes o federadas. La detección y localización de fallos son tratadas por Kwan-Yee *et al.* [64], que describen un esquema de sondeo cíclico que identifica rápidamente fallos y localiza su origen; su evaluación en red real con ONOS y switches comerciales exhibe tiempos de recuperación por debajo de 50 ms para fallos simples. Este trabajo muestra que la detección rápida es viable en hardware, aunque el coste en sondeos periódicos y su escalado a redes muy grandes requieren una caracterización más profunda. Holzmann *et al.* [47] presentan primero Izzy (mencionado anteriormente), un esquema robusto basado en árboles y *locators*, y posteriormente Seedling [65], que agrupa controladores por proximidad para reducir coste computacional y tablas de forwarding. Estas propuestas exploran *trade-offs* entre robustez y coste: Seedling reduce la carga frente a repetir Izzy en cada controlador, pero urge analizar pérdida de generalidad y límites de agrupamiento en topologías heterogéneas.

En general, la mayoría de las propuestas se han validado en simuladores o entornos emulados (Mininet, Java, OVS), aunque hay excepciones con pruebas en hardware real (p. ej. Kwan-Yee *et al.* con ONOS y HP5900). Si bien el código de Renaissance y otras implementaciones están disponibles, la transición a testbeds industriales y despliegues a escala sigue siendo limitada. La literatura sobre control distribuido para SDN ha avanzado en

marcos conceptuales (Medieval, Renaissance), mecanismos de sincronización y algoritmos para robustez y eficiencia (Izzy, Seedling), así como en soluciones pragmáticas para detección y coordinación (Hark *et al.*, Kwan-Yee *et al.*). No obstante, persisten huecos claros: reducir las dependencias de configuración inicial, escalar la coordinación entre numerosos controladores, trasladar garantías formales a entornos reales y optimizar coste/beneficio en detección y señalización. Estas lagunas dibujan líneas de trabajo futuro necesarias para llevar los resultados de laboratorio a despliegues industriales y operativos.

2.1.4. Conclusiones y alineación con los objetivos de la Tesis

El análisis crítico de la literatura sobre control *in-band*, el encaminamiento automático, arranque de red y control distribuido, ha permitido identificar una serie de huecos recurrentes que condicionan la adopción práctica de estos enfoques en entornos reales. Entre los hallazgos más relevantes destacan: la ausencia de un protocolo *in-band* estandarizado y agnóstico al fabricante/proveedor; la necesidad de mecanismos de arranque autónomo (*bootstrapping*) escalables, seguros y con baja latencia; la complejidad de coordinar controladores distribuidos sin incurrir en *overhead* excesivo; y la escasa validación en infraestructuras reales o testbeds a gran escala.

Esto encaja de forma directa con el planteamiento y los objetivos de la Tesis. El primer bloque de objetivos (profundizar en los mecanismos de control de redes programables y extender el paradigma SDN a ámbitos como IIoT y redes de distribución eléctrica) queda plenamente justificado por los *gaps* detectados: en escenarios con nodos con recursos limitados (IIoT) o topologías jerárquicas (redes eléctricas), las soluciones *out-of-band* no son viables y se requiere avanzar en protocolos *in-band* que sean seguros, autónomos y adaptativos. Además, la necesidad de algoritmos capaces de decidir dinámicamente (encaminamiento, reconfiguración, intercambio de recursos como cómputo o energía) refuerza la pertinencia de investigar técnicas que integren procedimientos clásicos con herramientas de AI/ML para predicción de fallos y toma de decisiones proactivas.

El segundo bloque de objetivos (diseño de infraestructuras software para gestión, orquestación y seguridad en redes softwarizadas) también responde a los *gaps* encontrados. La literatura muestra propuestas puntuales de arquitectura y prototipos que no siempre consideran la integración con sistemas de monitorización, gestión en el arranque, o la orquestación autónoma en entornos heterogéneos. Por tanto, resulta necesario definir arquitecturas modulares y alineadas con estándares que faciliten la interoperabilidad, permitan la incorporación de capacidades de AI/ML. A la luz de lo anterior, esta Tesis propone una hoja de ruta clara y coherente con los objetivos planteados. Desde la perspectiva metodológica, la Tesis busca cerrar la brecha entre modelado teórico y aplicabilidad práctica: no se trata solo de proponer algoritmos con buenas propiedades matemáticas, sino de validar su viabilidad operativa y sus implicaciones clave. En particular, la incorporación de escenarios aplicados (IIoT, SG) permitirá evaluar restricciones reales y topológicas que condicionan tanto la elección del paradigma de control, cómo su diseño.

2.2. Servicios y Tecnologías habilitantes en redes softwarezadas y heterogéneas

En este segundo bloque se quieren revisar las principales tecnologías habilitantes que permiten la creación, control y gestión de las redes programables y heterogéneas, así como identificar dónde y de que manera se integran con el diseño del paradigma SDN. Estas tecnologías son fundamentales para entender el marco de trabajo de la tesis, y cómo, posteriormente, se pueden llegar a aplicar a diferentes casos de uso.

Una vez revisado el paradigma SDN, se tiene que ahondar en el diseño arquitectónico promovido por la ONF, el cual, evolucionó desde la separación inicial entre plano de control y plano de datos hacia una visión más amplia centrada en servicios: el denominado Management–Control Continuum (MCC) [66]. En esta nueva visión, funcionalidades que implementa el controlador SDN se identifican como servicios, y los datos y los nodos a gestionar por el controlador, como recursos, según se describe en la Figura 2.6.

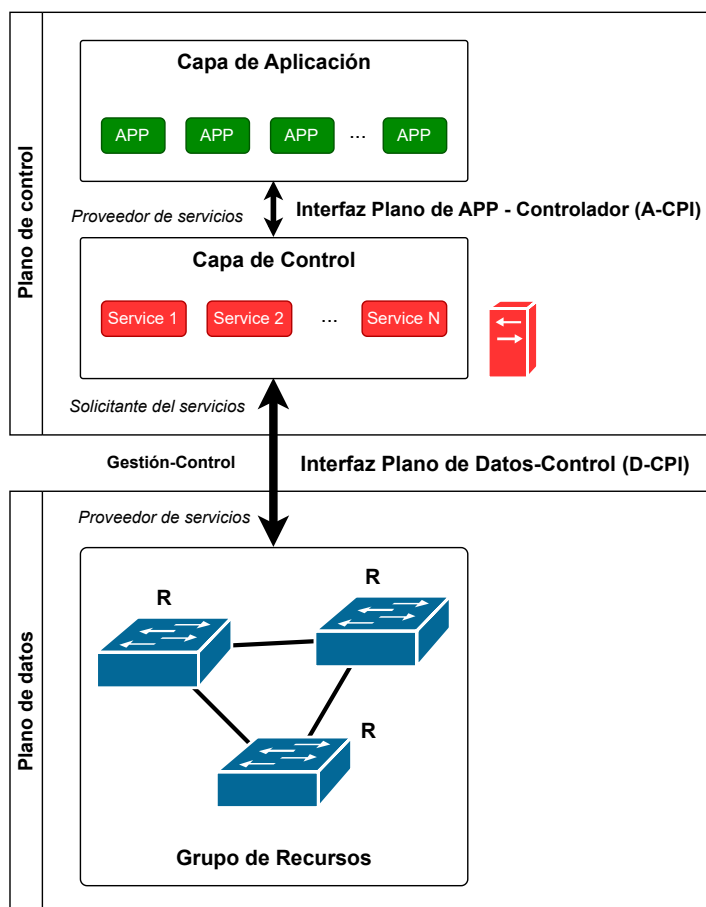


Figura 2.6: Paradigmas de control *Management–Control Continuum* (MCC)

Según la ONF, una arquitectura es una colección necesariamente incompleta de perspectivas sobre un conjunto de ideas subyacentes, por ello, partiendo del modelo básico de SDN, trataron de identificar todas las funcionalidades que implementaba el controlador como servicios que gestionan una serie de recursos, en este caso los nodos de la red, llevando a la mínima expresión la arquitectura de una manera agnóstica, para que esta fuera extrapolable a cualquier tipo de redes programables y heterogéneas. Por ende, un consumidor de servicio, será aquel que intercambia tanto datos como operaciones de control y gestión con un servidor o proveedor SDN (D-CPI). Aunque los datos del usuario son finalmente transmitidos o procesados por un conjunto de recursos (R) que pertenecen al proveedor SDN, el cliente final controla su servicio mediante una sesión de control y gestión, invocando acciones (A-CPI) sobre un conjunto de recursos virtuales alojados en el controlador que percibe como propios. El controlador SDN, entre otras funciones, se encarga de virtualizar y orquestar la vista de recursos y servicios que tiene el cliente, mapeándola sobre los recursos y servicios reales del proveedor. Los conceptos de “recursos” y “servicios” son intencionalmente flexibles y no tienen límites fijos. La arquitectura SDN amplía este modelo básico, al denominado como MCC, y aclara sus implicaciones, incluyendo extensiones clave como el uso compartido de recursos entre múltiples clientes, la asignación dinámica y la optimización de su uso.

El MCC organiza las capacidades de la red en una jerarquía de servicios que va desde los servicios básicos de gestión y control (descubrimiento, provisión, configuración, monitorización) hasta servicios avanzados de orquestación y automatización que permiten funcionalidades de alto valor sobre infraestructuras softwarizadas. Esta visión facilita describir la red como una plataforma de servicios en la que la infraestructura subyacente se abstrae y se pone a disposición de funciones gestionadas y componibles. Bajo el paraguas del MCC, resulta útil clasificar el estado del arte en dos bloques complementarios que recogen las tecnologías habilitantes principales que se exploran en esta Tesis: (i) servicios básicos que habilitan la operación mínima de la softwarización (descubrimiento de topología, arranque autónomo, provisión de canales de control, telemetría) y (ii) servicios avanzados que aportan capacidades de decisión, predicción y optimización (orquestación, planeamiento de recursos, predicción de fallos y adaptación proactiva), normalmente apoyados en técnicas de inteligencia artificial y aprendizaje automático. En las siguientes sub-Secciones, se explorarán los contenidos principales de esta Tesis englobados sobre los dos bloques de servicios anteriormente mencionados en el contexto del paradigma de control MCC.

2.2.1. Servicios básicos: arranque y provisión de canales de control en entornos densos y heterogéneos

Los servicios básicos de arranque y provisión de canales de control constituyen la base operativa del MCC. Sin un mecanismo fiable para que los dispositivos descubran la topología, obtengan parámetros iniciales y establezcan conectividad con el controlador, es imposible desplegar servicios de mayor nivel. En entornos densos y heterogéneos (IIoT, micro-redes eléctricas, redes logísticas), las restricciones de recursos, la topologías multi-salto y la posible ausencia de enlaces dedicados obligan a diseñar protocolos *in-band* o

híbridos que sean escalables, seguros y poco intrusivos, a la par que resilientes.

En la Sección 2.1.3 se pudo explorar todas las propuestas de mecanismos in-band, entre las cuales se pudo observar diferentes estrategias de arranque/provisión que emplea:

1. **Protocolos clásicos asistidos (DHCP/ARP/LLDP):** usan servicios L2/L3 existentes para obtener parámetros y descubrir topología (p. ej. [49]).
2. **Árboles raíz-centrados y exploración por difusión:** el nodo raíz (con enlace al controlador) propaga paquetes de exploración que recogen topología y permiten instalar rutas in-band [45]).
3. **Etiquetado jerárquico:** asignación de identificadores temporales o jerárquicos a nodos para encaminar mensajes de control sin conocer la topología completa (p. ej. la propuesta de Izzy [47]).
4. **Híbridos con canales out-of-band para arranque:** soluciones que requieren un enlace auxiliar o reglas preinstaladas para la fase inicial y luego migran a in-band (véase [43, 51]).
5. **Mecanismos basados en multipath/TCP avanzado:** uso de MPTCP para mejorar disponibilidad del canal de control mediante rutas múltiples (p. ej. [42]).

Sin embargo, gran parte de las soluciones revisadas fueron concebidas para redes de comunicaciones convencionales y, como consecuencia, dependen de protocolos auxiliares o de implementaciones específicas en equipos SDN. Ese fuerte acoplamiento limita su aplicabilidad a entornos densos y heterogéneos, por ejemplo, redes con nodos de recursos muy reducidos (IIoT) o redes de distribución eléctrica con topologías en árbol, donde no es viable confiar en servicios adicionales ni en modificaciones de hardware/software. En contraste, los enfoques más abstractos, basados en esquemas de etiquetado jerárquico y en la construcción de árboles enraizados combinados con mecanismos de difusión/control de exploración, resultan más extrapolables y prácticos para estos escenarios: reducen dependencias infraestructurales, facilitan la escalabilidad y favorecen la resiliencia mediante rutas alternativas y reconvergencia local, por lo que constituyen las candidatas más adecuadas para los objetivos de esta Tesis.

Los esquemas basados en etiquetado jerárquico combinados con la construcción de árboles enraizados constituyen una estrategia natural y eficiente para el arranque y la provisión de canales de control en redes densas y heterogéneas. En estos enfoques, la red se organiza lógica o temporalmente en niveles mediante identificadores (etiquetas) que codifican información de pertenencia o proximidad, de forma análoga a prefijos jerárquicos; dichas etiquetas permiten encaminar mensajes de control con mínimo conocimiento global, reduciendo la necesidad de tablas extensas en los nodos. Estas etiquetas, pueden identificar a nivel de nodo, a nivel de enlace o a nivel de puerto, y dependerá del caso de uso que tipo de identificador emplear. La construcción de un árbol enraizado con la raíz colocada en el nodo, o nodos, con acceso directo al controlador facilita el arranque de la red, dado que, paquetes de exploración pueden propagarse de forma controlada desde la raíz acumulando

información topológica y asignando rutas in-band sin inundar la red. Esta combinación aporta ventajas prácticas relevantes para entornos con recursos limitados: localiza las decisiones de reencaminamiento, ofrece caminos de protección natural (rutas alternativas hacia la raíz dentro del árbol o a través de etiquetas vecinas) y limita el *overhead* de señalización. No obstante, conlleva retos propios, como por ejemplo, la gestión dinámica de etiquetas ante cambios topológicos (re-etiquetado) puede generar coste de señalización, y es preciso diseñar mecanismos para evitar bucles y ataques por suplantación (p. ej. autenticación ligera de mensajes de exploración). Es por ello, que el etiquetado jerárquico más árboles enraizados proporciona un marco abstracto, escalable y resistente, bien adaptado a IIoT, micro-redes y topologías jerárquicas en árbol típicas en redes de distribución eléctrica, siempre que se complementen con políticas de convergencia eficientes y escalables adecuadas para el caso de uso final.

Para analizar con mayor detalle los enfoques basados en etiquetado jerárquico y la construcción de árboles enraizados, esta sección introduce los conceptos básicos de teoría de grafos necesarios para modelar la topología de una red y para explicar cómo, mediante esquemas de etiquetado, es posible derivar una estructura de árbol enraizado adecuada para el arranque y el encaminamiento de control. A continuación se formalizan los términos y notaciones que se emplearán en el resto del capítulo; posteriormente se estudian diferentes propuestas de etiquetado jerárquico y su aplicabilidad a las redes densas y heterogéneas.

2.2.1.1. Terminología básica

La teoría de grafos es la disciplina matemática que estudia las propiedades y estructuras de los grafos. Un grafo es una representación visual compuesta por una serie de objetos llamados nodos, conectados a través de otro tipo de objetos denominados enlaces. Históricamente, los orígenes de la teoría de grafos se remontan al problema de los puentes de Königsberg, también conocido como problema de los siete puentes de Königsberg. Su nombre se debe al nombre de la ciudad, Königsberg, ciudad de Prusia Oriental, y el problema consistía en encontrar un camino para cruzar a pie toda la ciudad pasando solo una vez por cada uno de los siete puentes y regresando al mismo punto de partida. El problema, formulado originalmente de manera informal, se propagó a modo de juego matemático entre los intelectuales de la época, siendo resuelto por Leonhard Euler en 1736 [67] y cuya resolución dio origen y sentó las bases de la teoría de grafos.

Formalmente, un grafo se define como $G = (\mathcal{N}, \mathcal{L})$, siendo \mathcal{N} un conjunto de N nodos, y $\mathcal{L} \subseteq \{\{i, j\} \mid i, j \in \mathcal{N} \text{ con } i \neq j\}$, un conjunto de enlaces los cuales se componen de un par desordenado de nodos, es decir, cada enlace tiene dos nodos diferentes. Estos enlaces pueden tener una direccionalidad, o no. Atendiendo a la direccionalidad del conjunto de enlaces, los grafos se pueden clasificar en grafos dirigidos o no dirigidos, siendo los grafos dirigidos los que restringen el tránsito por los enlaces a una o varias direcciones específicas, mientras que los grafos no dirigidos no establecen ninguna obligatoriedad en la dirección de tránsito. Asimismo, se puede establecer un coste asociado a atravesar un enlace, pudiendo clasificar los grafos también en grafos ponderados, donde los enlaces tienen un coste asociado, y no ponderados, donde los enlaces no cuentan con un coste asociado.

Otra característica interesante para clasificar los grafos es si tienen ciclos (bucles) o no. Un grafo se considera acíclico si no contiene ningún ciclo, esto significa que para cada nodo $i \in \mathcal{N}$, no hay un camino directo que empiece y termine en i .

Si, además, es conexo y no dirigido, es decir, existe un camino entre cualquier par de nodos, y acíclico, se denomina árbol. En el caso dirigido, cuando no existen ciclos dirigidos se habla de un Directed Acyclic Graph (DAG). La presencia o ausencia de ciclos es una propiedad estructural clave que condiciona los algoritmos de encaminamiento, los procesos de exploración/*bootstrapping* y los mecanismos de convergencia (por ejemplo, la prevención de bucles o la estrategia de etiquetado). Por ello, su consideración resulta esencial en el diseño de esquemas de etiquetado jerárquico y en la derivación de árboles enraizados a partir de la topología original G .

Por tanto, la teoría de grafos se puede aplicar de forma directa sobre las redes densas y heterogéneas, modelando dichas redes partiendo de una topología física a un grafo, el cual será hiperconectivo. Sobre la topología física habrá que definir cual será el nodo, o los nodos, que darán acceso al controlador, los cuales se identificarán como nodos raíz. Dichos nodos serán los encargados de iniciar el establecimiento del proceso de etiquetado jerárquico para poder construir el árbol enraizado. Este árbol, según se ha indicado, es un tipo especial de grafo acíclico conectado, el cual, también puede considerarse como un grafo con n nodos y $n - 1$ enlaces. Esto se puede analizar como la construcción de una topología lógica sobre una topología física, la cual contendrá todos los nodos de la topología anterior, pero un subconjunto de los enlaces de la topología física. Formulando este proceso del establecimiento de la topología lógica, podemos definir que dado un grafo $G = (\mathcal{N}, \mathcal{L})$, siendo \mathcal{N} un conjunto de N nodos y $\mathcal{L} \subseteq \{\{i, j\} \mid i, j \in \mathcal{N} \text{ con } i \neq j\}$ un conjunto de enlaces los cuales se componen de un par desordenado de nodos (es decir, cada enlace tiene dos nodos distintos), se puede obtener la topología lógica como un subgrafo, sea $G' = (\mathcal{N}', \mathcal{L}')$ donde se que cumple $\mathcal{N}' \subseteq \mathcal{N}$ y $\mathcal{L}' \subseteq \mathcal{L}$. En este caso, de forma adicional, se cumple la relación de igualdad para los nodos, dado que $\mathcal{N}' \subseteq \mathcal{N} \wedge \mathcal{N} \subseteq \mathcal{N}'$, haciendo que $\mathcal{N}' = \mathcal{N}$. Es decir, $a_i \in \mathcal{N} / b_i \in \mathcal{N}'$, $0 \leq i \leq N$ se cumple $\forall a_i = b_i$.

Para ilustrar este proceso de establecimiento de la topología lógica, se presenta la Figura 2.7. Como se puede ver en la Figura 2.7 (a), se parte de un grafo inicial, el cual representará a pequeña escala una red densa y heterogénea. Sobre este grafo se pueden seleccionar distintos subgrafos que cumplan la relación de igualdad según se ha explicado anteriormente. Pudiendo elegir entre distintas topologías lógicas (b,c,d) a conveniencia, y función de que nodo se establezca como nodo raíz de la topología. En el caso de que el nodo raíz de la topología fuera el nodo 1, podemos seleccionar la topología lógica (b) que habilita al nodo 5 de una conexión más directa con el nodo que da acceso al controlador. Otra opción sería elegir la configuración (c), en caso de que los nodos 5 y 6 no tuvieran una necesidad explícita QoS en la comunicación con el controlador de la red. De igual forma podemos elegir otra configuración, como por ejemplo (d), donde el nodo 4 se encontrara al final de la topología lineal conformada. La elección concreta de la topología lógica, y por tanto la forma del árbol enraizado, depende de criterios de diseño del caso de uso. Por ejemplo, minimizar la profundidad del árbol (para reducir latencias hacia la

raíz), priorizar enlaces con mayor capacidad o menor pérdida (criterio QoS), favorecer rutas disjuntas para aumentar la resiliencia, o minimizar el overhead de señalización.

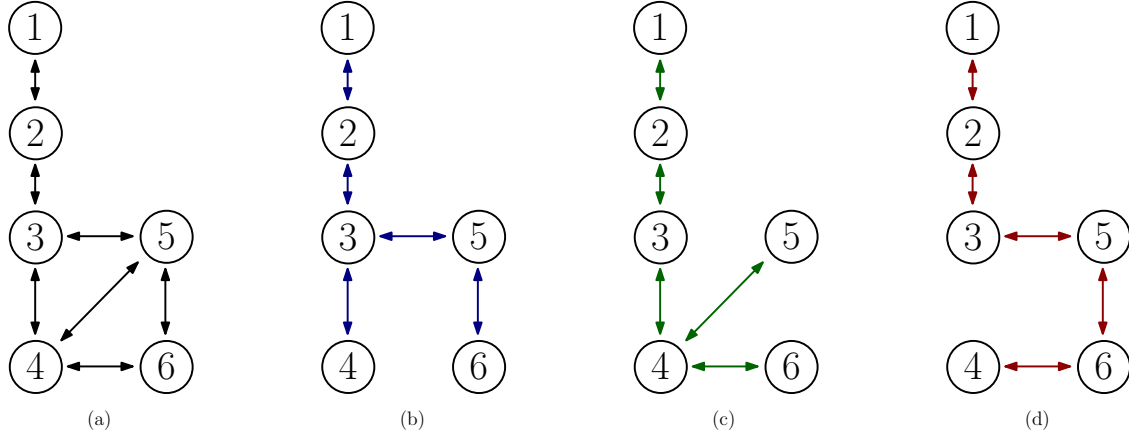


Figura 2.7: Elección de una topología lógica (árbol enraizado) partiendo una topología física

El etiquetado jerárquico es una técnica eficiente y abstracta para obtener una topología lógica, habitualmente un árbol enraizado, a partir de una topología física $G = (\mathcal{N}, \mathcal{L})$. En este paradigma, a cada nodo $i \in \mathcal{N}$ se le asigna una etiqueta que codifica su posición relativa dentro de una jerarquía o partición de la red. Estas etiquetas permiten a los nodos tomar decisiones de reenvío locales, por ejemplo, mediante coincidencia de prefijos, sin requerir una visión global completa de G . El etiquetado jerárquico resulta especialmente adecuado para entornos densos y heterogéneos porque reduce la información de estado requerida en nodos con recursos limitados, facilita la derivación de rutas hacia la raíz o raíces usando información local y habilita mecanismos de agregación que simplifican la gestión. Existen diversas familias de esquemas de etiquetado jerárquico. En el etiquetado por prefijos, cada etiqueta se expresa como una secuencia jerárquica $i_1.i_2.\dots.i_k$. En el etiquetado por intervalos, a cada subárbol o partición se le asigna un intervalo de identificadores, de modo que un nodo conoce su pertenencia comprobando si su identificador está dentro del intervalo correspondiente. Otros enfoques utilizan identificadores compactos que codifican información relativa a la distancia o la topología respecto a la raíz, y todavía existen esquemas basados en clusterización, en los que la red se divide en grupos y cada grupo recibe una etiqueta de nivel superior con subetiquetas internas. La elección entre estas variantes suele depender de limitaciones de memoria, de la necesidad de agregar información (p. ej. capacidades o requisitos QoS) y de la facilidad de reconfiguración en caso de cambios topológicos.

Operativamente, la creación de la topología lógica mediante etiquetado se articula en varias fases. Primero se seleccionan las raíces, es decir, los nodos con acceso directo al controlador o controladores, que actuarán como iniciadores del proceso. A continuación, desde la raíz se emiten mensajes de exploración controlada que se propagan por la red física; al recibir un mensaje por primera vez, un nodo acepta la etiqueta propuesta (por ejemplo, concatenando un identificador del enlace, el puerto, su identificador o un sufijo

jerárquico) y establece una relación “vecindad” con el emisor. Con la etiqueta asignada, cada nodo obtiene reglas locales de reenvío que permitan encaminar paquetes de control hacia su padre y, por ende, hacia la raíz. De forma complementaria, es posible manejar varias etiquetas por nodo que identifiquen rutas alternativas, facilitando así la reconvergencia rápida ante fallos de enlace o de nodo. En entornos prácticos, estas fases deben diseñarse para evitar inundaciones de mensajes y para que la aceptación de etiquetas siga políticas antibucles.

Para ilustrar este proceso se presenta la Figura 2.8. En este caso se puede apreciar un grafo de cinco nodos, entre los cuales se ha elegido al nodo 1 como nodo raíz de la topología. Será este nodo quien empiece el proceso de difusión del etiquetado jerárquico. En este escenario se ha optado por un tipo de etiquetado basado en identificadores únicos los cuales representan de forma inequívoca al nodo dentro de la topología física. En el primer paso de la difusión, ver Figura 2.8 (a), el nodo raíz empieza a difundir su etiqueta con identificador único 1, la cual llega hasta el nodo 2, quien la recibe y acto seguido concatena su identificador único generando la etiqueta 1.2 que será almacenada y difundida por todos los enlaces disponibles menos por el que le llegó (siendo un aspecto de diseño en la estrategia de etiquetado). De forma análoga, el nodo 3 generará la etiqueta 1.2.3, almacenándola y difundiéndola a los nodos 4 y 5, obteniendo estos a su vez las etiquetas 1.2.3.4 y 1.2.3.5, respectivamente. En el segundo paso de la difusión, ver Figura 2.8 (b), los nodos de la topología ya tienen un camino directo para alcanzar el nodo raíz, el nodo 2 de forma directa, el nodo 3 a través del nodo 2, y los nodos 4 y 5 a través del nodo 3. Sin embargo, el proceso de difusión no ha concluido, dado que los nodos 4 y 5 aún tienen una etiqueta por difundir, la cual se intercambian entre ellos. En el tercer paso de la difusión, ver Figura 2.8 (c), los nodos 4 y 5 obtienen las etiquetas 1.2.3.5.4 y 1.2.3.4.5 respectivamente, lo que les habilita a una ruta adicional hacia el nodo raíz (del nodo 4 a través del nodo 5, y viceversa), pudiendo elegir entre un camino u otro.

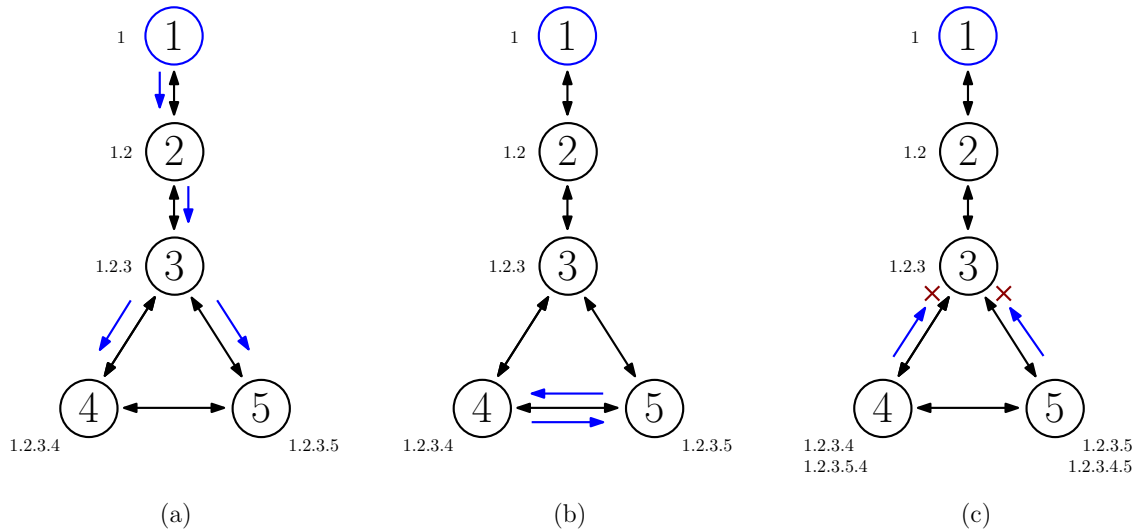


Figura 2.8: Ejemplo de difusión empleando el etiquetado jerárquico

En el último paso (Figura 2.8 (c)), se puede apreciar otro aspecto clave de los esquemas de etiquetado: los mecanismos antibucle. Los nodos 4 y 5 reciben respectivamente las etiquetas 1.2.3.5.4 y 1.2.3.4.5, la cuales tendrán que ser difundidas de forma análoga hacia el nodo 3. Al llegar dichas etiquetas al nodo 3, y al utilizar identificadores inequívocos, este tiene la capacidad de comprobar si las etiquetas recibidas contienen su propio identificador, y en tal caso descartarlas, dado que en propio significado de la etiqueta le estaría indicando al nodo que en esa ruta tendría que volver sobre el en una ocasión, reflejando un bucle (ciclo). Una vez descartadas las etiquetas, y con todas las etiquetas difundidas en la topología física, todos los nodos tendrían una o más rutas para alcanzar al nodo raíz y conocerían su posición jerárquica en la topología. De esta forma, aplicando diferentes criterios de los anteriormente mencionados, se podrían obtener tres topologías lógicas (tres árboles) a partir de una topología física: el primero donde los nodos 4 y 5 cuelgan directamente del nodo 3, el segundo donde el nodo 4 se conecta directamente al nodo 3 y el nodo 5 se conecta al nodo 4, y el tercero, donde el nodo 5 se conecta directamente al nodo 3 y el nodo 4 se conecta al nodo 5. Este proceso de etiquetado y selección de la topología lógica puede ser ejecutando de forma periódica, o configurable, para actualizar el encaminamiento automático, explorar de nuevo la topología en escenarios de movilidad o modificaciones de la topología física, siendo esto crítico en redes densas y heterogéneas.

El diseño de esquemas de etiquetado implica una serie de *trade-offs* relevantes. Etiquetas más compactas reducen la memoria y la carga de procesamiento en los nodos, pero limitan la información codificable (por ejemplo, la posibilidad de transportar metadatos de capacidad o prioridad). Los esquemas por prefijo e intervalos son naturalmente agregables y escalables, pero pueden requerir re-etiquetado amplio cuando la topología cambia; por ello, las soluciones prácticas suelen implementar re-etiquetado localizado, versionado de etiquetas o etiquetas temporales para mitigar la señalización. La prevención de bucles durante la fase de construcción exige políticas en los mensajes de exploración según se ha indicado.

Es por ello, que el etiquetado jerárquico encaja especialmente bien con los requisitos de redes IIoT, micro-redes y sistemas de distribución eléctrica y redes de comunicaciones, dado permite realizar *bootstrapping* sin depender de servicios L2/L3 adicionales ni de extensiones propietarias, limita el estado necesario en cada nodo, escala bien con el tamaño de la topología física, y facilita la introducción explícita de rutas de respaldo. Además, este enfoque se complementa de forma natural con técnicas de AI/ML, modelos predictivos pueden priorizar la exploración de determinados subárboles, seleccionar enlaces con menor probabilidad de fallo o recomendar políticas de re-etiquetado que optimicen la resiliencia y el rendimiento de la topología lógica. En la siguiente sección, se analizarán algunas de las propuestas de etiquetado jerárquico, se compararán sus prestaciones y bondades que serán de utilidad en los entornos estudiados en esta Tesis.

2.2.1.2. Propuestas de etiquetado jerárquico

En el contexto de los servicios básicos, en concreto, al arranque y provisión de canales de control en entornos densos y heterogéneos, el etiquetado jerárquico aparece como una

solución consolidada dentro del catálogo de funcionalidades que el MCC clasifica como servicios de gestión y control fundamentales. Aunque el etiquetado jerárquico no constituye una novedad conceptual [68], ha sido aplicado con éxito en dominios tan diversos como enrutamiento por prefijos [69], direccionamiento en redes de sensores y esquemas de localización para IoT [70], su carácter transversal y su capacidad de abstracción lo convierten en un mecanismo idóneo para materializar las capacidades mínimas que exige el MCC en escenarios con recursos limitados y topologías heterogéneas. Por otra parte, muchas de las propuestas y variaciones prácticas que se examinan a continuación proceden de nuestro grupo de investigación, NetIS, que lleva años investigando y prototipando esquemas de etiquetado y árboles enraizados; estas aportaciones servirán de referencia a la hora de comparar enfoques, identificar *gaps* y situar las contribuciones de esta Tesis en el marco de las redes densas y heterogéneas. A continuación se presenta una taxonomía de enfoques y se analizan propuestas representativas, prestando especial atención a su aplicabilidad práctica, requisitos de implementación y limitaciones en entornos densos, distribuidos y heterogéneos.

En primer lugar, los trabajos teóricos clásicos proporcionan la base formal del problema de asignación de identificadores en redes anónimas. Uno de los primeros trabajos identificados en materia de etiquetado jerárquico lo presentan Fraigniaud *et al.* [68], donde estudian la asignación de etiquetas únicas en redes anónimas sin conocimiento previo de la topología ni de su tamaño, empleando un modelo síncrono con una fuente distinguida, un nodo raíz, que inicia el proceso. Sus resultados son valiosos porque demuestran límites y *trade-offs* fundamentales entre longitud de etiqueta, tiempo y complejidad de mensajes; sin embargo, las fuertes hipótesis (p. ej. sincronía estricta) y la ausencia de tratamiento de topologías dinámicas o con fallos hacen que su aplicabilidad directa a entornos reales y heterogéneos sea limitada. Este tipo de trabajo marca el terreno teórico, pero deja abiertos problemas prácticos como la asincronía y las restricciones de recursos en redes de baja capacidad.

En el ámbito aplicado, una familia de propuestas orientadas a los DC explora mecanismos de etiquetado jerárquico para encaminamiento sin tablas y reenvío rápido. Rojas *et al.* [69], presentan Torii-HLMAC, donde introducen direcciones posicionales, Hierarchical Local MAC (HLMAC), y un mecanismo distribuido de asignación automática que habilita encaminamiento sin tablas en *bridges* y recuperación instantánea ante fallos. Más adelante, Rojas *et al.* [71, 72], evolucionan esta idea, presentando Generalized Automatic Address Assignment (GA3) y eTorii. GA3 propone un descubrimiento distribuido que genera múltiples direcciones HLMAC ordenadas y soporta encaminamiento sin tablas de reenvío sobre árboles de rutas mínimas; eTorii combina *source routing* con asignación automática para permitir encaminamiento *on-the-fly*. Estas propuestas comparten ventajas claras para topologías jerárquicas (baja ocupación de estado, conmutación rápida, balanceo por selección de etiquetas), pero su evaluación se ha centrado mayoritariamente en topologías de DC (fat-trees, topologías jerárquicas) y bajo condiciones relativamente homogéneas. Por ello, se pueden identificar una serie de debilidades evidentes, como por ejemplo, la extrapolación a topologías irregulares o muy heterogéneas, el comportamiento bajo enlaces de muy distinta capacidad/latencia, y el coste de implementación en dispo-

sitivos con capacidades limitadas requieren más estudio experimental y, en algunos casos, soporte en hardware (p. ej. P4) para validar su uso en producción.

En paralelo aparecen soluciones probabilísticas y adaptativas pensadas para redes dinámicas. El trabajo publicado por Walraed *et al.* [73], presentan la solución ALIAS bajo el protocolo *Decider/Chooser Protocol* (DCP), donde proponen algoritmos aleatorizados para la selección de etiquetas en redes con cambios frecuentes; su fortaleza es la adaptabilidad y las buenas garantías probabilísticas de convergencia, pero su evaluación se centra en escenarios de centros de datos y deja abierto cómo se comporta en entornos con recursos muy limitados o con tasas extremas de *churn* (tasa de abandono y unión masiva de nodos). Estos enfoques aleatorizados aportan resiliencia frente a dinamismo, pero su sobrecoste de mensajes y su idoneidad para nodos de baja energía deben cuantificarse para IIoT.

Respecto a redes de baja capacidad con restricciones en consumo, y en almacenamiento de información de estado, Rojas *et al.* [70], presentan IoTorii, donde se adapta el paradigma jerárquico a las Low-Power and Lossy Networks (LLNs). IoTorii se basa también en las HLMAC, reutilizando direcciones MAC típicas transformándolas en etiquetas jerárquicas y emplea sondas *broadcast* para descubrir rutas múltiples sin modificar la pila de protocolos utilizada en las LLNs. Los resultados experimentales muestran reducción de entradas en tablas de reenvío, y un menor *overhead* respecto a Low-Power and Lossy Networks (RPL) (protocolo de facto en las LLNs) en escenarios estáticos, lo que evidencia la promesa del etiquetado en entornos de baja potencia. No obstante, quedan sin resolver la robustez ante movilidad elevada, la variabilidad extrema de enlaces y la necesidad de políticas energéticas más agresivas para redes de sensores reales, densas y heterogéneas.

En cuanto a las soluciones enfocadas directamente a las redes típicas SDN, tenemos que volver a mencionar Lopez-Pajares *et al.* [45], y Holzmann *et al.* [47], vistos en la Sección 2.1.3, pero haciendo ahora hincapié en su enfoque de etiquetado jerárquico. Lopez-Pajares *et al.* [45], presentaban Amaru, donde apuestan por la exploración controlada desde un nodo raíz para recopilar topología e instalar rutas in-band. Amaru combina exploración y construcción de rutas con respaldo múltiples, y demuestra tiempos de recuperación muy bajos con *overhead* casi nulo. El etiquetado que emplean se basa en los identificadores de los puertos de los nodos SDN que gestionan, haciendo que su principal limitación práctica es que requiere ligeras modificaciones en los switches SDN para soportar el protocolo, lo que plantea barreras de despliegue en entornos heterogéneos donde no es posible modificar el plano de datos de todos los equipos. Este es un debilidad operativa importante para entornos industriales o de infraestructura crítica donde la interoperabilidad con equipos *legacy* es necesaria. De forma similar, Holzmann *et al.* [47], proponen Izzy, basado en un un diseño modular, que combina un árbol de expansión con identificadores temporales dependientes de la topología para asegurar conectividad in-band con tiempos de recuperación por debajo de 100 ms en simulaciones de WAN. Los identificadores temporales les ayudan a detectar cuando uno de los nodos establecidos en la ruta in-band deja de estar operativo, lo que les permite actualizar dichas rutas y reparar el canal de comunicación en un tiempo adecuado. Izzy aporta una arquitectura bien estructurada para

mantener tablas y rutas de respaldo, y demuestra que la combinación etiquetado junto al árbol de expansión puede alcanzar latencias de conmutación muy bajas. Sin embargo, su validación está limitada a simulación y queda por demostrar su factibilidad en despliegues heterogéneos reales; además, la generación y gestión de identificadores temporales plantea cuestiones de coordinación entre controladores en escenarios multi-root, es decir, donde hay más de un nodo con acceso al controlador SDN.

Recapitulando y comparando, todos los trabajos explotan la idea común de codificar información topológica en etiquetas jerárquicas para reducir estado local y facilitar el encaminamiento hacia la raíz, pero difieren en decisiones de diseño clave: (i) si la asignación es determinista (prefijos ordenados, intervalos) o probabilística; (ii) si el encaminamiento es sin tabla de reenvío (Torii/GA3) o requiere reglas instaladas (Amaru/Izzy); (iii) si la solución requiere o no modificaciones en el dataplane; y (iv) el alcance de la evaluación (simulación vs. prototipo vs. despliegue sobre hardware). Esos ejes determinan la aplicabilidad a entornos densos y heterogéneos: por ejemplo, las soluciones sin tabla de reenvío son atractivas para switches con escaso Ternary Content-Addressable Memory (TCAM), pero su robustez en enlaces heterogéneos no ha sido ampliamente probada; las soluciones que requieren cambios en switches presentan una barrera de adopción en infraestructuras mixtas; las propuestas probabilísticas toleran *churn* pero su coste de señalización puede ser alto para sensores con baterías limitadas.

2.2.1.3. Conclusiones y alineación con los objetivos de la Tesis

El análisis del estado del arte ha puesto de manifiesto que el etiquetado jerárquico y las técnicas basadas en árboles enraizados constituyen un marco fértil y consolidado para abordar el arranque y la provisión de canales de control en entornos densos y heterogéneos. Las propuestas revisadas (trabajos teóricos sobre asignación de identificadores, protocolos para centros de datos como Torii/GA3/eTorii, soluciones adaptadas a LLNs como IoTorii, protocolos in-band como Amaru e implementaciones modulares como Izzy) comparten la idea de explotar información codificada localmente en etiquetas para reducir estado y facilitar encaminamiento hacia la raíz o raíces. Sin embargo, de la comparación sistemática emergen limitaciones claras en relación con el objetivo de aplicar estas técnicas fuera del dominio clásico de las redes de comunicaciones. Muchas soluciones se han validado únicamente en topologías jerárquicas homogéneas o en simulación, varias requieren modificaciones del *dataplane* o soporte hardware específico, y pocas consideran nativamente la heterogeneidad de enlaces, la restricción de recursos de nodos IIoT o la coordinación multi-raíz entre controladores distribuidos.

Esto encaja de forma directa con el planteamiento y los dos objetivos de la Tesis: (i) el estudio y diseño de algoritmos y mecanismos de control (etiquetado, encaminamiento, reconfiguración proactiva) aplicables a SDN, IIoT y redes de distribución eléctrica; y (ii) el análisis de la infraestructura habilitadora (herramientas de despliegue y monitorización), incorporando técnicas de AI/ML como elemento auxiliar para predicción y toma de decisiones autónoma. En concreto, cada gap identificado se traduce en uno o varios posibles trabajos dentro de la Tesis: diseño de esquemas de etiquetado tolerantes a he-

terogeneidad, desarrollo de una capa de compatibilidad dataplane-agnóstica, protocolos de bootstrap ligeros, algoritmos de coordinación multi-raíz, y estudios que reduzcan la señalización en nodos con recursos limitados mediante decisiones locales haciendo uso del etiquetado jerárquico (posiblemente asistidas por técnicas de AI/ML). Estas líneas se integran de manera coherente con los objetivos generales de la Tesis y marcan un camino claro desde la revisión del estado del arte hacia contribuciones concretas, evaluables y orientadas a la transferencia tecnológica en contextos reales (IIoT, smart grids y otros entornos densos y heterogéneos).

2.2.2. Servicios avanzados: gestión y planeamiento de recursos en entornos densos y heterogéneos

En esta subsección se revisan las capacidades avanzadas que, apoyadas en el MCC y en las capas de orquestación, habilitan la gestión inteligible y el planeamiento de recursos en redes softwarizadas densas y heterogéneas. Estas capacidades van más allá de las funciones básicas de descubrimiento y provisión: incluyen la orquestación multi-capa (cloud/edge), la planificación de recursos con restricciones (computación, energía, latencia), mecanismos de autoscalado y balanceo, y técnicas de apoyo basadas en AI/ML para la predicción y la toma de decisiones proactiva. A continuación se propone una estructura detallada para abordar estos temas.

2.2.3. Servicios avanzados: optimización y reconfiguración proactiva del control en entornos densos y heterogéneos

2.3. Casos de uso

En este último bloque se revisan los casos de uso más relevantes que se pueden encontrar en la literatura. Estos casos de uso son ejemplos prácticos de cómo las tecnologías habilitantes y las redes programables y softwarizadas se aplican en contextos reales, como las SG y las redes de sensores IIoT.

Capítulo 3

Capítulo de prueba

Este capítulo tiene como propósito verificar la correcta configuración de los índices e instrumentos del documento.

3.1. Uso de acrónimos y referencias

En este documento se utiliza el concepto de SDN, el cual es fundamental en redes programables. Para más detalles, véase el capítulo de introducción.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus finibus, diam non blandit interdum, eros urna accumsan lectus, in aliquet libero neque in tortor. Quisque condimentum posuere ex, ac bibendum dolor posuere ac. Integer dolor metus, sollicitudin a consequat id, eleifend in magna. Pellentesque sed magna hendrerit, aliquet diam non, mollis massa. Phasellus tempus efficitur eros, quis malesuada felis pulvinar ut. Cras feugiat eget velit ut ultricies. Vivamus a tincidunt ante. Fusce neque tellus, molestie quis augue id, pulvinar maximus sapien. Sed blandit ex quis felis sodales, quis semper libero ultrices. Aenean eros lacus, bibendum at ex auctor, accumsan sagittis mauris.

Ut gravida mauris in velit maximus pellentesque. Praesent ultricies, mi eu convallis laoreet, lorem leo euismod urna, non imperdiet ligula purus vestibulum felis. Nam consequat lorem eget leo dictum, vel viverra enim commodo. Maecenas ac porttitor velit, nec dignissim velit. In tellus massa, ornare id est finibus, mattis tristique velit. Sed vitae interdum lectus. Sed placerat quam sit amet lacus lacinia, nec tristique libero efficitur. Ut varius lobortis velit. Mauris euismod dictum luctus. Aliquam neque quam, vehicula quis mollis eget, sollicitudin id elit. Integer cursus risus ac purus fringilla, facilisis condimentum dui sagittis. Pellentesque gravida turpis dui, nec consequat lacus scelerisque et.

Sed mollis, purus at malesuada mattis, dui sem scelerisque lectus, ut fermentum leo urna eget diam. Duis facilisis turpis nibh, in commodo nisi pretium nec. Donec finibus elit et felis elementum finibus. Donec turpis purus, rutrum sit amet orci eu, tincidunt porta quam. Etiam velit mauris, varius in risus at, sodales consectetur metus. Ut urna turpis, ornare id lectus vitae, ultrices cursus urna. Nunc lacinia ullamcorper nunc in facilisis. Fusce rhoncus eros elit, at posuere ex vulputate vel. Phasellus ullamcorper neque eu ante porttitor, vel iaculis justo lobortis. Vestibulum ornare eros ex, eget porta felis ultrices vitae. Donec mauris arcu, vulputate vel lectus vitae, semper tempus ligula.

Maecenas eros dolor, auctor tincidunt enim in, bibendum ultricies magna. Aenean pellentesque interdum condimentum. Cras sollicitudin vel lorem vitae lacinia. Phasellus pulvinar suscipit volutpat. Sed ac vulputate erat, vel luctus ligula. Curabitur pretium mollis ornare. In sit amet nisl quis eros efficitur mollis. Quisque iaculis nisl sed tincidunt condimentum.

Suspendisse finibus, nunc a ultricies tempor, ex neque scelerisque diam, in varius ligula lectus ut nulla. Suspendisse dapibus mi vitae tellus consequat fermentum. Integer bibendum nisl quam, dignissim egestas augue efficitur vitae. Integer pellentesque felis nisl, id iaculis orci maximus sed. Sed blandit pretium leo, volutpat varius quam vestibulum sed. Curabitur sit amet volutpat eros. Nulla vitae tristique tellus. Duis efficitur nec libero placerat pharetra. Cras luctus neque a lorem mattis, eget laoreet magna pharetra. Nam ipsum ligula, hendrerit a orci vitae, ultricies auctor felis. Vestibulum at elit non tellus dapibus lobortis. Donec quis consectetur nibh, sit amet commodo massa. Sed ultrices, velit sed lacinia congue, ipsum mauris suscipit libero, vel posuere est ipsum eu justo. Aliquam ut venenatis neque, ut volutpat nunc. Suspendisse finibus ornare dolor sit amet tristique. Fusce a porta mauris.

Phasellus egestas augue id purus vestibulum, vel posuere ipsum dictum. Fusce eget bibendum dui. Maecenas tempus, sapien at vulputate pretium, elit felis dignissim velit, sit amet feugiat tellus quam vel turpis. Aenean quis viverra nulla, vel rhoncus augue. Morbi sit amet elit fermentum, finibus justo tristique, ultricies sapien. Sed sit amet lacinia nunc, ac pretium mauris. Donec dapibus velit non nunc sagittis semper. Ut ut nulla volutpat, rhoncus libero at, venenatis lorem. Maecenas iaculis dictum arcu eu accumsan. Donec non metus justo. Praesent non ipsum a nisl venenatis efficitur. Etiam lacinia, elit non tristique tincidunt, sapien erat malesuada diam, sed accumsan augue est eget ante. Maecenas at odio accumsan, luctus sem id, lacinia quam.

Sed at nisi erat. Integer scelerisque erat vitae aliquam sagittis. Duis id ipsum auctor orci dapibus porta vel quis augue. Vivamus consectetur dapibus turpis. Vivamus id facilisis nisi, quis fermentum urna. Ut fringilla auctor faucibus. Ut eu sapien nisi. Cras lorem risus, finibus sed scelerisque vel, commodo pulvinar lacus. In at luctus ante, non commodo quam. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus finibus, diam non blandit interdum, eros urna accumsan lectus, in aliquet libero neque in tortor. Quisque condimentum posuere ex, ac bibendum dolor posuere ac. Integer dolor metus, sollicitudin a consequat id, eleifend in magna. Pellentesque sed magna hendrerit, aliquet diam non, mollis massa. Phasellus tempus efficitur eros, quis malesuada felis pulvinar ut. Cras feugiat eget velit ut ultricies. Vivamus a tincidunt ante. Fusce neque tellus, molestie quis augue id, pulvinar maximus sapien. Sed blandit ex quis felis sodales, quis semper libero ultrices. Aenean eros lacus, bibendum at ex auctor, accumsan sagittis mauris.

Ut gravida mauris in velit maximus pellentesque. Praesent ultricies, mi eu convallis laoreet, lorem leo euismod urna, non imperdiet ligula purus vestibulum felis. Nam consequat lorem eget leo dictum, vel viverra enim commodo. Maecenas ac porttitor velit, nec dignissim velit. In tellus massa, ornare id est finibus, mattis tristique velit. Sed vitae interdum lectus. Sed placerat quam sit amet lacus lacinia, nec tristique libero efficitur. Ut varius lobortis velit. Mauris euismod dictum luctus. Aliquam neque quam, vehicula quis mollis eget, sollicitudin id elit. Integer cursus risus ac purus fringilla, facilisis condimentum dui sagittis. Pellentesque gravida turpis dui, nec consequat lacus scelerisque et.

Sed mollis, purus at malesuada mattis, dui sem scelerisque lectus, ut fermentum leo urna eget diam. Duis facilisis turpis nibh, in commodo nisi pretium nec. Donec finibus elit et felis elementum finibus. Donec turpis purus, rutrum sit amet orci eu, tincidunt porta quam. Etiam velit mauris, varius in risus at, sodales consectetur metus. Ut urna turpis, ornare id lectus vitae, ultrices cursus urna. Nunc lacinia ullamcorper nunc in facilisis. Fusce rhoncus eros elit, at posuere ex vulputate vel. Phasellus ullamcorper neque eu ante porttitor, vel iaculis justo lobortis. Vestibulum ornare eros ex, eget porta felis ultrices vitae. Donec mauris arcu, vulputate vel lectus vitae, semper tempus ligula.

Maecenas eros dolor, auctor tincidunt enim in, bibendum ultricies magna. Aenean pellentesque interdum condimentum. Cras sollicitudin vel lorem vitae lacinia. Phasellus pulvinar suscipit volutpat. Sed ac vulputate erat, vel luctus ligula. Curabitur pretium mollis ornare. In sit amet nisl quis eros efficitur mollis. Quisque iaculis nisl sed tincidunt condimentum.

Suspendisse finibus, nunc a ultricies tempor, ex neque scelerisque diam, in varius ligula lectus ut nulla. Suspendisse dapibus mi vitae tellus consequat fermentum. Integer bibendum nisl quam, dignissim egestas augue efficitur vitae. Integer pellentesque felis nisl, id iaculis orci maximus sed. Sed blandit pretium leo, volutpat varius quam vestibulum sed. Curabitur sit amet volutpat eros. Nulla vitae tristique tellus. Duis efficitur nec libero placerat pharetra. Cras luctus neque a lorem mattis, eget laoreet magna pharetra. Nam ipsum ligula, hendrerit a orci vitae, ultricies auctor felis. Vestibulum at elit non tellus dapibus lobortis. Donec quis consectetur nibh, sit amet commodo massa. Sed ultrices, velit sed lacinia congue, ipsum mauris suscipit libero, vel posuere est ipsum eu justo. Aliquam ut venenatis neque, ut volutpat nunc. Suspendisse finibus ornare dolor sit amet tristique. Fusce a porta mauris.

Phasellus egestas augue id purus vestibulum, vel posuere ipsum dictum. Fusce eget bibendum dui. Maecenas tempus, sapien at vulputate pretium, elit felis dignissim velit, sit amet feugiat tellus quam vel turpis. Aenean quis viverra nulla, vel rhoncus augue. Morbi sit amet elit fermentum, finibus justo tristique, ultricies sapien. Sed sit amet lacinia nunc, ac pretium mauris. Donec dapibus velit non nunc sagittis semper. Ut ut nulla volutpat, rhoncus libero at, venenatis lorem. Maecenas iaculis dictum arcu eu accumsan. Donec non metus justo. Praesent non ipsum a nisl venenatis efficitur. Etiam lacinia, elit non tristique tincidunt, sapien erat malesuada diam, sed accumsan augue est eget ante. Maecenas at odio accumsan, luctus sem id, lacinia quam.

Sed at nisi erat. Integer scelerisque erat vitae aliquam sagittis. Duis id ipsum auctor orci dapibus porta vel quis augue. Vivamus consectetur dapibus turpis. Vivamus id facilisis nisi, quis fermentum urna. Ut fringilla auctor faucibus. Ut eu sapien nisi. Cras lorem risus, finibus sed scelerisque vel, commodo pulvinar lacus. In at luctus ante, non commodo quam. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus finibus, diam non blandit interdum, eros urna accumsan lectus, in aliquet libero neque in tortor. Quisque condimentum posuere ex, ac bibendum dolor posuere ac. Integer dolor metus, sollicitudin a consequat id, eleifend in magna. Pellentesque sed magna hendrerit, aliquet diam non, mollis massa. Phasellus tempus efficitur eros, quis malesuada felis pulvinar ut. Cras feugiat eget velit ut ultricies. Vivamus a tincidunt ante. Fusce neque tellus, molestie quis augue id, pulvinar maximus sapien. Sed blandit ex quis felis sodales, quis semper libero ultrices. Aenean eros lacus, bibendum at ex auctor, accumsan sagittis mauris.

Ut gravida mauris in velit maximus pellentesque. Praesent ultricies, mi eu convallis laoreet, lorem leo euismod urna, non imperdiet ligula purus vestibulum felis. Nam consequat lorem eget leo dictum, vel viverra enim commodo. Maecenas ac porttitor velit, nec dignissim velit. In tellus massa, ornare id est finibus, mattis tristique velit. Sed vitae interdum lectus. Sed placerat quam sit amet lacus lacinia, nec tristique libero efficitur. Ut varius lobortis velit. Mauris euismod dictum luctus. Aliquam neque quam, vehicula quis mollis eget, sollicitudin id elit. Integer cursus risus ac purus fringilla, facilisis condimentum dui sagittis. Pellentesque gravida turpis dui, nec consequat lacus scelerisque et.

Sed mollis, purus at malesuada mattis, dui sem scelerisque lectus, ut fermentum leo urna eget diam. Duis facilisis turpis nibh, in commodo nisi pretium nec. Donec finibus elit et felis elementum finibus. Donec turpis purus, rutrum sit amet orci eu, tincidunt porta quam. Etiam velit mauris, varius in risus at, sodales consectetur metus. Ut urna turpis, ornare id lectus vitae, ultrices cursus urna. Nunc lacinia ullamcorper nunc in facilisis. Fusce rhoncus eros elit, at posuere ex vulputate vel. Phasellus ullamcorper neque eu ante porttitor, vel iaculis justo lobortis. Vestibulum ornare eros ex, eget porta felis ultrices vitae. Donec mauris arcu, vulputate vel lectus vitae, semper tempus ligula.

Maecenas eros dolor, auctor tincidunt enim in, bibendum ultricies magna. Aenean pellentesque interdum condimentum. Cras sollicitudin vel lorem vitae lacinia. Phasellus pulvinar suscipit volutpat. Sed ac vulputate erat, vel luctus ligula. Curabitur pretium mollis ornare. In sit amet nisl quis eros efficitur mollis. Quisque iaculis nisl sed tincidunt condimentum.

Suspendisse finibus, nunc a ultricies tempor, ex neque scelerisque diam, in varius ligula lectus ut nulla. Suspendisse dapibus mi vitae tellus consequat fermentum. Integer bibendum nisl quam, dignissim egestas augue efficitur vitae. Integer pellentesque felis nisl, id iaculis orci maximus sed. Sed blandit pretium leo, volutpat varius quam vestibulum sed. Curabitur sit amet volutpat eros. Nulla vitae tristique tellus. Duis efficitur nec libero placerat pharetra. Cras luctus neque a lorem mattis, eget laoreet magna pharetra. Nam ipsum ligula, hendrerit a orci vitae, ultricies auctor felis. Vestibulum at elit non tellus dapibus lobortis. Donec quis consectetur nibh, sit amet commodo massa. Sed ultrices, velit sed lacinia congue, ipsum mauris suscipit libero, vel posuere est ipsum eu justo. Aliquam ut venenatis neque, ut volutpat nunc. Suspendisse finibus ornare dolor sit amet tristique. Fusce a porta mauris.

Phasellus egestas augue id purus vestibulum, vel posuere ipsum dictum. Fusce eget bibendum dui. Maecenas tempus, sapien at vulputate pretium, elit felis dignissim velit, sit amet feugiat tellus quam vel turpis. Aenean quis viverra nulla, vel rhoncus augue. Morbi sit amet elit fermentum, finibus justo tristique, ultricies sapien. Sed sit amet lacinia nunc, ac pretium mauris. Donec dapibus velit non nunc sagittis semper. Ut ut nulla volutpat, rhoncus libero at, venenatis lorem. Maecenas iaculis dictum arcu eu accumsan. Donec non metus justo. Praesent non ipsum a nisl venenatis efficitur. Etiam lacinia, elit non tristique tincidunt, sapien erat malesuada diam, sed accumsan augue est eget ante. Maecenas at odio accumsan, luctus sem id, lacinia quam.

Sed at nisi erat. Integer scelerisque erat vitae aliquam sagittis. Duis id ipsum auctor orci dapibus porta vel quis augue. Vivamus consectetur dapibus turpis. Vivamus id facilisis nisi, quis fermentum urna. Ut fringilla auctor faucibus. Ut eu sapien nisi. Cras lorem risus, finibus sed scelerisque vel, commodo pulvinar lacus. In at luctus ante, non commodo quam.

3.2. Figura de prueba

En la Figura 3.1, se muestra un esquema conceptual básico de una red softwarizada.



Figura 3.1: Esquema de arquitectura softwarizada.

3.3. Tabla de prueba

La Tabla 3.1 presenta una coparativa entre arquitecturas de red tradicionales y softwarizadas.

Característica	Tradicional	Softwarizada
Flexibilidad	Baja	Alta
Control Centralizado	No	Sí
Automatización	Limitada	Completa

Tabla 3.1: Comparativa de arquitecturas de red.

3.4. Algoritmo de prueba

A continuación, el Algoritmo 1 muestra un pseudocódigo de ejemplo utilizando `algorithm2e`.

Algoritmo 1: Selección del nodo con mayor capacidad

Input: Lista de nodos N **Output:** Nodo con mayor capacidad

```
1  $maxNode \leftarrow \text{null};$ 
2  $maxCap \leftarrow 0;$ 
3 foreach  $n \in N$  do
4   if  $n.capacidad > maxCap$  then
5      $maxCap \leftarrow n.capacidad;$ 
6      $maxNode \leftarrow n;$ 
7   end
8 end
9 return  $maxNode;$ 
```

Bibliografía

- [1] T. Meuser and R. Kundel, “The History of Highly Adaptive and Programmable Networks,” in *From Multimedia Communications to the Future Internet: Essays Dedicated to Ralf Steinmetz on the Occasion of His Retirement*. Springer, 2024, pp. 61–73.
- [2] D. Levy and N. McKeown, “Overhaul May Bring Better, Faster Internet to 100 Million Homes,” *Stanford University News*, 2003, accessed: 2025-06-09. [Online]. Available: <https://news.stanford.edu/news/2003/october8/network-108.html>
- [3] Open Networking Foundation, “ONF Overview,” <https://opennetworking.org/>, 2025, accessed: 2025-06-09.
- [4] T. D. Nadeau and K. Gray, *SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies*. O’Reilly Media, Inc., 2013.
- [5] ETSI, “Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action,” European Telecommunications Standards Institute (ETSI), White Paper Introductory White Paper, 2012, accessed: 2025-06-09. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper.pdf
- [6] A. Vahdat, D. Clark, and J. Rexford, “A purpose-built global network: Google’s move to SDN,” *Communications of the ACM*, 2016.
- [7] F. Tomonori, “Introduction to ryu sdn framework,” *Open Networking Summit*, pp. 1–14, 2013.
- [8] S. Racherla, D. Cain, S. Irwin, P. Ljungström, P. Patil, A. M. Tarenzio *et al.*, *Implementing ibm software defined network for virtual environments*. IBM Redbooks, 2014.
- [9] R. S. Montero, E. Rojas, A. A. Carrillo, and I. M. Llorente, “Extending the Cloud to the Network Edge.” *Computer*, vol. 50, no. 4, pp. 91–95, 2017.
- [10] 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)*. IEEE, 2018, pp. 71–79.
- [11] D. Carrascal, E. Rojas, D. Lopez-Pajares, N. Manso, J. Alvarez-Horcajo, and I. Martinez-Yelmo, “Softwarized Data-Driven Architecture for Edge Computing IIoT

BIBLIOGRAFÍA

- Environments: A Proof of Concept,” in *2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN)*. IEEE, 2025, pp. 64–68.
- [12] S. S. Hussain, M. A. Aftab, F. Nadeem, I. Ali, and T. S. Ustun, “Optimal energy routing in microgrids with IEC 61850 based energy routers,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 5161–5169, 2019.
- [13] European Union, “Directive (EU) 2018/2001 of the European Parliament and of the Council of 11 December 2018 on the promotion of the use of energy from renewable sources,” <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32018L2001>, 2018, accessed: 2025-06-10.
- [14] Gobierno de España, “Real Decreto 244/2019, de 5 de abril, por el que se regulan las condiciones administrativas, técnicas y económicas del autoconsumo de energía eléctrica,” <https://www.boe.es/eli/es/rd/2019/04/05/244>, 2019, bOE-A-2019-5089, Accessed: 2025-06-10.
- [15] R. E. Mackiewicz, “Overview of IEC 61850 and Benefits,” in *2006 IEEE Power Engineering Society General Meeting*. IEEE, 2006, pp. 8–pp.
- [16] E. Molina, E. Jacob, J. Matias, N. Moreira, and A. Astarloa, “Using Software Defined Networking to manage and control IEC 61850-based systems,” *Computers & Electrical Engineering*, vol. 43, pp. 142–154, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790614002626>
- [17] H. Maziku and S. Shetty, “Software Defined Networking enabled resilience for IEC 61850-based substation communication systems,” in *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2017, pp. 690–694.
- [18] W. Hu, Y. Hou, L. Tian, and Y. Li, “Selection of logistics distribution center location for SDN enterprises,” *Journal of Management Analytics*, vol. 2, no. 3, pp. 202–215, 2015.
- [19] D. Carrascal, E. Rojas, J. M. Arco, D. Lopez-Pajares, J. Alvarez-Horcajo, and J. A. Carral, “A comprehensive survey of in-band control in sdn: Challenges and opportunities,” *Electronics*, vol. 12, no. 6, p. 1265, 2023.
- [20] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks,” *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.
- [21] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese *et al.*, “P4: Programming protocol-independent packet processors,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014.
- [22] F. Hauser, M. Häberle, D. Merling, S. Lindner, V. Gurevich, F. Zeiger, R. Frank, and M. Menth, “A survey on data plane programming with p4: Fundamentals, advances, and applied research,” *Journal of Network and Computer Applications*, vol. 212, p. 103561, 2023.

- [23] D. Carrascal, E. Rojas, J. Alvarez-Horcajo, D. Lopez-Pajares, and I. Martínez-Yelmo, “Analysis of P4 and XDP for IoT Programmability in 6G and Beyond,” *IoT*, vol. 1, no. 2, pp. 605–622, 2020.
- [24] P4 Language Consortium, “P4Runtime Specification,” <https://p4.org/p4-spec/p4runtime/main/P4Runtime-Spec.html>, 2023, accessed: 2025-06-16.
- [25] R. Vilalta, R. Muñoz, R. Casellas, R. Martínez, V. López, O. G. de Dios, A. Pastor, G. P. Katsikas, F. Klaedtke, P. Monti, A. Mozo, T. Zinner, H. Øverby, S. Gonzalez-Diaz, H. Lønsethagen, J.-M. Pulido, and D. King, “TeraFlow: Secured Autonomic Traffic Management for a Tera of SDN flows,” in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2021, pp. 377–382.
- [26] D. López Pajares, “Nuevos conmutadores de red para redes integradas con SDN,” Tesis doctoral, Universidad de Alcalá, 2021, accessed: 2025-06-16. [Online]. Available: <https://ebuah.uah.es/dspace/handle/10017/51030>
- [27] C. Suo, I.-C. Tsai, and C. H.-P. Wen, “ERIC: Economical & reconfigurable hybrid-band control for software-defined datacenter network,” in *2016 International Conference on Information Networking (ICOIN)*. IEEE, 2016, pp. 214–219.
- [28] A. Jalili, H. Nazari, S. Namvarasl, and M. Keshtgari, “A comprehensive analysis on control plane deployment in SDN: In-band versus out-of-band solutions,” in *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, 2017, pp. 1025–1031.
- [29] D. Kafetzis, S. Vassilaras, G. Vardoulas, and I. Koutsopoulos, “Software-Defined Networking meets Software-Defined Radio in Mobile Ad hoc Networks: State of the Art and Future Directions,” *IEEE Access*, 2022.
- [30] I. I. Awan, N. Shah, M. Imran, M. Shoaib, and N. Saeed, “An improved mechanism for flow rule installation in-band SDN,” *Journal of Systems Architecture*, vol. 96, pp. 1–19, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1383762118304739>
- [31] S. Khorsandroo, A. G. Sánchez, A. S. Tosun, J. Arco, and R. Doriguzzi-Corin, “Hybrid SDN evolution: A comprehensive survey of the state-of-the-art,” *Computer Networks*, vol. 192, p. 107981, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128621001109>
- [32] E. Rojas, R. Amin, C. Guerrero, M. Savi, and A. Rastegarnia, “Challenges and Solutions for hybrid SDN,” *Computer Networks*, vol. 195, p. 108198, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128621002498>
- [33] M. K. MURTADHA, “SDN based device to device communication architecture for 5G mobile networks,” *Journal of Engineering Science and Technology*, vol. 16, no. 4, pp. 3033–3047, 2021.

- [34] J. Guo, L. Yang, X. Liu, Q. Chen, C. Fan, and X. Li, “Performance Modelling and Evaluation of In-Band Control Mode in Software-Defined Satellite Networks Based on Queuing Theory,” in *2021 2nd International Conference on Computing, Networks and Internet of Things*, ser. CNIOT '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: <https://doi.org/10.1145/3468691.3468697>
- [35] Y. Shi, Q. Yang, X. Huang, D. Li, and X. Huang, “An SDN-Enabled Framework for a Load-Balanced and QoS-Aware Internet of Underwater Things,” *IEEE Internet of Things Journal*, pp. 1–1, 2022.
- [36] I. Maity, R. Dhiman, and S. Misra, “EnPlace: Energy-Aware Network Partitioning for Controller Placement in SDN,” *IEEE Transactions on Green Communications and Networking*, 2022.
- [37] S. Chattopadhyay, S. Chatterjee, S. Nandi, and S. Chakraborty, “Aloe: An elastic auto-scaled and self-stabilized orchestration framework for iot applications,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 802–810.
- [38] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, “In-band control, queuing, and failure recovery functionalities for openflow,” *IEEE Network*, vol. 30, no. 1, pp. 106–112, 2016.
- [39] P. Goltsmann, M. Zitterbart, A. Hecker, and R. Bless, “Towards a Resilient In-Band SDN Control Channel,” *Universität Tübingen*, 2017.
- [40] A. S. Khakhalin and E. V. Chemeritskiy, “A reliable in-band control in a Software-Defined Network,” *Journal of Theoretical & Applied Information Technology*, vol. 95, no. 17, 2017.
- [41] P. M. Mohan, T. Truong-Huu, and M. Gurusamy, “Towards resilient in-band control path routing with malicious switch detection in SDN,” in *2018 10th International Conference on Communication Systems Networks (COMSNETS)*, 2018, pp. 9–16.
- [42] A. Raza, A. Gohar, and S. Lee, “MPTCP based in-band controlling for the software defined networks,” in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 2017, pp. 163–167.
- [43] S. González, A. De la Oliva, C. J. Bernardos, and L. M. Contreras, “Towards a Resilient Openflow Channel Through MPTCP,” in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2018, pp. 1–5.
- [44] A. S. M. Asadujjaman, E. Rojas, M. S. Alam, and S. Majumdar, “Fast Control Channel Recovery for Resilient In-band OpenFlow Networks,” in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 19–27.
- [45] D. Lopez-Pajares, J. Alvarez-Horcajo, E. Rojas, A. S. M. Asadujjaman, and I. Martinez-Yelmo, “Amaru: Plug&Play Resilient In-Band Control for SDN,” *IEEE Access*, vol. 7, pp. 123 202–123 218, 2019.

- [46] B. Görkemli, S. Tathcıoğlu, A. M. Tekalp, S. Civanlar, and E. Lokman, “Dynamic Control Plane for SDN at Scale,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2688–2701, 2018.
- [47] P. Holzmann and M. Zitterbart, “Izzy: A Distributed Routing Protocol for In-band SDN Control Channel Connectivity,” in *2019 IEEE 44th LCN Symposium on Emerging Topics in Networking (LCN Symposium)*, 2019, pp. 18–25.
- [48] K. Kumazoe, M. Shibata, and M. Tsuru, “A P4 BMv2-Based Feasibility Study on a Dynamic In-Band Control Channel for SDN,” in *Advances in Intelligent Networking and Collaborative Systems*, L. Barolli and H. Miwa, Eds. Cham: Springer International Publishing, 2022, pp. 442–451.
- [49] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, “Automatic bootstrapping of openflow networks,” in *2013 19th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN)*, 2013, pp. 1–6.
- [50] C.-C. Tu, P.-W. Wang, and T.-c. Chiueh, “In-Band Control for an Ethernet-Based Software-Defined Network,” in *Proceedings of International Conference on Systems and Storage*, ser. SYSTOR 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 1–11. [Online]. Available: <https://doi.org/10.1145/2611354.2611359>
- [51] Y.-L. Su, I.-C. Wang, Y.-T. Hsu, and C. H.-P. Wen, “FASIC: A Fast-Recovery, Adaptively Spanning In-Band Control Plane in Software-Defined Network,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.
- [52] E. Sakic, M. Avdic, A. Van Bemten, and W. Kellerer, “Automated Bootstrapping of A Fault-Resilient In-Band Control Plane,” in *Proceedings of the Symposium on SDN Research*, ser. SOSR ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–13. [Online]. Available: <https://doi.org/10.1145/3373360.3380829>
- [53] F. Wu and A. Tian, “rXstp: A Topology Discovery Mechanism Based on Rapid Spanning Tree for SDN In-Band Control,” in *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2021, pp. 703–706.
- [54] M. Silva Freitas, R. Oliveira, D. Molinos, J. Melo, P. Frosi Rosa, and F. de Oliveira Silva, “ConForm: In-band Control Plane Formation Protocol to SDN-Based Networks,” in *2020 International Conference on Information Networking (ICOIN)*, 2020, pp. 574–579.
- [55] C.-Y. Li, L.-H. Yen, K.-H. Chi, and C.-C. Tseng, “One-Pass In-Band Automatic Bootstrapping for OpenFlow Switches,” *IEEE Access*, vol. 9, pp. 153 349–153 359, 2021.
- [56] L. Schiff, S. Schmid, and M. Canini, “Medieval: Towards A Self-Stabilizing, Plug & Play, In-Band SDN Control Network,” in *ACM Sigcomm Symposium on SDN Research (SOSR)*, 2015. [Online]. Available: <http://eprints.cs.univie.ac.at/5745/>

- [57] —, “Ground control to major faults: Towards a fault tolerant and adaptive SDN control network,” in *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*. IEEE, 2016, pp. 90–96.
- [58] L. Schiff, S. Schmid, and P. Kuznetsov, “In-Band Synchronization for Distributed SDN Control Planes,” *SIGCOMM Comput. Commun. Rev.*, vol. 46, no. 1, p. 37–43, January 2016. [Online]. Available: <https://doi.org/10.1145/2875951.2875957>
- [59] M. Canini, I. Salem, L. Schiff, E. M. Schiller, and S. Schmid, “Renaissance: A Self-Stabilizing Distributed SDN Control Plane,” in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 233–243.
- [60] —, “Renaissance: A self-stabilizing distributed SDN control plane using in-band communications,” *Journal of Computer and System Sciences*, vol. 127, pp. 91–121, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022000022000150>
- [61] B. Görkemli, A. M. Parlakışık, S. Civanlar, A. Ulaş, and A. M. Tekalp, “Dynamic management of control plane performance in software-defined networks,” in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE, 2016, pp. 68–72.
- [62] R. Hark, A. Rizk, N. Richerzhagen, B. Richerzhagen, and R. Steinmetz, “Isolated in-band communication for distributed SDN controllers,” in *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, 2017, pp. 1–2.
- [63] M. Canini, I. Salem, L. Schiff, E. M. Schiller, and S. Schmid, “A Self-Organizing Distributed and In-Band SDN Control Plane,” in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2656–2657.
- [64] K.-Y. Chan, C.-H. Chen, Y.-H. Chen, Y.-J. Tsai, S. S. W. Lee, and C.-S. Wu, “Fast Failure Recovery for In-Band Controlled Multi-Controller OpenFlow Networks,” in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 396–401.
- [65] P. Holzmann, A. Hecker, and M. Zitterbart, “Towards a Distributed Routing Protocol for In-Band Control Channel with Elastic Controller Clusters,” in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.
- [66] Open Networking Foundation, “SDN Architecture, Issue 1.1,” Open Networking Foundation, Technical Reference (TR) TR-521, 2016. [Online]. Available: https://opennetworking.org/wp-content/uploads/2014/10/TR-521_SDN_Architecture_issue_1.1.pdf
- [67] L. Euler, “Solutio problematis ad geometriam situs pertinentis,” *Commentarii academiae scientiarum Petropolitanae*, pp. 128–140, 1741.
- [68] P. Fraigniaud, A. Pelc, D. Peleg, and S. Perennes, “Assigning labels in unknown anonymous networks,” in *Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing*, 2000, pp. 101–111.

- [69] E. Rojas and G. Ibáñez, “Torii-HLMAC: A distributed, fault-tolerant, zero configuration fat tree data center architecture with multiple tree-based addressing and forwarding,” in *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 2523–2528.
- [70] E. Rojas, H. Hosseini, C. Gomez, D. Carrascal, and J. R. Cotrim, “Outperforming RPL with scalable routing based on meaningful MAC addressing,” *Ad Hoc Networks*, vol. 114, p. 102433, 2021.
- [71] E. Rojas, J. Alvarez-Horcajo, I. Martinez-Yelmo, J. M. Arco, and J. A. Carral, “GA3: scalable, distributed address assignment for dynamic data center networks,” *Annals of Telecommunications*, vol. 72, no. 11, pp. 693–702, 2017.
- [72] E. Rojas, J. Alvarez-Horcajo, I. Martinez-Yelmo, J. M. Arco, and M. Briso-Montiano, “Scalable and Reliable Data Center Networks by Combining Source Routing and Automatic Labelling,” *Network*, vol. 1, no. 1, pp. 11–27, 2021.
- [73] M. Walraed-Sullivan, R. Niranjana Mysore, K. Marzullo, and A. Vahdat, “A Randomized Algorithm for Label Assignment in Dynamic Networks,” *Open Access Publications from the University of California*, 2013.