Universidad
Rey Juan Carlos

Máster Universitario en Ingeniería de Sistemas de Información

2023-2024

Trabajo Fin de Máster

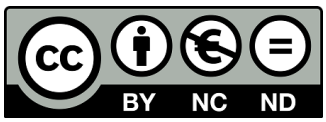
“ Estudio, diseño y desarrollo de una solución multidominio para redes deterministas”

David Rico Menéndez

Tutor/es

David Granada Mejía

Madrid, 12 de Junio de 2024



RESUMEN

En este Trabajo Fin de Máster, se presenta un exhaustivo estudio, diseño y desarrollo de una solución multidominio para redes deterministas, integrando las tecnologías avanzadas de Time-Sensitive Networking (TSN) y 3rd Generation Partnership Project (3GPP). La finalidad principal de esta investigación es garantizar la coherencia y precisión necesarias en ambientes altamente dinámicos y críticos, como la industria 4.0, aplicaciones médicas, y sistemas de transporte autónomo.

A través de un enfoque experimental meticuloso, se ha implementado y evaluado una red determinista utilizando componentes comerciales de última generación de TSN y un núcleo 5G de código abierto configurado en modo Standalone (SA). La implementación incluye la utilización de hardware especializado para la infraestructura de TSN, y la plataforma OpenAirInterface (OAI) para el núcleo y radio 5G. Este despliegue ha permitido explorar la interacción entre dominios tecnológicos heterogéneos, enfocándose en la optimización del tráfico determinista y de mejor esfuerzo (BE) bajo diferentes condiciones de carga de la red.

Los resultados obtenidos demuestran la viabilidad, robustez y eficiencia de la solución propuesta. Se ha comprobado que la red es capaz de manejar múltiples flujos de tráfico determinista con latencias controladas y mínimas variaciones de retardo (jitter), incluso en presencia de tráfico BE significativo. Además, la integración de mecanismos de Quality of Service (QoS) y técnicas de Traffic Shaping ha sido crucial para mantener la calidad del servicio en aplicaciones críticas.

Este trabajo también profundiza en los desafíos técnicos encontrados durante la implementación, tales como la configuración óptima de los parámetros de TSN y 5G, y las soluciones adoptadas para superarlos. Asimismo, se discuten las implicaciones socioeconómicas de la implementación de estas tecnologías en aplicaciones reales, incluyendo los potenciales cambios en el mercado laboral, las consideraciones éticas y de privacidad de datos, y el impacto en la percepción pública de las tecnologías avanzadas.

Palabras clave: redes deterministas, TSN, 3GPP, 5G SA, automatización industrial, industria 4.0, baja latencia, alta fiabilidad

ABSTRACT

In this Master's Thesis, an exhaustive study, design, and development of a multi-domain solution for deterministic networks integrating advanced technologies such as Time-Sensitive Networking (TSN) and the 3rd Generation Partnership Project (3GPP) is presented. The main objective of this research is to ensure the coherence and precision

In this Master's Thesis, an exhaustive study, design, and development of a multidomain solution for deterministic networks integrating advanced technologies such as Time-Sensitive Networking (TSN) and the 3rd Generation Partnership Project (3GPP) is presented. The main objective of this research is to ensure the coherence and precision necessary in highly dynamic and critical environments, such as Industry 4.0, medical applications, and autonomous transportation systems.

Through a meticulous experimental approach, a deterministic network has been implemented and evaluated using state-of-the-art commercial TSN components and an open-source 5G core configured in Standalone (SA) mode. The implementation includes the use of specialized hardware for the TSN infrastructure and the OpenAirInterface (OAI) platform for the 5G core and radio. This deployment has allowed for the exploration of the interaction between heterogeneous technological domains, focusing on optimizing deterministic and best-effort (BE) traffic under varying network load conditions.

The results demonstrate the feasibility, robustness, and efficiency of the proposed solution. The network has been proven capable of handling multiple deterministic traffic flows with controlled latencies and minimal jitter, even in the presence of significant BE traffic. Additionally, the integration of Quality of Service (QoS) mechanisms and Traffic Shaping techniques has been crucial in maintaining service quality in critical applications.

This work also delves into the technical challenges encountered during the implementation, such as the optimal configuration of TSN and 5G parameters, and the solutions adopted to overcome them. Furthermore, it discusses the socio-economic implications of implementing these technologies in real-world applications, including potential changes in the labor market, ethical and data privacy considerations, and the impact on public perception of advanced technologies.

Keywords: deterministic networks, TSN, 3GPP, 5G SA, industrial automation, Industry 4.0, low latency, high reliability

Índice general

Índice de figuras

Índice de tablas

1. INTRODUCCIÓN

El aumento constante en la demanda de servicios críticos y aplicaciones sensibles al tiempo ha llevado al desarrollo de redes que pueden garantizar altos niveles de determinismo. Este trabajo se centra en el estudio, diseño y desarrollo de una solución multidominio para redes deterministas que aseguren la coherencia y precisión necesarias en ambientes altamente dinámicos y críticos. A través de la integración de tecnologías y estándares emergentes, este estudio busca abordar las complejidades inherentes a la gestión y operación de redes deterministas en diferentes dominios tecnológicos, con un enfoque particular en 3rd Generation Partnership Project (3GPP) y Institute of Electrical and Electronics Engineers (IEEE) 802.1 TSN. Esto significa que trabajaremos principalmente sobre redes cableadas e inalámbricas móviles. [1]

La importancia de las redes deterministas cobra especial relevancia en el contexto de la Industria 4.0, donde la automatización y la interconectividad entre máquinas y sistemas exigen una sincronización y una respuesta en tiempo real. En estos entornos, cualquier retraso o imprecisión puede resultar en fallos costosos o interrupciones significativas de la producción. Las soluciones llamadas TSN emergen como un pilar crucial, permitiendo que los sistemas de comunicación soporten aplicaciones críticas con requerimientos estrictos de baja latencia y alta confiabilidad. La implementación de redes TSN en redes cableadas tradicionales en conjunción con las tecnologías 3GPP para redes móviles puede revolucionar la industria, facilitando la adopción de soluciones avanzadas como robots autónomos con gran rango de acción, sistemas de monitoreo en tiempo real y operaciones de mantenimiento predictivo. [2]

1.1. Motivación

El interés por las redes deterministas ha crecido significativamente con la introducción de tecnologías como 5G, que prometen capacidades mejoradas en términos de ancho de banda y latencia. [3] Sin embargo, la verdadera promesa de estas tecnologías solo puede realizarse a través de una integración efectiva que permita la coexistencia y cooperación sin fisuras entre diferentes dominios tecnológicos. [4] Este Trabajo de Fin de Máster (TFM) se motiva por la necesidad de desarrollar una plataforma que no solo soporte requerimientos estrictos de tiempo y fiabilidad, sino que también facilite la interoperabilidad entre distintas tecnologías de red, como TSN y 3GPP, para aplicaciones que van desde la manufactura avanzada hasta la medicina de precisión. [5]

1.2. Objetivos

El principal objetivo de este TFM es diseñar y desarrollar un plano de datos determinista multidominio que integre las tecnologías de TSN y 3GPP bajo una capa de Deterministic Networking (DetNet). Los objetivos específicos incluyen:

1. Estudiar y analizar los requerimientos de determinismo y fiabilidad en aplicaciones críticas.
2. Diseñar una solución que integre eficazmente los dominios de TSN y 3GPP.
3. Desarrollar e implementar la solución propuesta utilizando un entorno de Entorno de prueba (Testbed).
4. Evaluar el desempeño de la solución en términos de latencia, Variabilidad en la latencia de los paquetes (Jitter) y pérdida de paquetes.

1.3. Estructura de la memoria

La estructura de esta memoria se organiza de la siguiente manera:

- **Estado del Arte (Capítulo 2):** se lleva a cabo una revisión del estado del arte en redes deterministas, abordando los conceptos y aplicaciones principales de estas tecnologías. Se examinan las iniciativas recientes y los avances en los estándares pertinentes de IEEE, Internet Engineering Task Force (IETF) y 3GPP que impactan en el diseño y la implementación de redes deterministas.
- **Marco Teórico (Capítulo 3):** se exponen los principios fundamentales de las redes deterministas y las tecnologías de conectividad como Tecnología de redes de área local (Ethernet) y 5G. Este capítulo destaca la importancia de estos principios para asegurar la fiabilidad y el tiempo de respuesta en aplicaciones críticas.
- **Caracterización de los Dominios Tecnológicos (Capítulo 4):** se proporciona una descripción detallada de las tecnologías clave como TSN en Ethernet y las características deterministas en 3GPP. Se analiza cómo cada tecnología contribuye a la solución multidominio propuesta.
- **Solución Propuesta (Capítulo 5):** se describe en profundidad el diseño e implementación del plano de datos determinista que integra múltiples dominios tecnológicos. Incluye detalles sobre la arquitectura del sistema, implementación y roles específicos de los componentes.
- **Experimentación y Resultados (Capítulo 6):** se explican los experimentos realizados para evaluar el rendimiento de la solución propuesta. Se analizan los resultados obtenidos, destacando cómo la solución maneja los requisitos de determinismo y fiabilidad.

- **Impacto Socio-Económico (Capítulo 7):** se analiza el impacto que la implementación de la solución puede tener en la sociedad y la economía, considerando factores como la creación de empleo y la aceptación de la tecnología.
- **Sostenibilidad y Consideraciones Éticas (Capítulo 8):** Se consideran las implicaciones ambientales de las tecnologías de red y las cuestiones éticas relacionadas con la automatización y la inteligencia artificial.
- **Planificación Temporal y Presupuesto (Capítulo 9):** se detalla la planificación de las tareas del proyecto y se proporciona una estimación del presupuesto necesario, incluyendo los costos y recursos implicados.
- **Conclusiones y Trabajo Futuro (Capítulo 10):** se presentan las conclusiones finales del trabajo, discutiendo las limitaciones encontradas y sugiriendo futuras direcciones de investigación que podrían explorarse para extender y mejorar la solución desarrollada.

2. ESTADO DEL ARTE

En este capítulo, se realiza una revisión exhaustiva del estado del arte en redes deterministas, enfocándose en los conceptos fundamentales, aplicaciones, tecnologías y estándares relevantes. La creciente demanda de aplicaciones críticas y sensibles al tiempo ha impulsado el desarrollo y la adopción de redes deterministas, que ofrecen garantías estrictas en términos de latencia, Jitter y pérdida de paquetes. [6] A continuación, se presentan los conceptos y aplicaciones de las redes deterministas y se revisan las tecnologías y estándares principales que las soportan.

2.1. Redes Deterministas: Conceptos y Aplicaciones

Las redes deterministas se caracterizan por su capacidad para proporcionar comunicaciones fiables y predecibles, esenciales para aplicaciones que requieren precisión en el tiempo y alta disponibilidad. A diferencia de las redes convencionales, donde la latencia y el Jitter pueden variar considerablemente, las redes deterministas garantizan que los datos se transmitan dentro de parámetros temporales estrictamente definidos. [7]

Primero, revisaremos aquellos conceptos clave a la hora del diseño de redes deterministas:

- **Determinismo:** en el contexto de las redes, el determinismo se refiere a la capacidad de predecir con precisión el comportamiento del tráfico de red, asegurando que los paquetes se entreguen en intervalos de tiempo predefinidos. Esto se logra mediante técnicas como la planificación de tráfico y la reserva de recursos, que permiten controlar estrictamente el flujo de datos en la red. [8]
- **Latencia y Jitter:** la latencia es el tiempo que tarda un paquete de datos en viajar desde la fuente hasta el destino. El Jitter se refiere a la variabilidad en la latencia de los paquetes. Las redes deterministas minimizan ambos parámetros para garantizar comunicaciones consistentes. La latencia baja y constante es crucial para aplicaciones como la automatización industrial y la telemedicina, donde incluso pequeños retrasos pueden tener consecuencias significativas. [9]
- **Fiabilidad:** la capacidad de la red para entregar paquetes sin pérdidas o errores. Esto es crucial para aplicaciones que no pueden tolerar interrupciones en el flujo de datos. Las técnicas como la duplicación de paquetes y la corrección de errores se utilizan para aumentar la fiabilidad de las transmisiones en redes deterministas. [10]

Las aplicaciones de las redes deterministas abarcan una amplia gama de sectores, incluyendo:

- **Automatización Industrial:** en las fábricas inteligentes y los sistemas de producción automatizados, las redes deterministas aseguran que los robots y maquinaria trabajen en perfecta sincronización. Por ejemplo, en una línea de montaje, los robots deben realizar tareas en secuencia precisa y coordinada para evitar colisiones y maximizar la eficiencia. La capacidad de prever con exactitud cuándo y dónde se transmitirán los datos es esencial para mantener la producción en marcha sin interrupciones [11].

- **Vehículos Autónomos:** la operación segura y eficiente de vehículos autónomos depende de comunicaciones en tiempo real. Los vehículos necesitan intercambiar datos sobre su entorno y coordinar acciones con otros vehículos e infraestructura vial. Las redes deterministas permiten que estos datos se transmitan con la baja latencia necesaria para tomar decisiones rápidas y seguras. Por ejemplo, un coche autónomo que se aproxima a una intersección necesita recibir información en milisegundos para evitar colisiones [12].

- **Atención Médica:** en aplicaciones médicas críticas, como la cirugía robótica y el monitoreo remoto de pacientes, las redes deterministas garantizan la transmisión de datos médicos sensibles en tiempo real. Durante una operación quirúrgica asistida por robot, cualquier retraso o pérdida de datos podría poner en riesgo la vida del paciente. La fiabilidad y la baja latencia son también cruciales en el monitoreo remoto de pacientes, donde las lecturas de los dispositivos médicos deben llegar sin demoras al personal médico [13].

- **Redes Eléctricas Inteligentes:** la gestión y control de redes eléctricas distribuidas dependen de comunicaciones deterministas para asegurar la estabilidad y eficiencia del suministro eléctrico. En una red eléctrica inteligente, los sensores y dispositivos de control deben comunicarse de manera rápida y precisa para responder a las fluctuaciones en la demanda y oferta de energía. La capacidad de transmitir datos en tiempo real permite a los operadores de red prevenir apagones y optimizar la distribución de energía [14].

La adopción de redes deterministas promete transformar estos sectores, ofreciendo un nivel de control y predictibilidad que no es posible con las redes tradicionales. Los avances en tecnologías como TSN y las mejoras en los estándares de comunicación juegan un papel crucial en la implementación de estas redes.

2.1.1. Importancia en la Industria 4.0

En el contexto de la Industria 4.0, la importancia de las redes deterministas se magnifica considerablemente. La Industria 4.0, caracterizada por la digitalización avanzada, la automatización y el uso intensivo de datos, requiere una infraestructura de red que pueda manejar comunicaciones críticas en tiempo real. Las redes deterministas, y en particular

los servicios de TSN, ofrecen la capacidad de garantizar tiempos de respuesta precisos y una baja latencia, aspectos cruciales para aplicaciones como la robótica avanzada, la fabricación inteligente y los sistemas ciberfísicos.

La implementación de servicios TSN puede transformar el paradigma actual, permitiendo la sincronización exacta de máquinas y dispositivos, reduciendo tiempos de inactividad y aumentando la eficiencia operativa. Esto no solo mejora la productividad, sino que también abre nuevas posibilidades para innovaciones en el diseño y operación de sistemas industriales, estableciendo un nuevo estándar en la confiabilidad y precisión de las comunicaciones industriales. Por ejemplo, en una planta de manufactura, TSN permite la coordinación precisa de procesos complejos, desde la cadena de suministro hasta el ensamblaje final, optimizando cada paso y minimizando errores y retrasos.[15][16]

En los siguientes apartados, se revisarán las principales tecnologías y estándares que soportan el desarrollo y la implementación de redes deterministas.

2.2. Revisión de Tecnologías y Estándares

El desarrollo y la implementación de redes deterministas se sustentan en una serie de tecnologías y estándares que han sido específicamente diseñados para garantizar comunicaciones fiables y predecibles. En esta sección, se revisarán los estándares y especificaciones más relevantes que abordan el determinismo en redes. Estos incluyen los desarrollos en el marco de IEEE 802.1 TSN [17] [18], los esfuerzos hacia el determinismo en 3GPP [19] y las soluciones propuestas por el grupo de trabajo DetNet del IETF. Cada uno de estos contribuye de manera significativa a la creación de una infraestructura de red que puede satisfacer las demandas estrictas de aplicaciones críticas y sensibles al tiempo.

2.2.1. IEEE 802.1 TSN

IEEE 802.1 TSN es un conjunto de estándares que extiende las capacidades de Ethernet para soportar aplicaciones con requisitos estrictos de tiempo y fiabilidad. TSN se basa en varios estándares individuales que, combinados, permiten la creación de redes deterministas [20].

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés) es una organización profesional global dedicada al avance de la tecnología en beneficio de la humanidad. Fundada en 1963, IEEE es conocida por su extenso trabajo en el desarrollo de estándares técnicos en diversas áreas, incluyendo la informática, la electrónica y las telecomunicaciones. Los estándares de IEEE son ampliamente reconocidos y adoptados a nivel mundial debido a su rigor técnico, su proceso de desarrollo basado en el consenso y su contribución significativa a la interoperabilidad y la innovación tecnológica [21].

Los estándares de IEEE son relevantes porque proporcionan una base común sobre la cual los fabricantes y desarrolladores pueden construir sistemas y dispositivos inter-

operables. Estos estándares aseguran que los productos de diferentes proveedores puedan trabajar juntos sin problemas, lo cual es esencial en la era de la conectividad y la digitalización. En el contexto de redes deterministas, los estándares de IEEE como los de la familia 802.1 TSN son cruciales para asegurar que las redes puedan cumplir con los requisitos estrictos de aplicaciones críticas en términos de latencia, Jitter y fiabilidad [22].

El estándar IEEE 802.1 TSN es particularmente relevante para el diseño y desarrollo de una solución multidominio para redes deterministas debido a su enfoque en la creación de una infraestructura de red que pueda manejar aplicaciones sensibles al tiempo. Las características y mecanismos definidos en los diversos subestándares de TSN son fundamentales para garantizar que las redes Ethernet puedan ofrecer el nivel de precisión y fiabilidad necesario para soportar aplicaciones críticas [23].

A continuación, se describen en detalle los subestándares más importantes de IEEE 802.1 TSN:

- **IEEE 802.1Q:** este estándar define la arquitectura y los mecanismos para la priorización del tráfico y la configuración de redes VLAN (Redes de Área Local Virtual). Es fundamental para el funcionamiento de TSN ya que proporciona la base para la segmentación del tráfico y la priorización de paquetes, elementos esenciales para el control de latencia y Jitter en redes deterministas.[24]

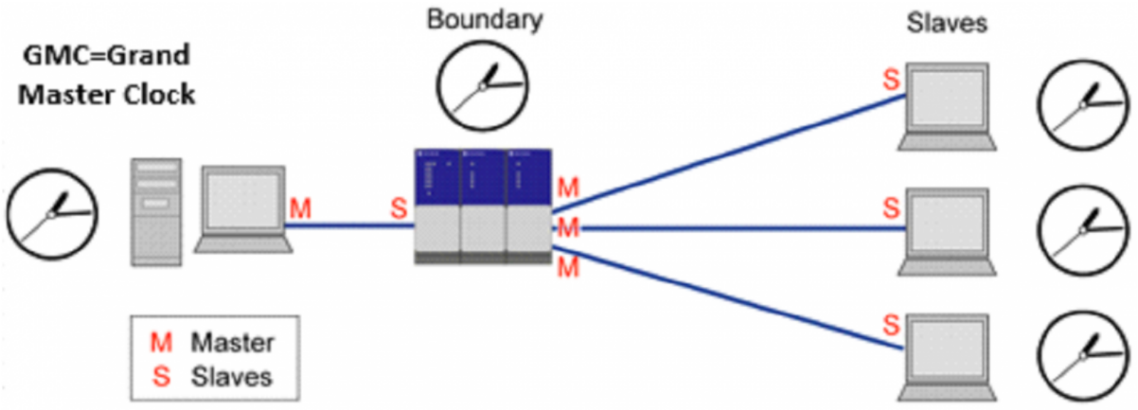


Fig. 2.1. Funcionamiento de PTP

- **IEEE 802.1AS-2020 – Timing and Synchronization:** este estándar especifica el protocolo para la sincronización del tiempo en redes de área local. Se basa en el IEEE 1588 Precision Time Protocol (PTP) (Ver Figura 2.1) y permite la sincronización precisa de relojes en todos los dispositivos de la red, asegurando que todos los elementos operen con una noción común del tiempo como se puede observar en la figura 2.2. Esto es crucial para aplicaciones que requieren coordinación precisa y sincronización de eventos. La precisión de tiempo es esencial para mantener la coherencia y la sincronización en redes deterministas, especialmente en entornos industriales y de automatización.[25]

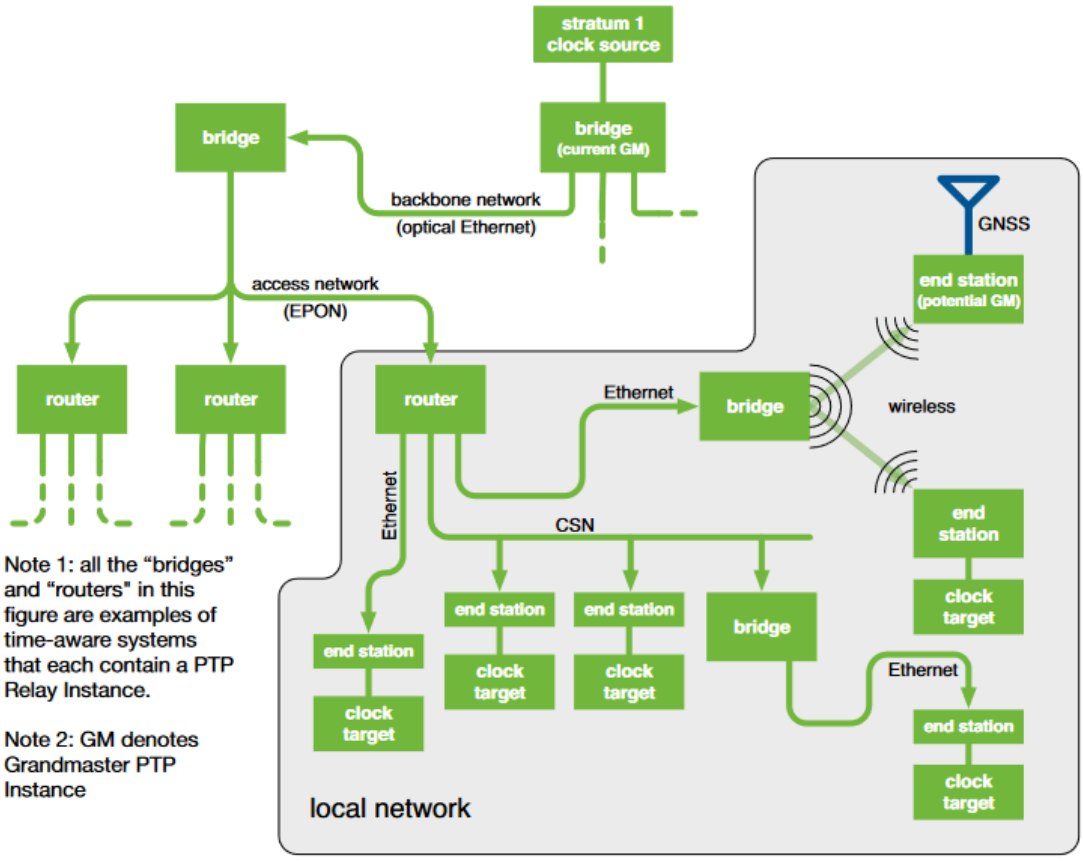


Fig. 2.2. Ejemplo de red Time Aware segun IEEE

- **IEEE 802.1Qbv – Time-Aware Shaper:** el Time-Aware Shaper (TAS) permite la programación de transmisiones de tráfico basado en el tiempo. Define ventanas de tiempo durante las cuales ciertos tipos de tráfico pueden ser transmitidos, garantizando así que el tráfico crítico tenga acceso prioritario a los recursos de la red en momentos específicos, como se muestra en la figura 2.3. Esto reduce la latencia y el Jitter para el tráfico de alta prioridad. TAS es vital para aplicaciones donde la puntualidad es crítica, como en sistemas de control industrial y vehículos autónomos. [26]

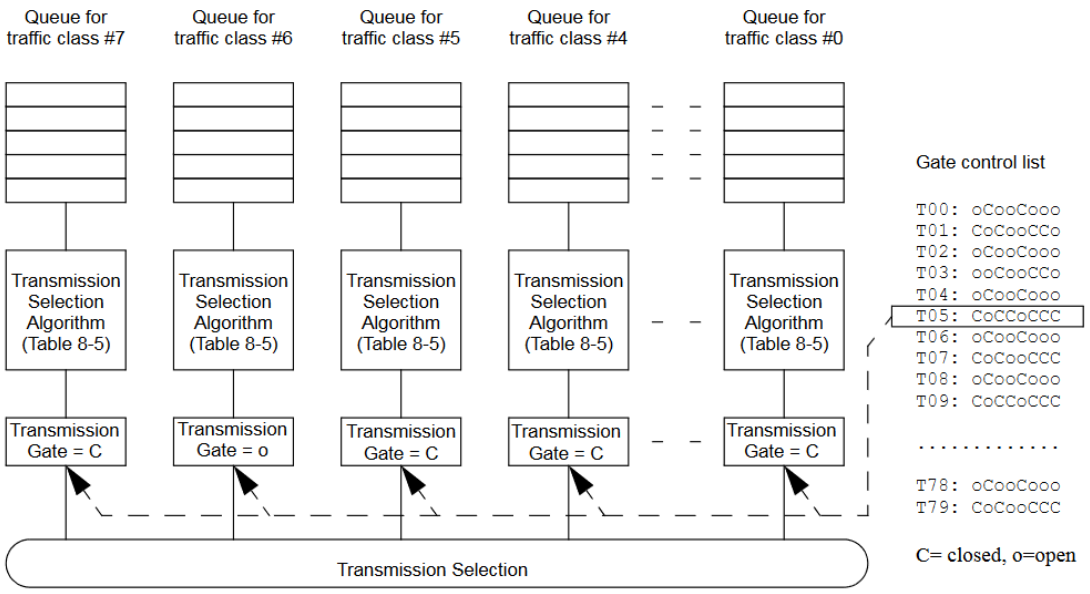


Fig. 2.3. TSN 802.1Qbv Bridge operation mode

- **IEEE 802.1CB – Frame Replication and Elimination for Reliability:** este estándar introduce la replicación y eliminación de tramas para mejorar la fiabilidad de la red. Al enviar múltiples copias de un paquete a través de diferentes rutas y eliminar las duplicadas en el destino, se puede asegurar la entrega de datos incluso en presencia de fallos en la red En la figura 2.4 se muestra esta funcionalidad sobre varios flujos en una red de ejemplo. Esto es especialmente importante en aplicaciones donde la pérdida de datos no es tolerable, como en sistemas de control industrial y comunicaciones críticas en salud.[27]

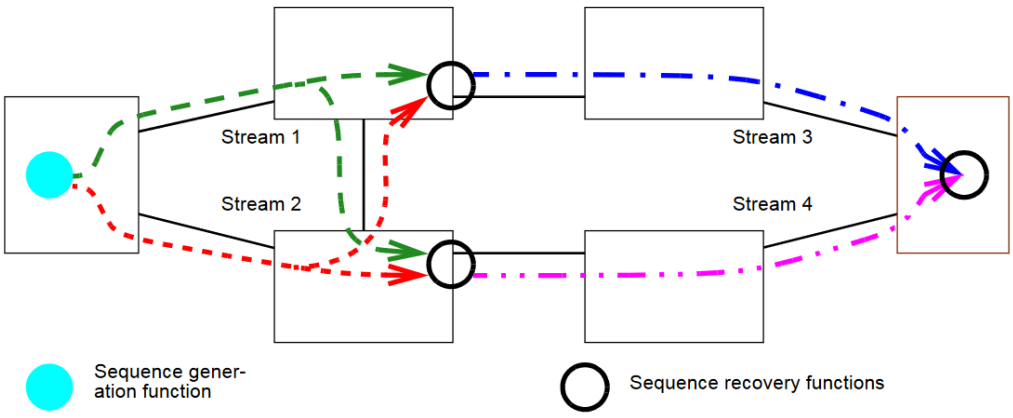


Fig. 2.4. Ejemplo básico de FRER en una red modelo

- **IEEE 802.1Qci – Per-Stream Filtering and Policing:** este estándar define mecanismos para la filtración y la regulación del tráfico por flujo. Permite la configuración de políticas específicas para cada flujo de datos, asegurando que solo el tráfico autorizado pueda pasar y que los recursos de la red se utilicen de manera eficiente.

Esto ayuda a mantener la integridad y la calidad del servicio en la red, permitiendo una gestión granular del tráfico y protegiendo la red de comportamientos no deseados [28].

En conjunto, estos estándares proporcionan una base robusta para la creación de redes deterministas mediante TSN. La combinación de sincronización precisa del tiempo, gestión sofisticada del tráfico y mecanismos de alta fiabilidad permite que las redes Ethernet soporten aplicaciones con requisitos estrictos de tiempo y fiabilidad, transformando la manera en que se diseñan y operan las redes para aplicaciones críticas en la Industria 4.0 y más allá.

2.2.2. Determinismo en 3GPP

Los logros en términos de determinismo y Redes Sensibles al Tiempo (TSN) en el contexto de las redes 3GPP y 5G han experimentado un progreso y desarrollo significativo. La integración de las redes 5G con TSN ha sido un aspecto crucial de estos avances, estandarizado en la versión 16 de 3GPP y mejorado en la versión 17, con el objetivo de soportar comunicaciones deterministas. Con el nuevo 5GC (5G Stand Alone) y utilizando nuevas técnicas TSN, es posible controlar el Jitter garantizando latencias acotadas, demostrando un paso fundamental hacia la comunicación determinista en los sistemas de acceso 5G-Avanzado y 6G [29].

Un nuevo esquema de programación de concesión configurada para 5G propuesto ofrece un mejor soporte para múltiples flujos de tráfico TSN con diversas periodicidades, coordinándose con los programadores TSN. Esto muestra una integración mejorada de TSN en las redes 5G, lo cual es crucial para aplicaciones industriales que demandan alta fiabilidad y baja latencia [30].

Además, la tecnología TSN es cada vez más reconocida como una solución prometedora para satisfacer los requisitos de conectividad del 5G, particularmente para servicios críticos junto con el tráfico de Banda Ancha Móvil. Estos desarrollos subrayan los avances en la consecución del determinismo y la integración efectiva de TSN dentro de los marcos de las normas 3GPP y 5G [31].

Para cumplir con los requisitos de Comunicación Ultra Confiable y de Baja Latencia (URLLC) en la automatización industrial y el Internet de las Cosas (IoT), se destaca la necesidad de un mecanismo de programación conjunta que integre eficientemente TSN con 5G. Este énfasis es integral para la evolución de las redes de comunicación industrial hacia el soporte de una alta calidad de servicio (QoS) y fiabilidad [32].

Otra contribución vital es la introducción de un novedoso mecanismo de priorización para la conformación del tráfico compatible con TSN sobre la Red de Acceso Radio (RAN) de 5G, orientado a mejorar el determinismo y la eficiencia en las comunicaciones industriales. Tal enfoque contextualiza el objetivo más amplio de avanzar en la red de comunicaciones deterministas dentro de las infraestructuras 5G [33].

A pesar de los avances, se reconoce que el campo del determinismo y TSN en 3GPP y 5G aún se encuentra en sus primeras etapas. El enfoque sigue siendo integrar las redes 5G y TSN para permitir el soporte de comunicaciones ultra confiables y de baja latencia, particularmente en aplicaciones de la Industria 4.0, lo que indica esfuerzos continuos de investigación y estandarización [34].

Además, el desarrollo de algoritmos como el de Programación de Políticas Profundas Determinísticas (DDPG) basados en la programación conjunta mejora la capacidad de transporte de múltiples aplicaciones de la arquitectura de colaboración 5G-TSN, marcando otro avance en la integración de capacidades de redes deterministas y sensibles al tiempo dentro de los marcos 5G [35].

En resumen, estos esfuerzos de investigación y marcos propuestos destacan un progreso sustancial en la integración de TSN con las tecnologías 5G, alineándose con los objetivos de las normas 3GPP para apoyar aplicaciones de redes deterministas en diversos escenarios de comunicación industrial y crítica.

2.2.3. IETF DetNet

El grupo de trabajo DetNet[36] del IETF (Internet Engineering Task Force)[37] ha desarrollado una serie de estándares que permiten la implementación de redes deterministas sobre tecnologías de red basadas en IP. El IETF es una organización internacional que desarrolla y promueve estándares voluntarios para la arquitectura y operación de Internet. Fundado en 1986, el IETF es conocido por su proceso de desarrollo colaborativo y abierto, donde ingenieros y desarrolladores de todo el mundo trabajan juntos para crear y refinar protocolos y estándares que aseguren la interoperabilidad y el correcto funcionamiento de Internet. Los estándares del IETF son ampliamente adoptados y utilizados, lo que garantiza una base común para la innovación y el desarrollo tecnológico en el ámbito de las redes y las comunicaciones.

El grupo de trabajo DetNet del IETF se centra en la definición de arquitecturas, mecanismos y protocolos que permitan la creación de redes deterministas sobre una infraestructura de red basada en IP. El propósito principal de DetNet es proporcionar servicios de red con garantías estrictas de latencia, Jitter y pérdida de paquetes, similar a las capacidades ofrecidas por las redes TSN de IEEE, pero extendidas a redes IP. Esto es especialmente relevante para aplicaciones industriales, de automoción, telecomunicaciones y otros sectores que requieren comunicaciones altamente fiables y predecibles.

La solución DetNet abarca varios componentes clave que trabajan juntos para ofrecer redes deterministas. La arquitectura de DetNet (RFC 8655)[38] define la estructura básica y los componentes necesarios para soportar servicios de red deterministas. La arquitectura se divide en dos capas principales: la capa de servicio y la capa de encaminamiento. La capa de servicio se encarga de las funciones de calidad de servicio (QoS), como la reserva de recursos y la gestión del tráfico, mientras que la capa de encaminamiento gestiona el

encaminamiento y la entrega de los paquetes de datos.

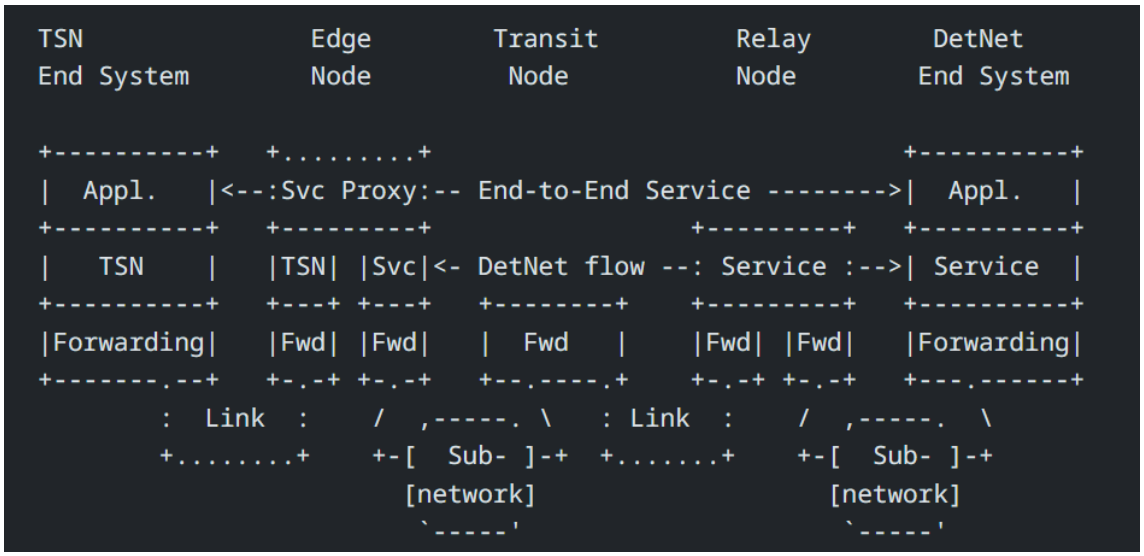


Fig. 2.5. A Simple DetNet-Enabled Network

El plano de datos de DetNet (RFC 8964)[39] especifica cómo se transmiten los datos en una red DetNet. Utiliza un enfoque de túneles MPLS (Multiprotocol Label Switching) o encapsulación sobre IP para asegurar que los paquetes se entreguen de manera predecible y fiable. El plano de datos incluye mecanismos para el etiquetado de flujos y la gestión de rutas específicas que cumplen con los requisitos de latencia y fiabilidad.

Un componente esencial de DetNet es el conjunto de técnicas conocido como Packet Replication, Elimination and Ordering Functions (PREOF) definido en la RFC 9566 [40]. PREOF se utiliza para aumentar la fiabilidad y la predictibilidad de la entrega de paquetes. La replicación de paquetes implica enviar múltiples copias de un paquete por diferentes rutas para asegurar que al menos una copia llegue a su destino, incluso si una ruta falla. La eliminación de duplicados asegura que solo una copia del paquete se procese en el destino, mientras que el ordenamiento garantiza que los paquetes se entreguen en el orden correcto. Estos mecanismos son cruciales para minimizar el Jitter y la pérdida de paquetes, proporcionando un nivel de servicio determinista.

Por último, el control de flujo y la gestión de recursos son también componentes críticos en la solución DetNet. Este estándar especifica los mecanismos para la gestión de recursos y el control de flujo, incluyendo la reserva de ancho de banda y la priorización del tráfico, para asegurar que los flujos deterministas reciban los recursos necesarios para cumplir con sus requisitos de rendimiento.

En resumen, la solución DetNet del IETF proporciona un marco integral para implementar redes deterministas sobre una infraestructura IP. A través de la combinación de mecanismos de QoS, encapsulación de datos, PREOF y gestión de recursos, DetNet permite la creación de redes que pueden cumplir con los estrictos requisitos de latencia, Jitter y fiabilidad necesaria para aplicaciones críticas. Este enfoque extiende las capacidades de

las redes deterministas más allá de los límites de Ethernet, permitiendo su implementación en una amplia variedad de entornos y aplicaciones.

3. MARCO TEÓRICO

3.1. Principios de Redes Deterministas

Las redes deterministas desempeñan un papel crucial en sistemas donde la predictibilidad y la confiabilidad son esenciales. A diferencia de las redes convencionales que se basan en el principio de mejor esfuerzo, las redes deterministas garantizan la entrega de datos dentro de límites de tiempo estrictos, proporcionando así un rendimiento de red previsible. Este tipo de redes son especialmente importantes en aplicaciones críticas donde los retrasos o pérdidas de datos pueden tener consecuencias graves. Es por ello, que en la mayoría de las ocasiones todos o la mayoría de los componentes de dichas infraestructuras estén sincronizados en tiempo, como se puede ver en la figura 3.1. Esta sección explora los principios fundamentales que subyacen a las redes deterministas, su implementación y aplicaciones.

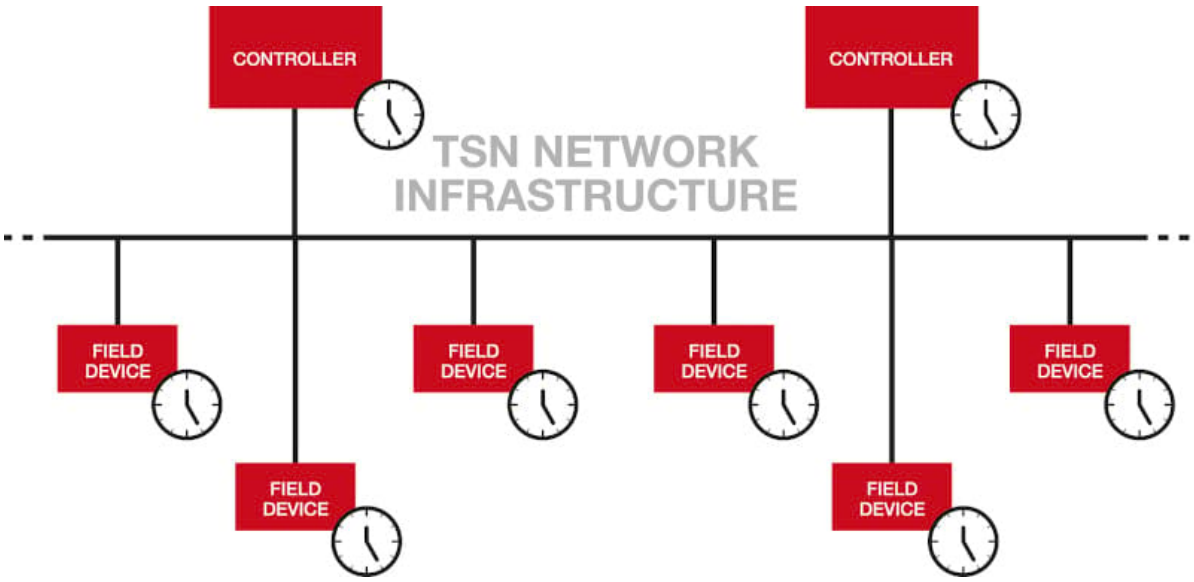


Fig. 3.1. Ejemplo de infraestructura determinista sincronizada en tiempo.

3.1.1. Definición y Características Clave

Las redes deterministas se definen por su capacidad para proveer servicios de comunicación con garantías estrictas sobre el ancho de banda, la latencia, la pérdida de paquetes y el Jitter. Estas garantías son fundamentales en aplicaciones donde el retardo en la transmisión de datos o la incertidumbre en la entrega pueden resultar en fallos críticos o en una degradación significativa del rendimiento [41].

Principales características:

1. **Baja latencia y jitter:** Tiempos de transmisión reducidos y predecibles que son esenciales para procesos críticos en tiempo real. La baja latencia asegura que los datos se transmitan casi instantáneamente, mientras que un Jitter bajo garantiza que las variaciones en los tiempos de transmisión sean mínimas, proporcionando una experiencia consistente.

2. **Sincronización y coordinación:** Capacidad para coordinar acciones en múltiples dispositivos con precisión temporal alta. Esto es crucial en sistemas donde varios componentes deben operar de manera sincronizada, como en la automatización industrial o en vehículos autónomos, donde una desincronización podría resultar en fallos de operación.

3. **Alto rendimiento y confiabilidad:** Minimización de las pérdidas de paquetes y aseguramiento de la transmisión efectiva incluso en condiciones de red adversas. La confiabilidad es esencial para aplicaciones donde la pérdida de datos puede resultar en errores críticos o en la necesidad de retransmisiones costosas.

Las redes deterministas logran estas características a través de la implementación de protocolos y tecnologías específicas que permiten la gestión precisa del tráfico de red y la sincronización temporal entre dispositivos. Estos incluyen mecanismos avanzados de control de tráfico, técnicas de redundancia para asegurar la entrega de datos y protocolos de sincronización temporal.

3.1.2. Diferenciación de Redes Tradicionales

A diferencia de las redes basadas en el principio de "mejor esfuerzo", donde no hay garantías explícitas sobre el rendimiento de la red, las redes deterministas utilizan una serie de técnicas y protocolos especializados para asegurar la entrega de datos en timeframe específicos [42]. Estas técnicas incluyen, pero no están limitadas a:

- **Reserva de recursos:** Asignación previa de recursos de red para flujos de datos específicos, garantizando el ancho de banda necesario. Esto asegura que los datos críticos siempre tengan los recursos necesarios para su transmisión, independientemente de la carga de la red.

- **Planificación de tráfico:** Uso de algoritmos de programación para organizar y priorizar el tráfico de red, asegurando que los datos críticos se transmitan en el momento adecuado. La planificación de tráfico permite que las redes deterministas manejen múltiples flujos de datos con diferentes prioridades sin interferencias.

- **Sincronización de tiempo:** Coordinación temporal precisa entre dispositivos en la red, crucial para mantener la consistencia en la transmisión de datos. La sincronización de tiempo es esencial en aplicaciones donde la precisión temporal es crítica, como en sistemas de control industrial y en redes de sensores.

Estas técnicas permiten que las redes deterministas ofrezcan un nivel de servicio que no es posible con las redes tradicionales, asegurando que los datos críticos lleguen a su destino a tiempo y sin pérdidas. Esta diferenciación es esencial para soportar aplicaciones críticas que dependen de comunicaciones fiables y predecibles.

3.1.3. Estándares y Tecnologías

Los estándares como TSN para Ethernet son fundamentales para habilitar la determinancia en redes modernas. TSN, desarrollado por el IEEE, incluye un conjunto de estándares que permiten la transmisión de datos en tiempo real sobre redes Ethernet (IEEE Std 802.1Q-2014). Estas especificaciones abordan varios aspectos críticos de la comunicación en tiempo real, incluyendo:

- **Sincronización de tiempo:** Estándares como IEEE 802.1AS proporcionan mecanismos para sincronizar los relojes de todos los dispositivos en la red. La sincronización precisa de tiempo es crucial para garantizar que todas las acciones coordinadas en la red ocurran simultáneamente.
- **Calidad de servicio (QoS):** Estándares como IEEE 802.1Qci y IEEE 802.1Qav definen métodos para priorizar y gestionar el tráfico de red. Estos estándares aseguran que los flujos de datos críticos reciban un tratamiento preferencial, manteniendo bajos niveles de latencia y Jitter.
- **Configuración de la red:** Herramientas y protocolos que permiten la configuración dinámica y la gestión de la red para asegurar un rendimiento óptimo. La configuración de la red incluye la capacidad de ajustar dinámicamente los parámetros de la red para adaptarse a las cambiantes condiciones de tráfico.

Estos estándares son esenciales para construir redes que puedan cumplir con los requisitos estrictos de aplicaciones sensibles al tiempo. La adopción de TSN y otros estándares relacionados permite que las redes deterministas soporten una amplia variedad de aplicaciones críticas, proporcionando las garantías necesarias de rendimiento y fiabilidad.

3.1.4. Aplicaciones Críticas

Las aplicaciones de redes deterministas abarcan diversos dominios, incluyendo sistemas de control industrial, comunicaciones vehiculares, aplicaciones militares y más. En estos entornos, la fiabilidad y la puntualidad de la comunicación son imperativas.

- **Sistemas de control industrial:** En la manufactura automatizada, la coordinación precisa entre robots y sensores es esencial para mantener la eficiencia y la seguridad de las operaciones [43]. Los fallos de sincronización pueden resultar en paradas de producción, defectos en los productos o incluso en accidentes.

- **Comunicaciones vehiculares:** Los vehículos autónomos dependen de comunicaciones en tiempo real para coordinar sus movimientos y garantizar la seguridad. La capacidad de transmitir datos de sensores y comandos de control sin retrasos es crucial para evitar colisiones y asegurar una operación segura.
- **Aplicaciones militares:** Los sistemas de defensa y control requieren redes confiables y rápidas para la transmisión de datos críticos en situaciones tácticas. La pérdida o el retraso en la transmisión de datos puede comprometer operaciones y poner en riesgo la seguridad.
- **Atención médica:** En la telemedicina y la cirugía robótica, la transmisión en tiempo real de datos médicos es crucial para el éxito de los procedimientos. Cualquier retraso o pérdida de datos puede afectar negativamente la precisión y la eficacia de los tratamientos.
- **Redes eléctricas inteligentes:** La gestión de redes eléctricas distribuidas depende de comunicaciones precisas y fiables para mantener la estabilidad y la eficiencia del suministro de energía. Las redes deterministas permiten una respuesta rápida a las fluctuaciones de demanda y la integración de fuentes de energía renovable.

La capacidad de proporcionar comunicaciones predecibles y fiables en estos escenarios es crucial para el éxito y la seguridad de las operaciones. Las redes deterministas permiten que estas aplicaciones funcionen de manera óptima, garantizando que los datos críticos se entreguen de manera oportuna y segura.

En conclusión, los principios de las redes deterministas subrayan la importancia de un desempeño de red predecible y confiable, algo que cada vez cobra más relevancia en el mundo interconectado de hoy. A medida que avanzamos hacia una sociedad más digitalizada y automatizada, la implementación de redes deterministas se volverá cada vez más crítica para asegurar la fiabilidad y la eficiencia de nuestros sistemas más importantes. Estas redes no solo soportan las aplicaciones actuales, sino que también habilitan el desarrollo de nuevas tecnologías y servicios que requieren niveles de rendimiento y confiabilidad sin precedentes.

La implementación de redes deterministas permitirá que industrias como la automotriz, la manufactura, la atención médica y muchas otras alcancen nuevos niveles de eficiencia y seguridad. La capacidad de proporcionar comunicaciones en tiempo real con garantías estrictas de rendimiento será un factor clave en el avance de estas tecnologías y en la creación de nuevos paradigmas de conectividad y automatización.

3.2. Tecnologías de Conectividad: Ethernet y 5G

En la actualidad, existen diversas tecnologías de conectividad que permiten la transmisión de datos a través de medios cableados e inalámbricos. Estas tecnologías han evolucionado significativamente, mejorando la velocidad, la eficiencia y la fiabilidad de las

comunicaciones. Aunque la comunicación entre diferentes tecnologías de conectividad está casi resuelta en muchos campos, aún existen desafíos en áreas específicas, como las redes deterministas, que es el enfoque de este trabajo. Esta sección introduce las principales tecnologías de conectividad, tanto cableadas como inalámbricas, y discute sus características y funcionamiento.

3.2.1. Internet Cableado

Las redes cableadas, específicamente las redes Ethernet, son una de las tecnologías de conectividad más comunes y ampliamente utilizadas en todo el mundo. Ethernet, desarrollada inicialmente en la década de 1970, ha evolucionado para soportar velocidades de transmisión de datos extremadamente altas y una gran fiabilidad.

Las redes Ethernet operan mediante el uso de cables físicos para conectar dispositivos dentro de una red local (LAN). La transmisión de datos en una red Ethernet se realiza en forma de tramas, que son paquetes de datos estructurados que incluyen información sobre el origen y el destino de los datos, así como los datos mismos. Los componentes clave de una red Ethernet incluyen:

- **Switches:** Dispositivos que conectan varios segmentos de red y dirigen el tráfico de datos a los dispositivos correctos.
- **Routers:** Dispositivos que conectan diferentes redes y permiten la comunicación entre ellas.
- **Cables:** Cables de cobre (como Cat5e y Cat6) y fibra óptica que transmiten datos entre dispositivos.

Ethernet utiliza un protocolo conocido como CSMA/CD (Carrier Sense Multiple Access with Collision Detection) para gestionar el acceso al medio de transmisión y evitar colisiones de datos. A lo largo de los años, Ethernet ha evolucionado para soportar mayores velocidades de transmisión, desde 10 Mbps en sus inicios hasta 100 Gbps y más en las versiones más recientes.

3.2.2. Internet Móvil

Las redes móviles han revolucionado la forma en que nos comunicamos, permitiendo la transmisión de datos a través de medios inalámbricos. La evolución de las redes móviles desde 3G hasta 5G ha sido marcada por mejoras significativas en la velocidad, la capacidad y la latencia, como se muestra en la figura 3.2.

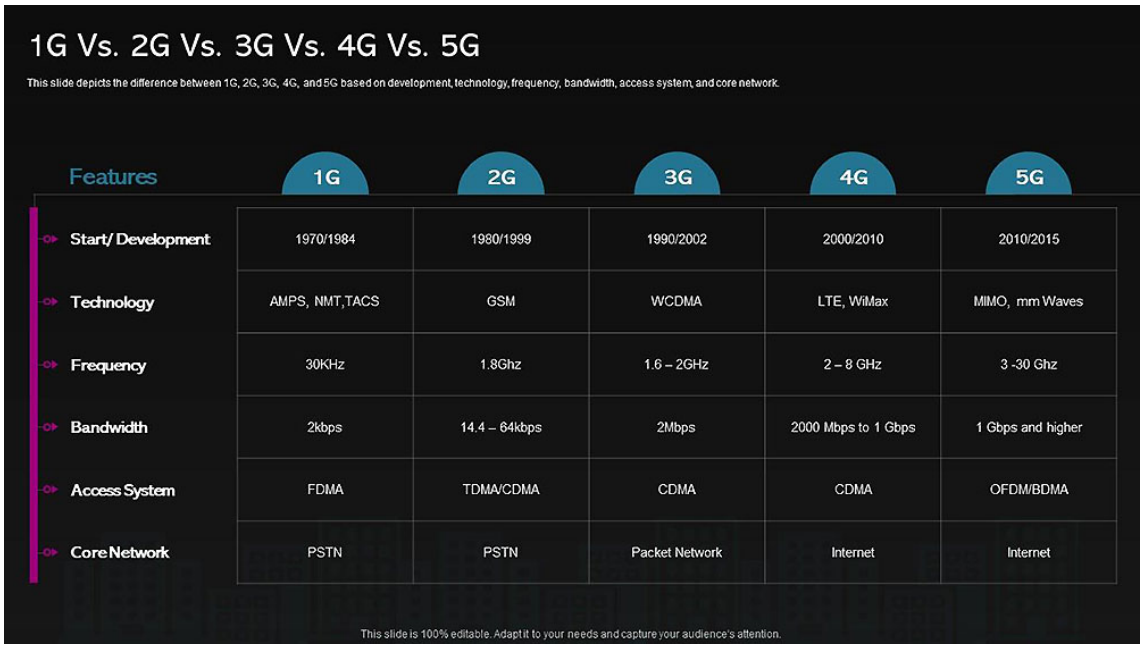


Fig. 3.2. Evolución tecnologías de acceso a internet móvil.

- **3G:** Introdujo la posibilidad de transferir datos a velocidades más altas que las ofrecidas por las redes 2G, soportando aplicaciones como la navegación web y el correo electrónico.
- **4G:** Ofreció velocidades de banda ancha móvil que permitieron la transmisión de video en alta definición y servicios de datos más avanzados.[44]
- **5G:** Promete velocidades de hasta 10 Gbps, latencias extremadamente bajas (menos de 1 ms) y la capacidad de conectar una gran cantidad de dispositivos simultáneamente.[45]

Las redes 5G representan un salto significativo en comparación con las generaciones anteriores. Utilizan técnicas avanzadas como MIMO (Multiple-Input Multiple-Output) y beamforming para mejorar la cobertura y la capacidad de la red. Además, 5G soporta tanto el modo NSA (Non-Standalone) como el modo SA (Standalone).

3.2.2.1. 5G NSA vs SA

El despliegue de redes 5G puede realizarse de dos maneras principales: utilizando el modo NSA (Non-Standalone) o el modo SA (Standalone).

5G NSA (Non-Standalone): En este modo, la red 5G se basa en la infraestructura de red 4G existente. Esto implica que la señal se transmite desde una estación base 4G (eNB) y una estación base 5G (gNB). La ventaja de este enfoque es que requiere menos inversión inicial, ya que aprovecha la infraestructura existente. Sin embargo, la mejora en

la latencia no es significativa, ya que el control de la red sigue dependiendo de la red 4G, resultando en latencias de aproximadamente 15-20 ms.[46]

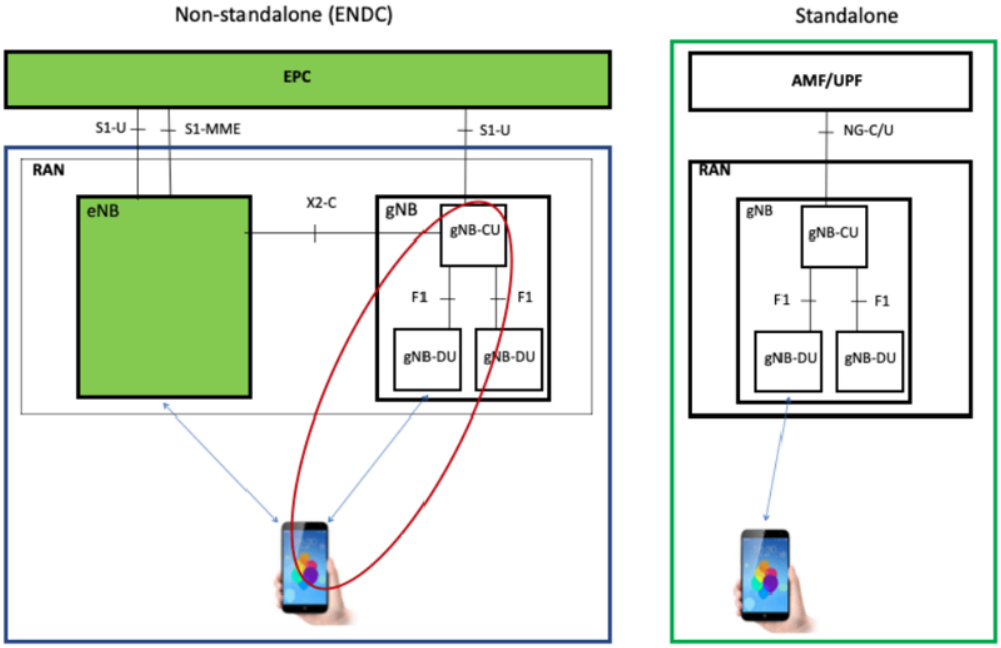


Fig. 3.3. Comparación de las arquitecturas NSA y SA de 5G. Fuente: [47].

5G SA (Standalone): En este modo, la red 5G opera de manera independiente, sin depender de la infraestructura 4G. Esto permite aprovechar todo el potencial de 5G, incluyendo latencias extremadamente bajas (menos de 1 ms) y velocidades de transmisión de hasta 1 Gbps. La implementación de redes 5G SA requiere una inversión mayor, pero proporciona una mejora significativa en el rendimiento y la capacidad de la red. Por estos motivos, en este trabajo nos inclinamos por el despliegue de nuestra red de 5G en esta modalidad, como se verá en consecutivas secciones. re[46]

La capacidad de implementar redes 5G en modos NSA y SA permite una transición flexible y escalonada hacia una infraestructura completamente 5G, maximizando los beneficios a medida que las inversiones se realizan.

A continuación, se discutirá cómo estas tecnologías de conectividad, tanto cableadas como inalámbricas, pueden integrarse para crear redes deterministas que satisfagan los estrictos requisitos de rendimiento de aplicaciones críticas.

4. CARACTERIZACIÓN DE LOS DOMINIOS TECNOLÓGICOS

En este capítulo, se explicará el despliegue, la implementación y la posterior caracterización de cada uno de los dominios TSN que conformarán nuestro sistema multidominio, específicamente los dominios TSN Ethernet y 3GPP. Este análisis es fundamental para entender cómo se integran y funcionan estas tecnologías en conjunto para ofrecer un rendimiento determinista y fiable en aplicaciones críticas.

El capítulo se estructura en dos secciones principales, cada una dedicada a un dominio tecnológico diferente. Primero, abordaremos el dominio Ethernet, que incluye la descripción detallada de los componentes hardware y software utilizados, así como su configuración y despliegue. Posteriormente, se analizará el dominio 3GPP, destacando los aspectos clave de su implementación y las adaptaciones necesarias para integrarse en un entorno multidominio.

4.1. Dominio Ethernet

El dominio Ethernet constituye una parte esencial de nuestro sistema multidominio debido a su capacidad para proporcionar comunicaciones de alta velocidad y baja latencia. En secciones anteriores, ya hemos introducido el concepto de Ethernet y su importancia en las redes modernas. Ahora, nos centraremos en detallar en qué consiste nuestro dominio TSN (Time-Sensitive Networking) en el contexto de Ethernet, así como las bases tecnológicas proporcionadas por el hardware de RELY.

El objetivo principal de esta sección es proporcionar una visión comprensiva de cómo se ha diseñado y desplegado nuestro dominio TSN en Ethernet. Se explorarán los componentes hardware seleccionados, la topología de la red, y las configuraciones específicas que permiten asegurar la fiabilidad y el rendimiento necesario para aplicaciones deterministas. Además, se describirán las herramientas y metodologías empleadas para la optimización y gestión del tráfico en la red, incluyendo mecanismos avanzados como IEEE 802.1Qbv y FRER (Frame Replication and Elimination for Reliability).

A través de esta caracterización detallada, se pretende demostrar cómo se logra una comunicación eficiente y segura en entornos críticos, y cómo el uso de tecnologías avanzadas de TSN puede mejorar significativamente el rendimiento y la fiabilidad de las redes Ethernet en aplicaciones industriales, automotrices, aeroespaciales y ferroviarias. La comprensión de estos aspectos es crucial para el desarrollo de un sistema multidominio robusto y eficaz, capaz de satisfacer las exigentes demandas de las comunicaciones deterministas.

4.1.1. RELY TSN Hardware

En esta sección, se describe el hardware de RELY utilizado en nuestro dominio TSN. Utilizo el RELY-TSN-BRIDGE4, un switch pionero de 4 puertos de RELYUM basado en la avanzada tecnología SoC-e. Este dispositivo está diseñado para sectores críticos como la automatización industrial. El RELY-TSN-BRIDGE4 está equipado con una FPGA integrada que ofrece capacidades de conmutación de red de alta velocidad y estampado de tiempo preciso mediante PTP (Precision Time Protocol). Esta característica es crucial para aplicaciones que requieren una sincronización precisa entre dispositivos, como las industriales y automotrices. Además, la FPGA permite una alta flexibilidad en la implementación de funciones específicas de red, mejorando la capacidad de respuesta y adaptabilidad del sistema. [48]

Communications

- 4x TSN Ethernet port.
- Media options (SFP cages):
  - 10/100/1000Base-T.
  - 1000Base-X.
  - 100Base-FX.
- 1 x 10/100/1000BaseTX Ethernet Service port.
- RSTP IEEE802.1w.
- LLDP support.
- VLAN support.
- IEEE 802.1P Traffic prioritization.
- 1 x USB port.

Processing performance

- On-board UltraScale™ FPGA for high-speed network switching and PTP timestamping.
- Multi-core CPU unit to support autonomous software applications.

TSN features

- IEEE 802.1AS – Timing and Synchronization.
- IEEE 802.1Qbv – Enhancements for Scheduled Traffic.
- IEEE 802.1Qav – Forwarding and Queuing Enhancements for Time-Sensitive Streams.
- IEEE 802.1Qcc – Enhancements for Stream Reservation Protocol.
- IEEE 802.1CB – Frame Replication and Elimination for Reliability.
- IEEE 802.1Qci – Per Stream Ingress Policing.

Rugged devices

- Fanless design and full metal enclosure.
- Power Supply: 9VDC to 30 VDC.
- Operating. temperature: -40°C to +70°C.
- Storage temperature: -40°C to +85°C.
- Optional mounting: DIN rail.

Configuration and Management

- SSH, Netconf support.
- Web-based HTML5-GUI access/configuration:
  - Accessible through HTTP(S).
  - Configuration profiles and Firmware updates.
  - Real-time network monitoring.

Fig. 4.1. Hoja de especificaciones del RELY TSN BRIDGE 4 [48]

La configuración del dispositivo está respaldada por una CPU multinúcleo que aloja aplicaciones de software autónomas, mejorando la capacidad de respuesta general del sistema y la flexibilidad funcional. Esta CPU multinúcleo permite la ejecución de múltiples tareas simultáneamente, optimizando el rendimiento del switch y asegurando que se puedan gestionar múltiples flujos de datos sin comprometer la calidad del servicio.

Para optimizar el rendimiento de la red, se ha adquirido una herramienta de configuración especializada, llamada RTAW PEGASE. Esta herramienta permite gestionar el IEEE 802.1Qbv, así como otros mecanismos singulares de TSN (IEEE 802.1Q, IEEE 802.1Qav, IEEE 802.1Qbu, IEEE 802.1AS) soportados por el TSN4 como se muestra en la Figura 4.1. La herramienta es capaz de calcular el Worst Case Scenario para el tráfico de la red y seleccionar automáticamente la configuración óptima basada en uno de los mecanismos TSN seleccionados, para posteriormente generar los ficheros de configuración que han de ser flasheados en cada uno de los dispositivos a través de su interfaz web. Este nivel de automatización y optimización es esencial para mantener un rendimiento de red consis-

tente, especialmente en entornos donde se manejan tanto tráfico determinista como tráfico de mejor esfuerzo (BE). [49]

Para este propósito, he seleccionado cuatro switches TSN de RELY, dispuestos en una topología de rombo, lo que nos permite habilitar tanto IEEE 802.1Qbv (Time-Aware Shaper) como FRER (Frame Replication and Elimination for Reliability) en el dominio TSN, algoritmos que considero clave para el caso de uso que presentamos, donde es de suma importancia mostrar como el tráfico crítico no se ve afectado por ningún agente externo u otro tráfico, incluso compartiendo el mismo canal. El diseño de red elegido maneja de manera efectiva ambos tipos de tráfico, asegurando que se cumplan los requisitos de latencia y fiabilidad. La capacidad de manejar tráfico determinista es crucial para aplicaciones críticas que no pueden tolerar demoras o pérdida de datos, mientras que el tráfico de mejor esfuerzo se gestiona de manera que no interfiera con los requisitos de alta prioridad del tráfico determinista.

Fig. 4.2. Topología de los switches TSN en el dominio Ethernet.

4.1.1.1. Configuración y despliegue

La configuración y despliegue de los switches TSN en nuestro dominio Ethernet siguen un proceso meticuloso para asegurar que todos los componentes funcionen en armonía y que se cumplan los estrictos requisitos de la red determinista. A continuación, se detallan los pasos clave de esta configuración:

1. **Instalación Física:** Los switches TSN se instalan en una topología de rombo para optimizar la redundancia y la fiabilidad como se comentó en la sección previa. Esta

disposición permite la implementación efectiva de los mecanismos de Qbv y FRER. La topología de rombo es particularmente eficaz para proporcionar múltiples rutas de comunicación, lo que incrementa la resiliencia de la red frente a fallos de hardware y garantiza que el tráfico pueda ser redirigido sin interrupciones significativas en caso de una falla.

2. **Configuración de Red:** Utilizando la herramienta de RTAW PEGASE, se ajustan los parámetros de Qbv, incluyendo el tiempo de ciclo, tiempo de transmisión, las clases de tráfico, el mapeado de estas a las colas, el número de colas y la caracterización de cada uno de los tipos de tráfico. También se configuran los mecanismos adicionales de TSN.Todo esto se muestra en las figuras 4.3 y 4.4. La herramienta de configuración proporciona una interfaz intuitiva para definir y ajustar los parámetros de la red, permitiendo una gestión precisa y eficiente del tráfico de datos.

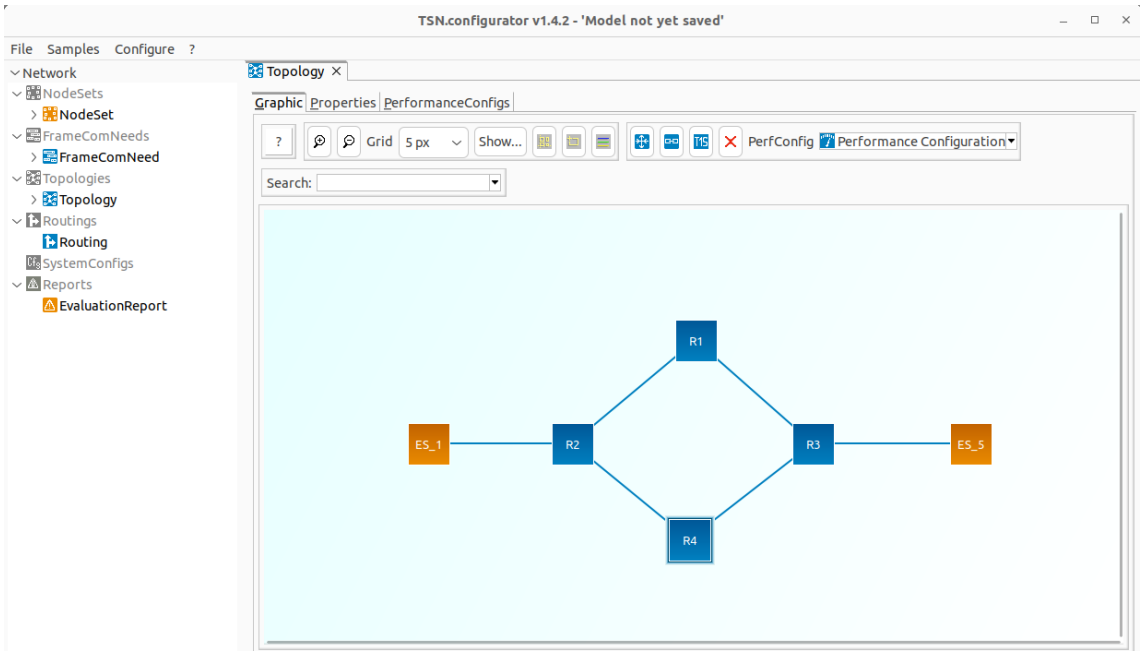


Fig. 4.3. PEGASE Configurator. Topologia y clases de tráfico.

3. **Optimización y Pruebas:** Como se comenta posteriormente en este trabajo, en la sección 4.1.2, al ser un dominio comercial, la caracterización no es crítica, pero ello no implica que no se necesite testear tanto la configuración como los dispositivos utilizados. Por ello se ejecutan pruebas para verificar dicha configuración y se utilizan herramientas de análisis (iperf3, tcpdump, TSN-LAB) para identificar posibles cuellos de botella y optimizar el rendimiento. La herramienta de configuración se utiliza para calcular escenarios de peor caso y ajustar la configuración en consecuencia. Los valores de estos escenarios son contrastados con los datos reales obtenidos en busca de posibles discrepancias. Durante esta fase, también se realizan pruebas de estrés y simulaciones de tráfico para asegurar que la red puede manejar las cargas de trabajo esperadas sin degradación del rendimiento. Las mediciones realizadas

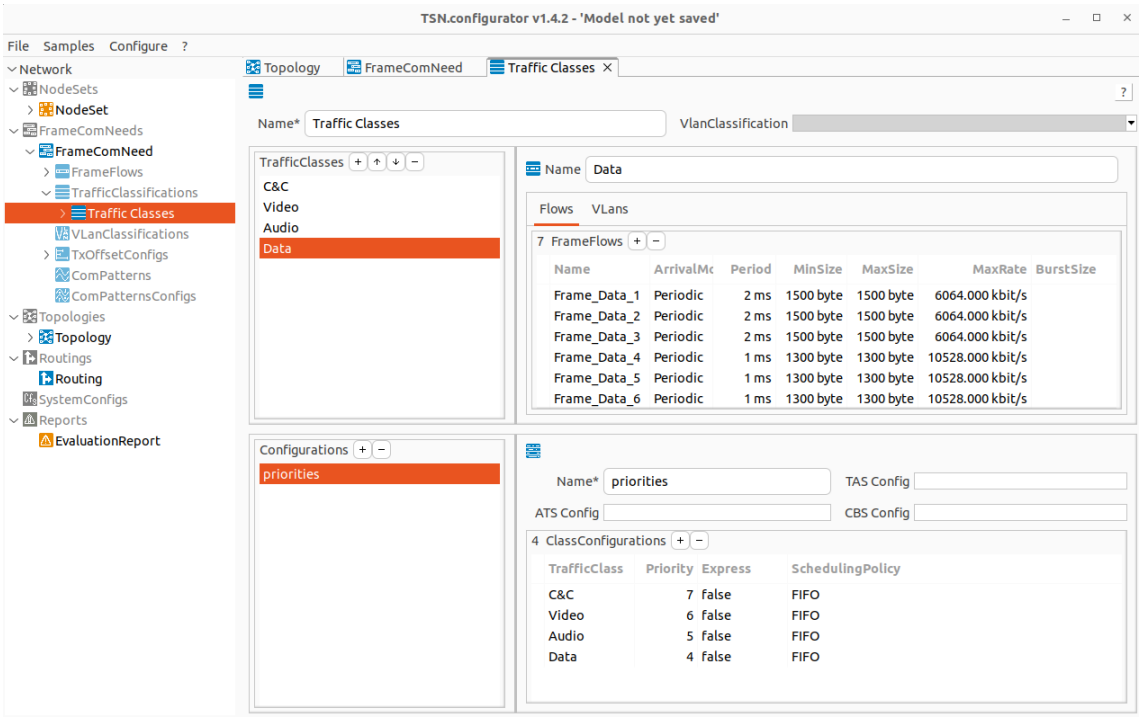


Fig. 4.4. PEGASE Configurator. Clases de trafico definidas.

4. **Monitoreo Continuo:** Una vez desplegada, la red se monitorea continuamente para asegurar que se mantienen los niveles de servicio esperados. Se emplean mecanismos de OAM (Operaciones, Administración y Mantenimiento) para detectar y resolver proactivamente cualquier problema. El monitoreo continuo incluye la supervisión de métricas clave como la latencia, el jitter y la pérdida de paquetes, permitiendo una respuesta rápida a cualquier anomalía que pueda surgir. Para la obtención de estas medidas en tiempo real, se utiliza la API interna de los RELYUM en conjunción con el dispositivo RELY TSN LAB 4.6, el cual funciona a modo de probe y permite hacer hardware timestamping de paquetes, lo que hace que nuestras métricas tengan el nivel de precisión razonable en este caso de uso.

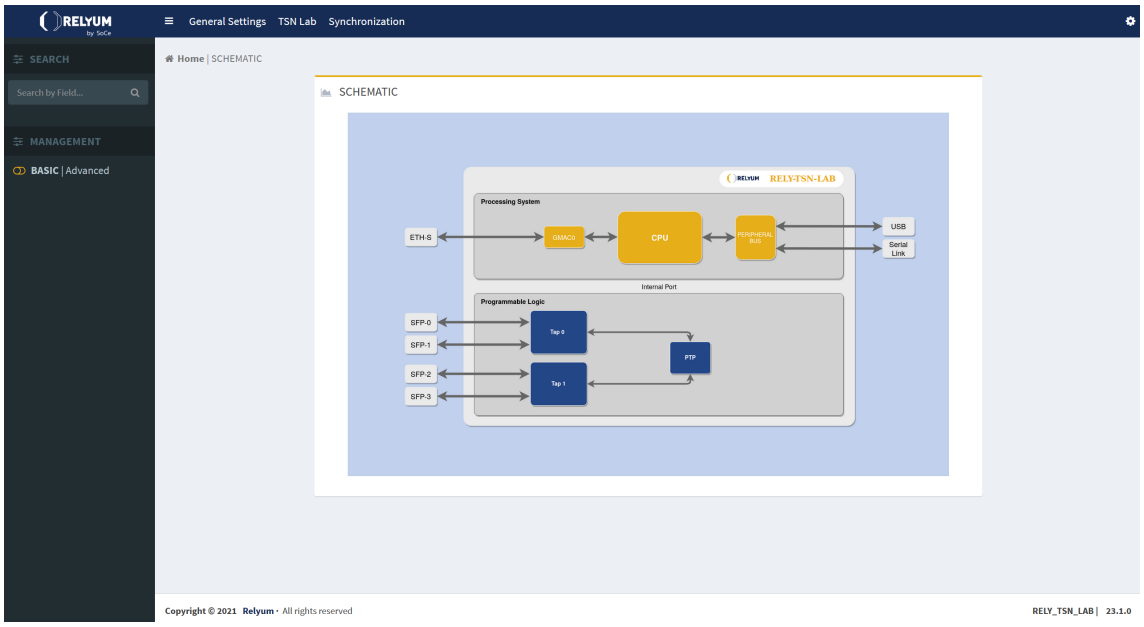


Fig. 4.5. RELY TSN LAB Interface.

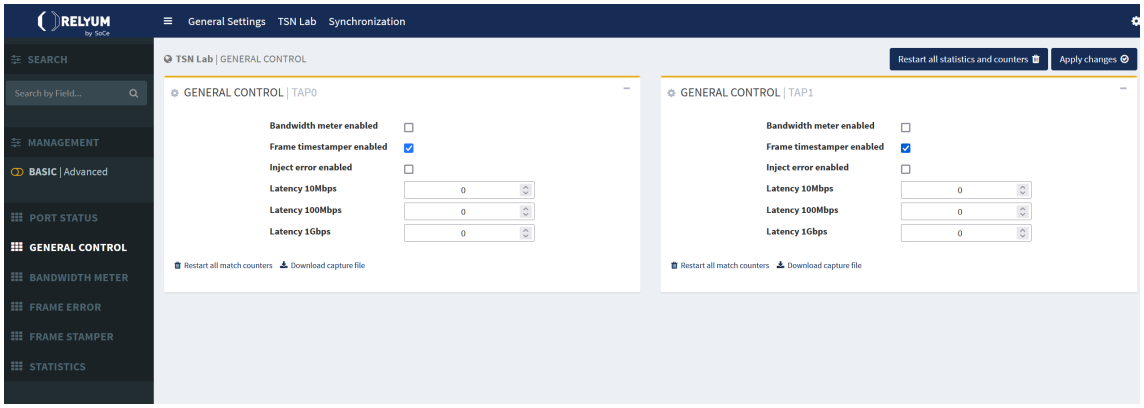


Fig. 4.6. RELY TSN LAB Capture Options.

Esta configuración garantiza que nuestro dominio TSN en Ethernet esté preparado para manejar aplicaciones críticas con altos requisitos de fiabilidad y baja latencia. Al combinar hardware avanzado con herramientas de configuración y monitoreo sofisticadas, podemos asegurar que nuestra red TSN cumple con los más altos estándares de rendimiento y fiabilidad.

4.1.2. Caracterización de la Red Ethernet TSN

Dado que nuestro dominio Ethernet TSN está compuesto por dispositivos comerciales avanzados y específicamente diseñados para proporcionar comunicaciones deterministas, la caracterización detallada de la red en términos de delay, jitter y otros parámetros clave no es estrictamente necesaria ni aporta valor a este trabajo de investigación. Esto se debe

a que los valores de rendimiento de estos dispositivos son conocidos y han sido validados exhaustivamente por el fabricante.

Los switches RELY-TSN-BRIDGE4 utilizados en nuestra red están diseñados para cumplir con los estándares más exigentes de la industria, proporcionando garantías de tiempo preciso y una fiabilidad excepcional. Adicionalmente, nuestra herramienta de configuración, que gestiona los mecanismos TSN, proporciona informes detallados de rendimiento en tiempo real. Esta herramienta es capaz de medir y reportar valores de delay, jitter y pérdida de paquetes directamente desde los switches. Estas métricas son calculadas automáticamente y permiten una monitorización continua de la red, asegurando que se mantengan los niveles de rendimiento esperados.

En lugar de realizar una caracterización manual de la red, confiamos en los datos proporcionados por la herramienta de configuración y los datasets validados por el fabricante. Esta metodología no solo ahorra tiempo y recursos, sino que también proporciona una mayor precisión en la monitorización del rendimiento de la red.

En resumen, la utilización de hardware TSN comercial, combinado con herramientas avanzadas de configuración y monitorización, permite asegurar que nuestra red Ethernet TSN cumple con los requisitos de rendimiento y fiabilidad necesarios para aplicaciones críticas, sin la necesidad de caracterizaciones manuales adicionales.

4.2. Dominio 3GPP

En esta sección, se presentará nuestro dominio 3GPP, que consistirá en un núcleo 5G de código abierto y un equipo de usuario (UE) comercial. Este despliegue está diseñado para proporcionar una plataforma robusta y flexible para redes deterministas, aprovechando las capacidades avanzadas del 5G SA (Standalone) para cumplir con los estrictos requisitos de rendimiento y fiabilidad.

Cabe señalar que per sé la implementación que se despliega, a diferencia de en el anterior dominio, no contiene elemnetos deterministas de manera nativa, por lo cual he tenido que idear e implementar una solucion que me permita habilitar estas características.

4.2.1. Despliegue de Open Air Interface 5G SA

OpenAirInterface (OAI) es una iniciativa de código abierto que proporciona implementaciones de software para redes de comunicación 4G y 5G. Este proyecto es mantenido por la OpenAirInterface Software Alliance (OSA), que se dedica a la investigación y el desarrollo de tecnologías de comunicación inalámbrica. OAI permite a los investigadores y desarrolladores desplegar y experimentar con redes 5G utilizando una infraestructura de software completamente abierta. [50]

He elegido desplegar una red 5G SA (Standalone) utilizando OAI debido a las ven-

tajas que ofrece en términos de rendimiento determinista. A diferencia de las redes 5G NSA (Non-Standalone), que dependen de una infraestructura 4G existente, las redes 5G SA operan de manera independiente, lo que permite una menor latencia y una mayor fiabilidad. Estas características son esenciales para aplicaciones deterministas donde los retrasos y la pérdida de paquetes pueden tener consecuencias críticas.

El despliegue de una red 5G SA utilizando OAI permite aprovechar al máximo las capacidades del 5G, incluyendo la ultra baja latencia y la alta fiabilidad. Estas capacidades son fundamentales para cumplir con los requisitos de las aplicaciones deterministas, haciendo de OAI la opción ideal para nuestro dominio 3GPP. A continuación, se detallará la implementación específica de OpenAirInterface y sus diferencias con otras soluciones.

4.2.1.1. Open RAN & Open Air Interface

Open RAN (Radio Access Network) es un concepto que promueve la apertura y estandarización de las interfaces en las redes de acceso de radio, permitiendo la interoperabilidad entre equipos de diferentes proveedores. Este enfoque permite una mayor flexibilidad y eficiencia en el despliegue de redes de comunicación.

OpenAirInterface (OAI) es una implementación específica dentro del ecosistema Open RAN que ofrece una pila completa de 4G y 5G, incluyendo tanto el núcleo de red (Core Network) como la red de acceso de radio (RAN). OAI se destaca por su flexibilidad y capacidad de personalización, lo que lo convierte en una herramienta valiosa para investigadores y desarrolladores que buscan experimentar con nuevas tecnologías y optimizaciones de red.

Las principales características de OAI incluyen:

- **Flexibilidad y personalización:** OAI permite modificar y personalizar las implementaciones de red para adaptarse a necesidades específicas, lo que es especialmente útil en entornos de investigación y desarrollo.
- **Interoperabilidad:** Gracias a su adherencia a los estándares Open RAN, OAI puede inter operar con equipos de diferentes proveedores, facilitando la integración de diversas tecnologías.
- **Implementación completa:** OAI proporciona una pila completa de 4G y 5G, incluyendo funciones del núcleo de red (como MME, HSS, SPGW en 4G y AMF, SMF, UPF en 5G) y de la red de acceso de radio.
- **Soporte para 5G SA:** OAI soporta implementaciones de 5G SA, lo que permite desplegar redes independientes de 5G con capacidades avanzadas de baja latencia y alta fiabilidad.

Las diferencias clave entre OAI y otras soluciones comerciales incluyen su naturaleza de código abierto, que permite un acceso completo al código fuente y la posibilidad de

realizar modificaciones profundas. Además, OAI está diseñado para ser altamente modular, permitiendo a los desarrolladores habilitar o deshabilitar componentes específicos según las necesidades del proyecto.

En comparación con soluciones comerciales, OAI ofrece una plataforma accesible y flexible para la experimentación y el desarrollo, aunque puede requerir más esfuerzo en términos de configuración y mantenimiento debido a su naturaleza abierta. Sin embargo, para propósitos de investigación y despliegues personalizados, OAI proporciona una base sólida y extensible para explorar las capacidades del 5G SA y su aplicación en redes deterministas.

4.2.1.2. Configuración y despliegue

Para el despliegue de nuestra red 5G con Open Air Interface (OAI), hemos utilizado una configuración específica de hardware y software para asegurar la funcionalidad y rendimiento deseados. A continuación, se detallan los componentes y el proceso de configuración.

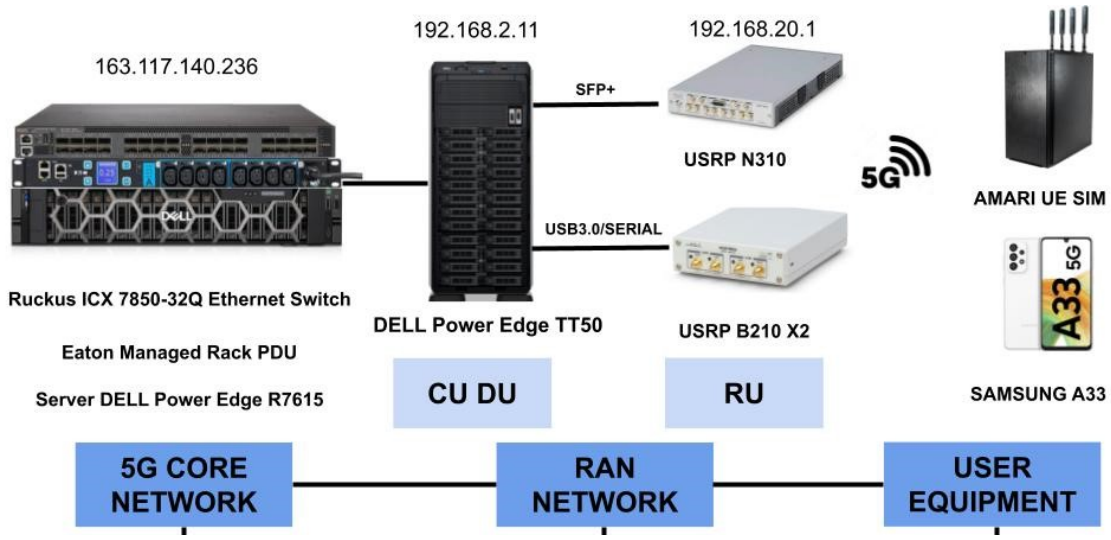


Fig. 4.7. Diagrama de la red desplegada.

En la Figura 4.7, se puede observar la configuración de nuestra red 5G. Los componentes principales son:

- **Servidor DELL Power Edge R7615:** Este servidor alberga la core network de 5G. Se encarga de gestionar todos los aspectos del core, incluyendo AMF, SMF, UPF, y otros componentes críticos.
- **Switch Ethernet Ruckus ICX 7850-32Q:** Este switch gestiona la conectividad entre los diferentes componentes de la red.

- **USRP N310 y B210 X2:** Estos dispositivos se utilizan como radios definidas por software (SDR) para la transmisión y recepción de señales de 5G.
- **Teléfono Samsung Galaxy A33 5G:** Utilizado como equipo de usuario (UE) para conectarse a la red 5G desplegada.
- **DELL Power Edge TT50:** Este servidor gestiona las funciones de la RAN, actuando como CU y DU.

A continuación, se muestra el proceso de configuración y despliegue de la red 5G:

1. Despliegue de contenedores Docker: El primer paso en la implementación de nuestra red 5G es el despliegue de los contenedores Docker que alojan los componentes del core de Open Air Interface (OAI). Docker es una plataforma que permite la creación y gestión de contenedores ligeros, que son paquetes de software que incluyen todas las dependencias necesarias para ejecutar una aplicación de manera consistente en cualquier entorno.

Configuración del Entorno Docker Antes de iniciar los contenedores, es fundamental configurar el entorno Docker adecuadamente. Esto incluye la instalación de Docker en el servidor DELL Power Edge R7615 y la preparación de las imágenes de Docker necesarias para los componentes de OAI. Las imágenes de Docker contienen el software y todas sus dependencias empaquetadas, lo que facilita su despliegue y actualización.

```
# Instalaci n de Docker
sudo apt-get update
sudo apt-get install -y docker.io

# Verificaci n de la instalaci n
docker --version
```

CÓDIGO 4.1. Instalación de Docker

Preparación de las Imágenes de Docker Open Air Interface proporciona imágenes Docker pre compiladas para sus componentes, lo que simplifica significativamente el proceso de despliegue. Estas imágenes deben ser descargadas y preparadas antes de iniciar los contenedores. [51]

```
# Descargar im genes de Docker de OAI
docker pull oai-nrf
docker pull oai-amf
docker pull oai-smf
docker pull oai-spgwu
docker pull oai-udr
docker pull oai-udm
```

```
docker pull oai-ausf
docker pull oai-nssf
```

CÓDIGO 4.2. Descargar imágenes de Docker de OAI

Despliegue de los Contenedores Una vez que las imágenes están preparadas, el siguiente paso es crear y desplegar los contenedores. Utilizamos Docker Compose, una herramienta que permite definir y ejecutar aplicaciones Docker multi-contenedor. Docker Compose utiliza un archivo YAML para configurar los servicios de la aplicación.

El archivo `docker-compose.yaml` define todos los servicios necesarios para el core de OAI, incluyendo el NRF (Network Repository Function), AMF (Access and Mobility Management Function), SMF (Session Management Function), SPGWU (Serving Gateway User Plane Function), entre otros.

```
version: '3.7'
services:
  oai-nrf:
    image: oai-nrf
    container_name: oai-nrf
    networks:
      oai-public-net:
        ipv4_address: 192.168.70.135

  oai-amf:
    image: oai-amf
    container_name: oai-amf
    depends_on:
      - oai-nrf
    networks:
      oai-public-net:
        ipv4_address: 192.168.70.136
    environment:
      - NRF_IPV4_ADDRESS=192.168.70.135

# de manera similar, todo el resto de dockers...
```

CÓDIGO 4.3. Configuración de Docker Compose para OAI

Para iniciar los contenedores, ejecutamos el siguiente comando en la terminal:

```
docker-compose -f docker-compose.yaml up -d
```

CÓDIGO 4.4. Despliegue de contenedores con Docker Compose

Este comando despliega todos los servicios definidos en el archivo `docker-compose.yaml` en modo desatendido (`-d`), permitiendo que los contenedores se ejecuten en segundo plano.

Verificación del Despliegue Una vez que los contenedores están en ejecución, es crucial verificar que todos los servicios se han iniciado correctamente. Utilizamos el comando `docker ps` para listar todos los contenedores activos y asegurarnos de que no haya errores en los logs de inicio.

```
# Listar contenedores activos
docker ps

# Verificar logs de un contenedor espec fico
docker logs oai-amf
```

CÓDIGO 4.5. Verificación del estado de los contenedores

En la Figura 4.8, se muestra la salida de la consola al ejecutar el comando `docker-compose`, donde se observa que todos los contenedores se han iniciado correctamente.

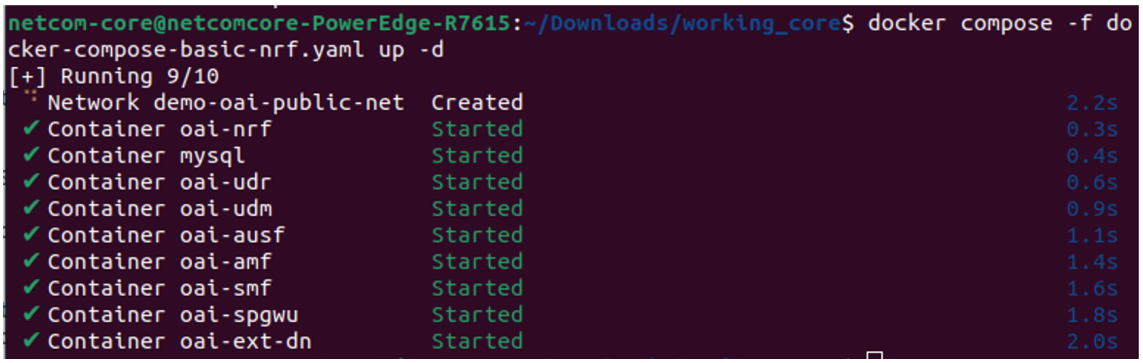


Fig. 4.8. Despliegue de contenedores Docker para OAI.

Este paso es fundamental para asegurar que los componentes del core de OAI están funcionando correctamente y están listos para gestionar las funciones de la red 5G. La utilización de contenedores Docker no solo simplifica el proceso de despliegue, sino que también proporciona una mayor flexibilidad y portabilidad, permitiendo que los mismos contenedores se ejecuten en diferentes entornos con mínima configuración adicional.

2. Configuración del AMF: Una vez desplegados los contenedores de Docker, el siguiente paso es la inicialización del Access and Mobility Management Function (AMF) y la conexión del gNB (gNodeB) al core de la red 5G. El AMF es responsable de la gestión de la movilidad y la conexión del usuario, mientras que el gNB actúa como el punto de acceso de radio para los dispositivos de usuario (UE).

Inicialización del AMF Para iniciar el AMF, se debe asegurar que el contenedor correspondiente esté en funcionamiento y que su configuración sea correcta. La configuración del AMF incluye parámetros esenciales como la dirección IP del NRF (Network Repository Function), que permite al AMF registrar sus servicios y descubrir otros servicios en la red.

El siguiente comando inicia el AMF (aunque en nuestro caso no es necesario, ya que se levanta cuando hacemos el paso previo con docker compose up) y muestra su proceso de inicialización:

```
docker-compose -f docker-compose.yaml up -d oai-amf
```

CÓDIGO 4.6. Inicialización del AMF

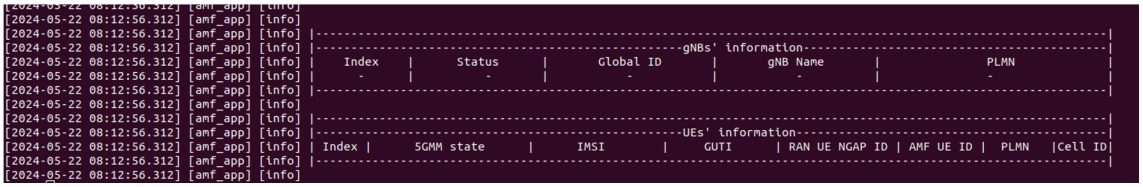


Fig. 4.9. Logs de inicialización del AMF.

En la Figura 4.9, se puede observar la salida de los logs del AMF durante su inicialización. Estos logs indican que el AMF se ha registrado correctamente en el NRF y está listo para gestionar la movilidad y las conexiones de los usuarios.

Conexión del gNB al AMF Una vez que el AMF está operativo, el siguiente paso es conectar el gNodeB (gNB) a la red. El gNB es el nodo que proporciona la interfaz de radio para los dispositivos de usuario y se conecta al core de la red a través del AMF.

Para configurar y conectar el gNB, se deben seguir los siguientes pasos:

- **Configuración del gNB:** La configuración del gNB incluye parámetros como la dirección IP del AMF y otros detalles de la red. Esta configuración se realiza en un archivo de configuración específico para el gNB.

```
nano /etc/gnb/config.conf
# Aqui se configuran los parametros previamente mencionados para
    el correcto funcionamiento del gnb
```

CÓDIGO 4.7. Configuración del gNB

- **Inicio del gNB:** Una vez configurado, el gNB se inicia con el siguiente comando. Este comando ejecuta el software del gNB, que luego intenta conectarse al AMF utilizando los parámetros de configuración proporcionados.

```
sudo ./nr-softmodem -O /etc/gnb/config.conf
```

CÓDIGO 4.8. Inicio del gNB

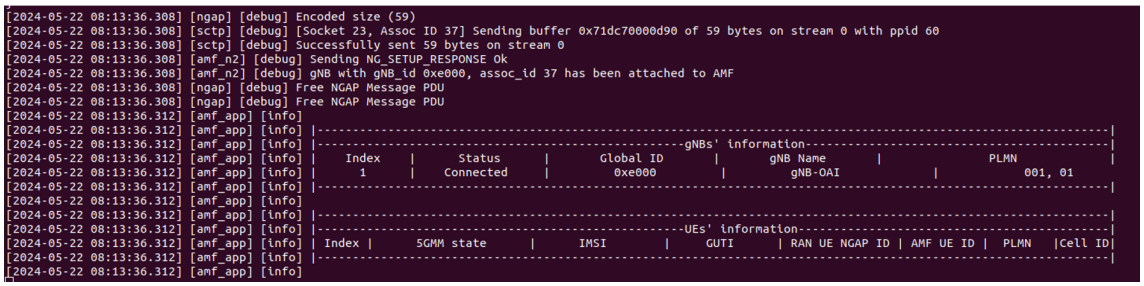


Fig. 4.10. Logs de conexión del gNB al AMF.

En la Figura 4.10, se muestran los logs que indican el proceso de conexión del gNB al AMF. Estos logs confirman que el gNB se ha conectado correctamente al AMF y que la comunicación entre ambos componentes está funcionando como se espera.

Verificación de la Conexión Después de la inicialización del AMF y la conexión del gNB, es crucial verificar que la conexión se haya establecido correctamente y que el gNB esté registrado en el AMF. Utilizamos los siguientes comandos para revisar el estado de la conexión:

```
docker logs oai-amf
docker exec -it oai-amf /bin/bash
# Dentro del contenedor, podemus utilizar vim para buscar la cadena
    de texto del registro, por ejemplo
```

CÓDIGO 4.9. Verificación de la conexión del gNB

Estos pasos aseguran que el AMF y el gNB estén configurados y operativos, permitiendo que la red 5G gestione las conexiones de los dispositivos de usuario de manera eficiente y confiable. La correcta configuración e inicialización de estos componentes es fundamental para el funcionamiento de la red 5G y para garantizar una conectividad continua y de alta calidad para los usuarios.

3. Configuración del USRP: En este paso, se detallará el proceso de configuración y ejecución del USRP (Universal Software Radio Peripheral) para el dominio 5G. El USRP actúa como el frontend de radio para la red 5G, permitiendo la transmisión y recepción de señales de radiofrecuencia. Para nuestro despliegue, utilizamos el modelo USRP N310, que es adecuado para aplicaciones de redes 5G debido a su capacidad y flexibilidad. [52]

Configuración del USRP La configuración del USRP implica la especificación de parámetros esenciales como la frecuencia de operación, el ancho de banda, la potencia de transmisión y las configuraciones de antena. A continuación, se muestra la configuración utilizada:

```
Active_gNBs = ( "gNB-OAI" );
Asn1_verbosity = "none";
```

```
3
4  gNBs = (
5  {
6  // Identification parameters:
7  gNB_ID = 0xe00;
8  gNB_name = "gNB-OAI";
9
10 makefile
11
12 // Tracking area code
13 tracking_area_code  =  1;
14 plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({
       sst = 1}) });
15
16 nr_cellid = 12345678L;
17
18 // Physical parameters:
19 do_CSIRS = 1;
20 do_SRS = 1;
21
22 pdcch_ConfigSIB1 = (
23 {
24     controlResourceSetZero = 12;
25     searchSpaceZero = 0;
26 }
27 );
28
29 servingCellConfigCommon = (
30 {
31     physCellId = 0;
32
33     // Downlink configuration
34     absoluteFrequencySSB = 641280;
35     dl_frequencyBand = 78;
36     dl_absoluteFrequencyPointA = 640008;
37     dl_subcarrierSpacing = 1;
38     dl_carrierBandwidth = 106;
39
40     initialDLBWPlocationAndBandwidth = 28875;
41     initialDLBWPsubcarrierSpacing = 1;
42     initialDLBWPcontrolResourceSetZero = 12;
43     initialDLBWPsearchSpaceZero = 0;
44
45     // Uplink configuration
46     ul_frequencyBand = 78;
47     ul_subcarrierSpacing = 1;
48     ul_carrierBandwidth = 106;
49     pMax = 20;
50     initialULBWPlocationAndBandwidth = 28875;
51     initialULBWPsubcarrierSpacing = 1;
52
```

```
53      // RACH configuration
54      prach_ConfigurationIndex = 98;
55      prach_msg1_FDM = 0;
56      prach_msg1_FrequencyStart = 0;
57      zeroCorrelationZoneConfig = 13;
58      preambleReceivedTargetPower = -96;
59      preambleTransMax = 6;
60      powerRampingStep = 1;
61      ra_ResponseWindow = 4;
62      ssb_perRACH_OccasionAndCB_PreamblesPerSSB_PR = 4;
63      ssb_perRACH_OccasionAndCB_PreamblesPerSSB = 14;
64      ra_ContentionResolutionTimer = 7;
65      rsrp_ThresholdSSB = 19;
66      prach_RootSequenceIndex_PR = 2;
67      prach_RootSequenceIndex = 1;
68      msg1_SubcarrierSpacing = 1;
69      restrictedSetConfig = 0;
70      msg3_DeltaPreamble = 1;
71      p0_NominalWithGrant = -90;
72
73      pucchGroupHopping = 0;
74      hoppingId = 40;
75      p0_nominal = -90;
76      ssb_PositionsInBurst_PR = 2;
77      ssb_PositionsInBurst_Bitmap = 1;
78      ssb_periodicityServingCell = 2;
79      dmrs_TypeA_Position = 0;
80      subcarrierSpacing = 1;
81
82      referenceSubcarrierSpacing = 1;
83      dl_UL_TransmissionPeriodicity = 6;
84      nrofDownlinkSlots = 7;
85      nrofDownlinkSymbols = 6;
86      nrofUplinkSlots = 2;
87      nrofUplinkSymbols = 4;
88      ssPBCH_BlockPower = -25;
89  }
90  );
91
92  // SCTP definitions
93  SCTP = {
94      SCTP_INSTREAMS  = 2;
95      SCTP_OUTSTREAMS = 2;
96  };
97
98  // AMF parameters
99  amf_ip_address  = ( { ipv4 = "192.168.80.132"; ipv6 =
        "192:168:30::17"; active = "yes"; preference = "ipv4"; } );
100
101 NETWORK_INTERFACES = {
102     GNB_INTERFACE_NAME_FOR_NG_AMF = "demo-oai";
```

```
    GNB_IPV4_ADDRESS_FOR_NG_AMF = "192.168.80.129/24";
    GNB_INTERFACE_NAME_FOR_NGU = "demo-oai";
    GNB_IPV4_ADDRESS_FOR_NGU = "192.168.80.129/24";
    GNB_PORT_FOR_S1U = 2152;
};

}
);

MACRLCs = (
{
num_cc = 1;
tr_s_preference = "local_L1";
tr_n_preference = "local_RRC";
pusch_TargetSNRx10 = 250;
pucch_TargetSNRx10 = 250;
ulsch_max_frame_inactivity = 0;
}
);

L1s = (
{
num_cc = 1;
tr_n_preference = "local_mac";
prach_dtx_threshold = 120;
pucch0_dtx_threshold = 100;
ofdm_offset_divisor = 8;
}
);

RUs = (
{
local_rf = "yes";
nb_tx = 1;
nb_rx = 1;
att_tx = 6;
att_rx = 6;
bands = [78];
max_pdschReferenceSignalPower = -27;
max_rxgain = 114;
eNB_instances = [0];
bf_weights = [0x00007fff, 0x0000, 0x0000, 0x0000];
clock_src = "internal";
}
);

THREAD_STRUCT = (
{
parallel_config = "PARALLEL_SINGLE_THREAD";
worker_config = "WORKER_ENABLE";
}
```

```
);

rfsimulator = {
serveraddr = "server";
serverport = "4043";
options = ();
modelname = "AWGN";
IQfile = "/tmp/rfsimulator.iqs";
};

security = {
ciphering_algorithms = ( "nea0" );
integrity_algorithms = ( "nia2", "nia0" );
drb_ciphering = "yes";
drb_integrity = "no";
};

log_config = {
global_log_level = "info";
hw_log_level = "info";
phy_log_level = "info";
mac_log_level = "info";
rlc_log_level = "info";
pdcp_log_level = "info";
rrc_log_level = "info";
ngap_log_level = "debug";
f1ap_log_level = "debug";
};
```

CÓDIGO 4.10. Configuración del USRP N310

Los parámetros destacados incluyen:

- **gNB_ID**: Identificador único del gNodeB.
- **absoluteFrequencySSB y dl_absoluteFrequencyPointA:** Configuración de frecuencias de enlace descendente.
- **prach_ConfigurationIndex:** Índice de configuración del canal de acceso aleatorio.
- **amf_ip_address:** Dirección IP del AMF al que se conectará el gNB.

Ejecución del USRP Una vez configurado el archivo, el siguiente paso es ejecutar el USRP con la configuración especificada. Esto se realiza utilizando el siguiente comando:

```
netcom-core@netcomcore-PowerEdge-R7615:~/openairinterface5g/cmake_targets/ran_build/build$ sudo ./nr-softmodem -O ../../../targets/PROJECTS/GENERIC-NR-5GC/CONF2/mcs/gnb.sa.band7
8.fr1.106PRB.usrpb210.12.conf --sa -E
```

Fig. 4.11. Ejecución de la configuración del USRP N310.

Este comando inicia el software del USRP, que luego se conecta al gNB y se registra en el core, como se muestra en la figura 4.12. Finalmente, comienza a transmitir y recibir señales de radio. La figura 4.13 muestra la salida de la consola durante la ejecución del USRP, indicando que el dispositivo está operativo y en comunicación con el gNB.

Fig. 4.12. Registro del UE en la red 5G.

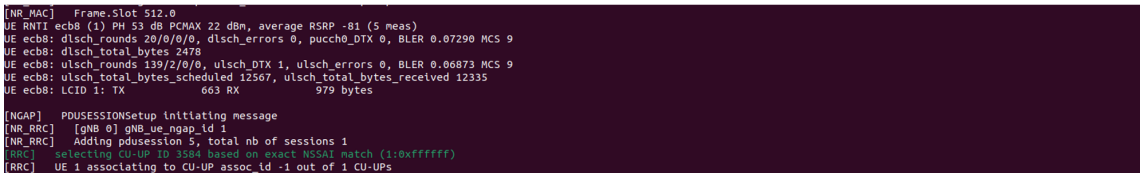


Fig. 4.13. UE finalmente conectado y transmitiendo en la red 5G.

La correcta configuración y ejecución del USRP es crucial para el funcionamiento de la red 5G, ya que permite la comunicación de radiofrecuencia entre los dispositivos de usuario y la infraestructura de la red.

Este proceso de configuración asegura que todos los componentes de la red 5G operen en armonía, proporcionando la funcionalidad necesaria para las comunicaciones deterministas en nuestro entorno de prueba. Cada paso se ha documentado cuidadosamente para garantizar la reproducibilidad y la fiabilidad del despliegue.

4.2.2. Implementación de Características Deterministas en 3GPP

Para asegurar la capacidad de red determinista dentro del dominio 3GPP, se han implementado varias mejoras clave. Estas mejoras permiten garantizar que el tráfico crítico mantenga sus requisitos de latencia, jitter y pérdida de paquetes, incluso en condiciones de red congestionadas. Las siguientes secciones describen estas mejoras en detalle.

La solución OpenAirInterface (OAI) utilizada en nuestro dominio 3GPP es una Radio Access Network (RAN) y un núcleo 5G open-source. Esta plataforma ha sido elegida debido a su flexibilidad para ser configurada y adaptada a necesidades específicas de red determinista. Además, OAI permite una fácil integración con otras tecnologías de red y ha sido fundamental para desarrollar una infraestructura 5G robusta.

Para implementar el tráfico determinista en 3GPP, se han realizado las siguientes mejoras:

1. **Implementación de Traffic Shaping (Throttling de Tráfico BE):** Se ha desarrollado un mecanismo de traffic shaping utilizando la disciplina de colas (qdisc) en

Linux. Este mecanismo permite la asignación de identidades de clase de calidad de servicio (QCI) para gestionar el tráfico no determinista, modelando su flujo de acuerdo con el ancho de banda disponible del enlace. Este controlador ajusta el throughput del tráfico BE entrante y lo limita para no sobrecargar el canal. El script utilizado para esta tarea se puede observar en el código 4.11. Esta técnica asegura que el tráfico Best Effort (BE) no interfiera con el tráfico determinista crítico, manteniendo así los niveles de calidad de servicio necesarios.

```
#!/bin/bash
IFNAME="enxaadd733ddee5"

# Definir valores DSCP para cada flujo
DSCP1="0x03"
DSCP2="0x04"

# Definir tasas de shaping para cada flujo
RATE1="3.8mbit"
RATE2="3.8mbit"

# Eliminar reglas de filtro existentes
tc qdisc del dev $IFNAME root

# Crear qdisc raz (htb) para la interfaz
tc qdisc add dev $IFNAME root handle 1: htb default 10

# Crear clases para cada flujo con las tasas de shaping
    respectivas
tc class add dev $IFNAME parent 1: classid 1:1 htb rate $RATE1
tc class add dev $IFNAME parent 1: classid 1:2 htb rate $RATE2

# Crear filtros para coincidir con paquetes con valores DSCP
    especficos y dirigirlos a clases respectivas
tc filter add dev $IFNAME protocol ip parent 1: prio 1 u32 match
    ip tos $DSCP1 0xff flowid 1:1
tc filter add dev $IFNAME protocol ip parent 1: prio 1 u32 match
    ip tos $DSCP2 0xff flowid 1:2
```

CÓDIGO 4.11. Código Bash para Throttling de Tráfico BE

2. **Selección Adaptativa de MCS:** En un sistema típico 3GPP, el algoritmo de Modulación y Codificación (MCS) se diseña para optimizar el throughput seleccionando un MCS que mantenga una tasa de error de bloque (BLER) de hasta el 10 %. Sin embargo, este enfoque puede introducir retransmisiones automáticas (ARQ) que afectan negativamente a la latencia y jitter. Para garantizar la fiabilidad determinista y eliminar las retransmisiones ARQ, ajustamos el MCS por debajo de las capacidades del canal para lograr un BLER del 0 %, asegurando transmisiones sin errores a través del aire.

La selección adaptativa de MCS implica monitorizar continuamente las condiciones del canal y ajustar dinámicamente el MCS utilizado. Este proceso asegura que siempre se seleccione un MCS que ofrezca la mejor combinación de baja latencia y alta fiabilidad, incluso en condiciones de canal fluctuantes. El algoritmo de MCS se basa en la calidad del enlace, medida a través de indicadores como la relación señal-ruido (SNR) y la tasa de error de bits (BER). Al elegir un MCS más bajo, se incrementa la robustez de la señal, lo que resulta en una mayor resistencia a las interferencias y una menor probabilidad de errores de transmisión.

Esta técnica es esencial para aplicaciones que no pueden tolerar ningún error en la transmisión de datos, como las comunicaciones de emergencia o los sistemas de control industrial. La eliminación de retransmisiones asegura que los paquetes de datos lleguen a su destino en el menor tiempo posible, manteniendo la latencia baja y constante. Esto es crítico para mantener la calidad de servicio en redes deterministas, donde cualquier variación en la latencia puede tener un impacto significativo en el rendimiento de la aplicación.

Estas mejoras aseguran que el tráfico determinista mantenga su integridad y fiabilidad, incluso en condiciones de alta carga de tráfico. La implementación de estas técnicas en el dominio 3GPP permite una operación eficiente y predecible de la red, garantizando que los requisitos de las aplicaciones críticas se cumplan de manera consistente.

4.2.3. Caracterización del Dominio 3GPP

Para evaluar y caracterizar el rendimiento del dominio 3GPP con las mejoras implementadas, se realizaron pruebas exhaustivas, siguiendo las directrices del artículo original. Estas pruebas incluyeron la generación de tráfico determinista y Best Effort (BE) en diferentes condiciones de carga de la red. Los parámetros clave evaluados fueron la latencia, el jitter y la tasa de error de bloque (BLER).

La configuración de las pruebas fue la siguiente:

1. **Cargas de tráfico determinista y BE:** Se generaron flujos de tráfico determinista de 1 Mbps junto con tráfico BE en tres escenarios: por debajo de la capacidad del enlace, igualando la capacidad del enlace y superando la capacidad del enlace.

2. **Evaluación de Diferentes MCS:** Se probaron cuatro MCS distintos (MCS 3, 9, 12 y 18) para observar su impacto en la latencia y la fiabilidad del tráfico determinista. Estos MCS fueron seleccionados para cubrir un rango representativo de condiciones de calidad del canal, desde situaciones de baja calidad (MCS 3) hasta condiciones óptimas (MCS 18).

4.2.3.1. Resultados de las Pruebas

Los resultados de las pruebas indicaron que el tráfico BE no afectó significativamente la latencia del tráfico determinista cuando la capacidad del enlace no se superó. Esto se debe a la implementación del throttling de tráfico BE, que garantiza que el tráfico determinista tenga prioridad y recursos suficientes para mantener su calidad de servicio.

4.2.3.2. Latencia y Fiabilidad del Tráfico Determinista

La latencia del tráfico determinista se mantuvo constante y baja en todos los escenarios probados con diferentes MCS, lo que demuestra la efectividad de la selección adaptativa de MCS y la gestión del tráfico. Por supuesto, en ninguno de estos casos se detectó perdida de paquetes, lo que sería inadmisible dada la naturaleza de este tráfico. En particular, se observó lo siguiente:

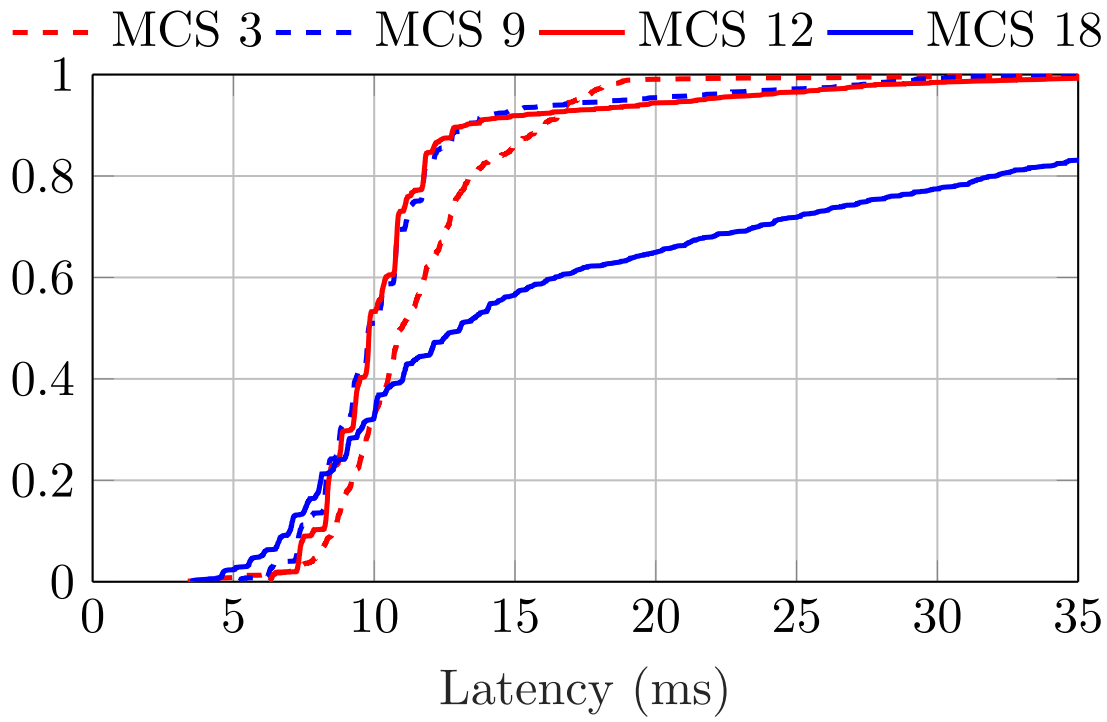


Fig. 4.14. Comparativa de MCSs.

- **MCS 3:** Alta robustez y baja tasa de errores, ideal para condiciones de canal pobres.
- **MCS 9 y 12:** Compromiso entre throughput y fiabilidad, adecuado para condiciones de canal medianas.
- **MCS 18:** Alta eficiencia de espectro pero sensible a condiciones de canal fluctuantes, adecuado solo para condiciones óptimas.

4.2.3.3. Impacto del Tráfico BE

Cuando se superó la capacidad del enlace, el tráfico BE experimentó aumentos significativos en la latencia y variabilidad (jitter), mientras que el tráfico determinista mantuvo su latencia dentro de los límites esperados, confirmando la eficacia de las mejoras implementadas. Este comportamiento es crucial para aplicaciones críticas donde la predictibilidad y la baja latencia son esenciales.

4.2.3.4. Conclusiones preliminares en 3GPP

El análisis de la tasa de error de bloque (BLER) mostró que al utilizar un MCS ajustado para mantener un BLER del 0 %, se eliminan las retransmisiones ARQ, asegurando transmisiones sin errores a través del aire. Este enfoque es fundamental para mantener la calidad de servicio en redes deterministas.

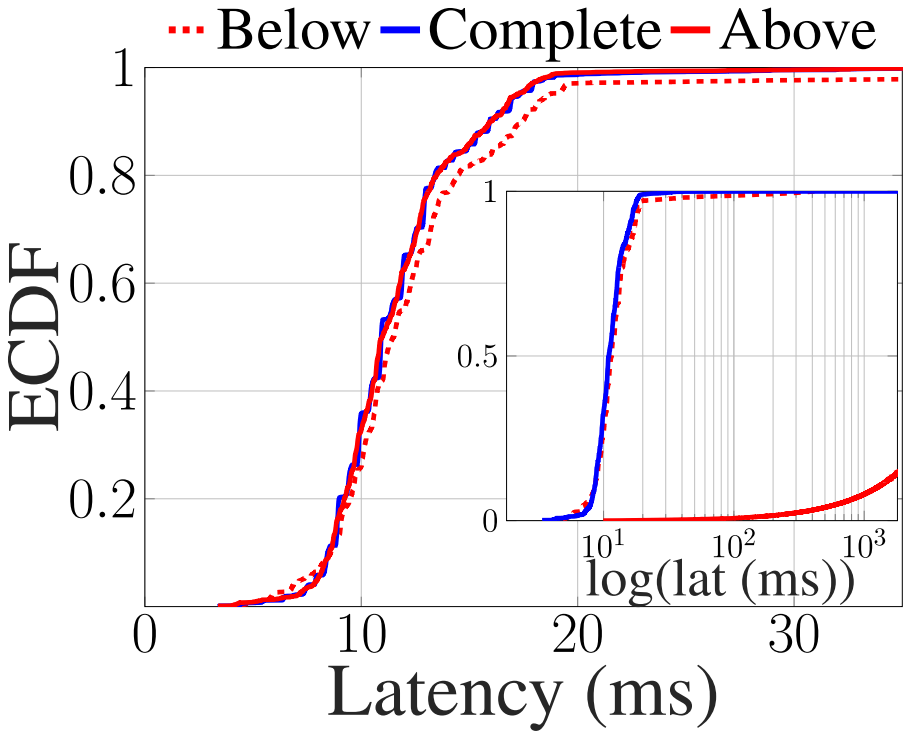


Fig. 4.15. Resultados de tráfico determinista (exterior) y BE (interior) para varios estados de canal (Below: Sin overflowing, Complete: Ajustando al Link Budget y Above: Superando el límite del canal).

Por otro lado, se puede concluir que el uso de un MCS adaptativo basado en la calidad del enlace permite:

- **Reducción de Retransmisiones:** Minimiza la necesidad de retransmisiones, reduciendo así la latencia y el jitter.

- **Optimización de Recursos:** Utiliza eficientemente el espectro disponible, ajustando dinámicamente las condiciones del canal.
- **Consistencia en Calidad de Servicio:** Mantiene la consistencia en la calidad del servicio, asegurando que las aplicaciones críticas funcionen sin interrupciones.

Estos resultados demuestran que las características deterministas implementadas en el dominio 3GPP mejoran significativamente la fiabilidad y la predictibilidad del tráfico crítico, incluso en condiciones de alta carga de tráfico. La implementación de estos mecanismos asegura que las redes 3GPP puedan soportar aplicaciones con requisitos estrictos de latencia y fiabilidad, proporcionando una plataforma robusta para el futuro de las comunicaciones críticas.

5. PLANO DE DATOS BASADO EN DETNET

Para lograr una integración efectiva entre distintos dominios tecnológicos y proporcionar un plano de datos determinista de extremo a extremo, aprovecho los avances recientes del grupo de trabajo de IETF Deterministic Networking (DetNet). DetNet se especializa en establecer rutas de datos deterministas que funcionan tanto en segmentos de capa 2 (enlaces de puente) como en capa 3 (enrutados), ofreciendo garantías en cuanto a reordenamiento, latencia, pérdida y variación del retardo de paquetes (jitter), además de alta fiabilidad. DetNet construye un plano de datos superpuesto en la capa 3, que opera sobre las características deterministas de cada dominio tecnológico, permitiendo una integración fluida y eficiente.

5.1. IETF Deterministic Networking (DetNet)

La arquitectura del plano de datos de DetNet se estructura en dos subcapas principales: la subcapa de servicio y la subcapa de reenvío. La subcapa de servicio está diseñada para ofrecer protección y reordenamiento dentro del servicio DetNet, lo que incluye la duplicación de paquetes para asegurar su entrega fiable, así como la eliminación de duplicados y la reordenación de los paquetes en su destino final. Por su parte, la subcapa de reenvío utiliza mecanismos de ingeniería de tráfico para proporcionar protección contra la congestión, asegurando una baja pérdida de paquetes, latencia garantizada y entrega limitada fuera de orden. Cada subcapa de reenvío puede contar con capacidades específicas que no están disponibles en otras subcapas, permitiendo una optimización adaptada a las características y necesidades del entorno de red.

Según la arquitectura de DetNet, los sistemas finales conectados a un dominio DetNet encapsulan los paquetes de acuerdo con sus requisitos de servicio específicos. Los nodos de borde de DetNet funcionan en ambas subcapas, de servicio y de reenvío, para proporcionar las funcionalidades necesarias a los paquetes entrantes, asegurando que se cumplan los requisitos de calidad de servicio y fiabilidad. Los nodos de tránsito de DetNet, que pueden no estar al tanto de los requisitos específicos de la subcapa de servicio de DetNet, proporcionan las capacidades de Calidad de Servicio (QoS) requeridas para los flujos de tráfico. Estos nodos de tránsito gestionan el tráfico de manera que se mantengan las garantías de rendimiento a lo largo de todo el recorrido del paquete.

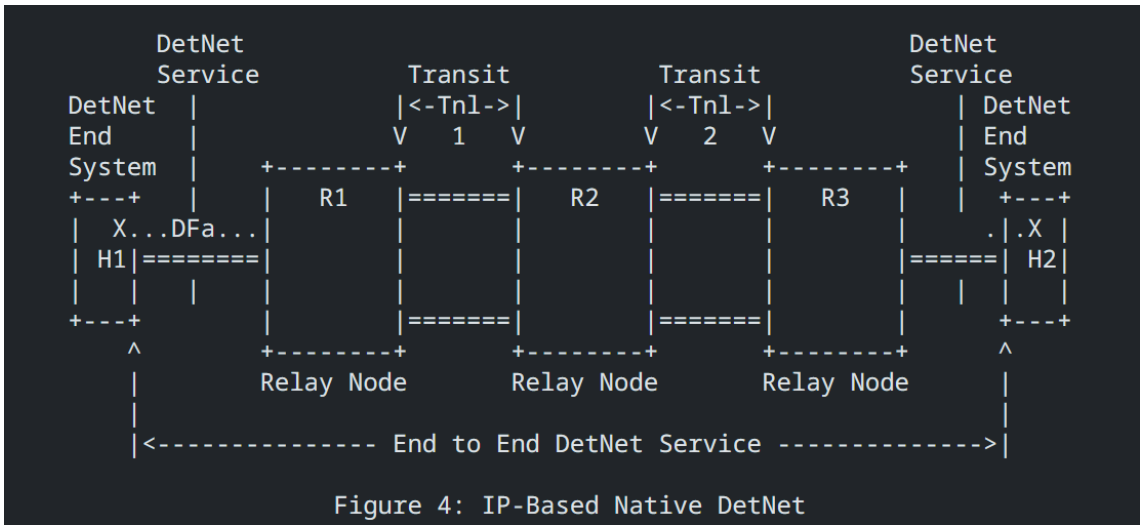


Fig. 5.1. Arquitectura DetNet. Fuente: [53].

Aunque no está incluido en los esfuerzos actuales de estandarización, la arquitectura de IETF DetNet también contempla otros componentes cruciales. Entre ellos, se encuentra un plano de control y gestión, diseñado para programar el plano de datos de los nodos de DetNet, permitiendo así proporcionar la calidad de servicio (QoS) por flujo necesaria para aplicaciones críticas. Además, incluye mecanismos de operaciones, administración y mantenimiento (OAM), esenciales para mantener una red determinista operativa y fiable. Estos mecanismos permiten monitorear el rendimiento de la red, detectar y corregir problemas de manera proactiva y garantizar que los niveles de servicio se mantengan consistentemente altos.

En los capítulos siguientes, describo cómo he integrado este plano de datos en los dominios mencionados anteriormente para permitir el soporte de comunicaciones deterministas multidominio. Esto se ha logrado utilizando el plano de datos de IETF DetNet sobre TSN cableado y la implementación de 5G Open Source. Esta integración ha permitido desarrollar una infraestructura de red capaz de cumplir con los estrictos requisitos de rendimiento y fiabilidad necesarios para aplicaciones deterministas en entornos complejos y diversificados.

5.2. Implementación y Optimizaciones para la Integración Multidominio

Como se menciona en la sección anterior, el grupo de trabajo de IETF DetNet está desarrollando soluciones para rutas de datos deterministas desde una perspectiva de capa 3. La arquitectura de DetNet se presenta como una solución flexible que mejora el determinismo y la fiabilidad de las transmisiones de paquetes mediante diversos mecanismos, como la asignación de recursos (dependiendo de las capacidades de la tecnología de capa 2 utilizada) y la configuración de rutas explícitas. La solución del plano de datos de DetNet se divide en subcapas de servicio y de reenvío. La subcapa de reenvío transporta la información necesaria para alcanzar el siguiente salto y proporciona funciones

de QoS para cumplir con los requisitos del flujo, por ejemplo, a través de métodos de cola o utilizando la conectividad subyacente de capa 2. La subcapa de servicio añade funcionalidades adicionales, como la replicación, eliminación y ordenación de paquetes (PREOF). Para lograr esto, DetNet codifica atributos específicos del flujo (identidad del flujo y número de secuencia) en los paquetes.

Se especifican diferentes tecnologías de plano de datos para ser utilizadas como planos de datos de DetNet. He adoptado la solución de MPLS sobre UDP/IP [54], ya que permite implementar completamente las funcionalidades de las subcapas de reenvío y de servicio del plano de datos de DetNet. Aprovechando la arquitectura de IETF DetNet, he decidido integrar múltiples dominios, haciendo que los nodos en los límites de los diferentes dominios operen como puertas de enlace con funcionalidades adicionales. Considero tres tipos principales de funciones DetNet: reenvío, para la transmisión de flujos de paquetes entre dominios; encapsulación y decapsulación. La ruta a través de los dominios se calcula globalmente, teniendo en cuenta los requisitos del servicio y las capacidades de cada dominio. La subcapa de servicio puede implementar PREOF para cumplir con requisitos que los dominios individuales no pueden manejar por sí solos. Esta función mejora las capacidades deterministas al replicar paquetes y eliminar duplicados, protegiendo contra retrasos en la retransmisión y pérdida de paquetes; y ordenándolos antes de salir del sistema para minimizar el jitter. Cabe destacar que para los experimentos realizados en este trabajo, configuré de forma estática las rutas para cada experimento de antemano.

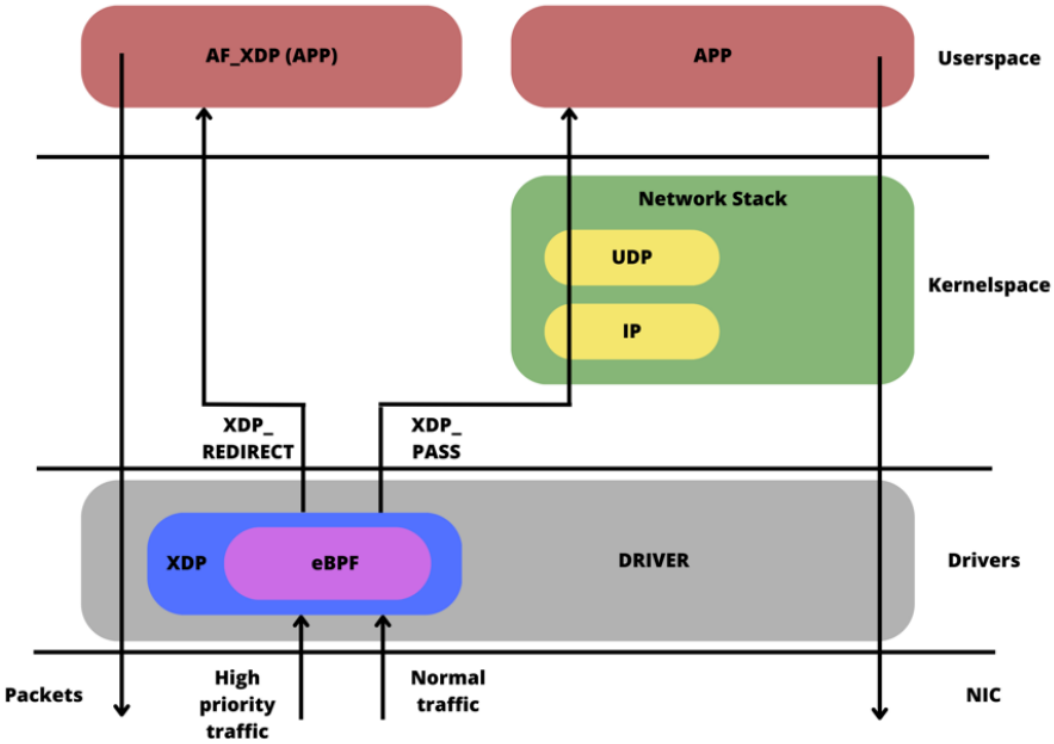


Fig. 5.2. Arquitectura XDP DetNet Router.

En cuanto a mi implementación, cualquier operación en el núcleo del sistema ope-

rativo carece inherentemente de determinismo debido a la programación impredecible y al manejo de interrupciones, lo que puede introducir latencia y jitter en el procesamiento de paquetes de red. Esto significa que construir una solución independiente para DetNet en hardware TSN no específico y de código abierto presenta desafíos significativos. Para superar esto, he desarrollado una solución utilizando la tecnología eXpress Data Path (XDP) [55] and Extended Berkeley Packet Filter (eBPF) [56], un plano de datos de red programable y de alto rendimiento en el núcleo de Linux. Al aprovechar XDP, puedo interceptar y procesar paquetes en el punto más temprano de la pila de software, omitiendo completamente la pila de red del núcleo como se muestra en la figura 5.2. Este enfoque no solo minimiza la latencia al eliminar la sobrecarga innecesaria del núcleo, sino que también asegura un manejo de paquetes más predecible y determinista. Estos beneficios son cruciales para proporcionar la gestión de paquetes robusta y eficiente necesaria para mi investigación y desarrollo avanzados en redes deterministas. [57]

El dominio 3GPP está integrado como un puente y los nodos de DetNet conectados a él actúan como traductores del lado de la red y del dispositivo (NW-TT y DS-TT), según lo definido por las especificaciones de 3GPP. Esto facilita la funcionalidad de extremo a extremo (E2E), ocultando los procedimientos específicos internos de 3GPP.

En la Figura 2, se muestran dos puntos finales (T1 y T2) conectados a través de dos hipotéticos dominios, independientemente de la naturaleza de estos, ya que DetNet es agnóstico a la tecnología subyacente. Las pilas de protocolos y los diferentes túneles involucrados en los experimentos también se ilustran en la figura. Mi implementación adopta el plano de datos de DetNet de MPLS sobre UDP/IP, utilizando la etiqueta de servicio MPLS (S-label) para identificar los flujos DetNet y la palabra de control DetNet (d-CW) para la secuenciación y la identificación de paquetes duplicados de un flujo DetNet en la subcapa de servicio DetNet.

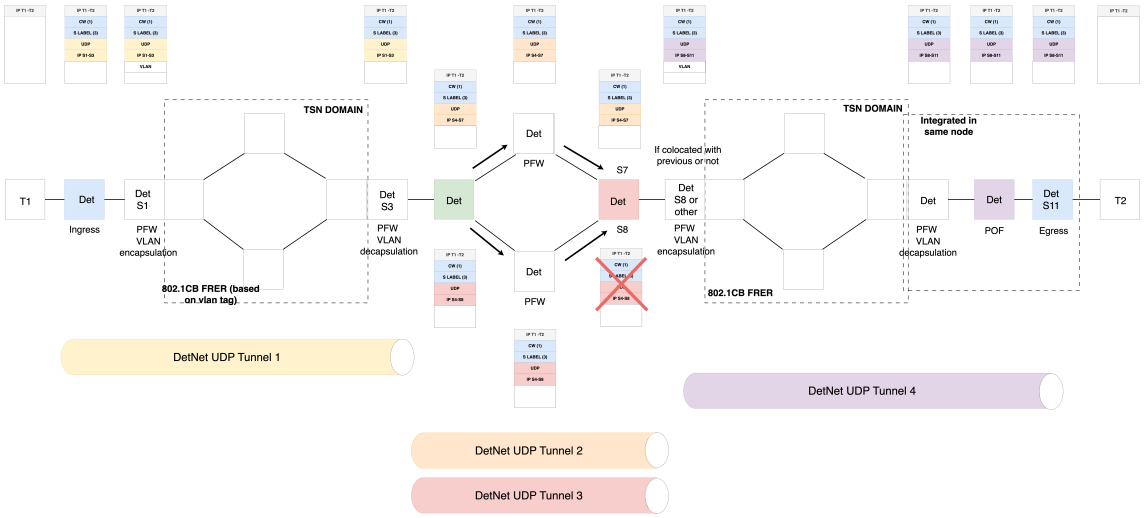


Fig. 5.3. Detnet sobre multidominio.

5.3. Configuración y código base de los nodos DetNet

En esta sección, se presenta la configuración y el código base utilizado para implementar los nodos DetNet en el entorno de pruebas. El objetivo es proporcionar una infraestructura que permita la gestión de flujos de datos deterministas de extremo a extremo a través de diferentes dominios tecnológicos.

La implementación está basada en el lenguaje de programación Go, el cual ofrece un rendimiento alto y una excelente capacidad para manejar concurrencia, lo cual es crucial para las aplicaciones de redes deterministas.

5.3.1. Constantes

En primer lugar, definimos las constantes utilizadas en el código. Estas constantes establecen los tamaños y los desplazamientos de los distintos encabezados utilizados en la implementación.

```
const L2_size = 18
const L3_size = 20
const L4_size = 8
const MPLS_size = 8

const nxheadL2_offset_1 int = 12
const nxheadL2_offset_2 int = 13
const slabel_offset int = L2_size + L3_size + L4_size
const vlan_offset int = L2_size - 4
const iplen_offset_1 int = L2_size + 2
const iplen_offset_2 int = L2_size + 3
const ipChkSm_offset_1 int = L2_size + 10
const ipChkSm_offset_2 int = L2_size + 11
const udplen_offset_1 int = L2_size + L3_size + 4
const udplen_offset_2 int = L2_size + L3_size + 5
const header_len = L2_size + L3_size + L4_size + MPLS_size
```

CÓDIGO 5.1. Definición de constantes.

Las constantes definidas arriba especifican tamaños y desplazamientos cruciales para el manejo de los encabezados de las distintas capas (L2, L3, L4 y MPLS). Estas constantes son esenciales porque permiten calcular con precisión las posiciones específicas dentro de los paquetes de datos. Por ejemplo, L2_size representa el tamaño del encabezado de capa 2, L3_size el de la capa 3, L4_size el de la capa 4 y MPLS_size el del encabezado MPLS. Los desplazamientos, como nxheadL2_offset_1 y nxheadL2_offset_2, indican las posiciones exactas dentro de los encabezados donde se pueden encontrar o modificar valores específicos, como direcciones MAC, longitudes de paquetes y sumas de verificación. Esta información detallada es vital para manipular correctamente los datos en los nodos DetNet, asegurando que cada paquete sea procesado con precisión y que las modificaciones

necesarias se realicen en las ubicaciones correctas del paquete.

5.3.2. Estructuras

Las siguientes estructuras representan los diferentes encabezados y servicios que los nodos DetNet manejarán.

```
type L2_info struct {
  Src string  `json:"src_mac,omitempty"`
  Dst string  `json:"dst_mac,omitempty"`
  Vid uint16  `json:"flow_id"`
  Pcp uint8   `json:"pcp"`
  Vlan string `json:"vlan,omitempty"`
}
type L3_info struct {
  Src string `json:"src_ip"`
  Dst string `json:"dst_ip"`
}
type L4_info struct {
  Src uint16 `json:"src_port"`
  Dst uint16 `json:"dst_port"`
}
type Service struct {
  Action string  `json:"action"`
  L2     L2_info `json:"L2,omitempty"`
  L3     L3_info `json:"L3,omitempty"`
  L4     L4_info `json:"L4,omitempty"`
}
```

CÓDIGO 5.2. Definición de estructuras.

Las estructuras L2_info, L3_info, L4_info y Service definen la información de los encabezados de las capas 2, 3 y 4, así como las acciones de servicio que se realizarán en los paquetes de datos. La estructura L2_info contiene campos para las direcciones MAC de origen y destino, el ID de flujo (Vid), la prioridad del tráfico (Pcp) y una indicación de VLAN. L3_info define las direcciones IP de origen y destino, mientras que L4_info especifica los puertos de origen y destino. La estructura Service encapsula toda esta información y añade un campo de acción que determina qué operación se realizará con los datos del paquete (por ejemplo, encapsulación o reenvío). Estas estructuras permiten una representación clara y organizada de los datos necesarios para el procesamiento de paquetes en los nodos DetNet, facilitando la lectura y manipulación de la información de los encabezados a medida que los paquetes pasan a través de la red.

5.3.3. Funciones Básicas

A continuación, se presentan las funciones básicas que manipulan los encabezados y calculan checksums.

```
func L2_header(data L2_info) []byte {
  a, _ := strconv.ParseInt(strconv.FormatInt(int64(data.Vid), 2) +
    strconv.FormatInt(int64(data.Pcp), 2), 2, 16)
  src, _ := hex.DecodeString(data.Src)
  dst, _ := hex.DecodeString(data.Dst)
  b := make([]byte, 2)
  binary.BigEndian.PutUint16(b, uint16(a))
  if data.Vlan == "true" {
    return append(dst, append(src, append([]byte{0x81, 0x00},
      append(b, []byte{0x08, 0x00}...)...)...)...)
  } else {
    return append(dst, append(src, []byte{0x08, 0x00}...)...)
  }
}

func L3_header(data L3_info) []byte {
  return append([]byte{0x45, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x40, 0x00,
    0x40, 0x11, 0x00, 0x00},
    append(parseIPv4addr(data.Src),
      parseIPv4addr(data.Dst)...)...)
}

func L4_header(data L4_info) []byte {
  src, dst := make([]byte, 2), make([]byte, 2)
  binary.BigEndian.PutUint16(src, data.Src)
  binary.BigEndian.PutUint16(dst, data.Dst)
  return append(src, append(dst, []byte{0, 0, 0, 0}...)...)
}

func MPLS_slabel(slabel uint16) []byte {
  lab := make([]byte, 2)
  binary.BigEndian.PutUint16(lab, slabel)
  return append(lab, []byte{0, 255}...)
}

func MPLS_seqnum() []byte {
  if seqNum == 0xFFFFFFF {
    seqNum = 0
  } else {
    seqNum++
  }
  b := make([]byte, 4)
  binary.BigEndian.PutUint32(b, seqNum)
  return b
```

```
46  }
47
48  func calculateChecksum(data []byte) uint16 {
49      var sum uint32
50      // Sum up 16-bit words
51      for i := 0; i < len(data); i += 2 {
52          if i+1 < len(data) {
53              sum += uint32(data[i])<<8 + uint32(data[i+1])
54          } else {
55              sum += uint32(data[i]) << 8 // In case of odd length
56          }
57      }
58      // Add the carries
59      for (sum >> 16) > 0 {
60          sum = (sum & 0xFFFF) + (sum >> 16)
61      }
62      // Return the one's complement of sum
63      return uint16(^sum)
64  }
65
66  func parseIPv4addr(addr string) []byte {
67    a := strings.Split(addr, ".")
68    b := make([]byte, 4)
69    for i := 0; i < 4; i++ {
70      c, _ := strconv.Atoi(a[i])
71      b[i] = byte(c)
72    }
73    return b
74  }
```

CÓDIGO 5.3. Definición de las funcionalidades basicas.

Las funciones L2_header, L3_header y L4_header construyen los encabezados para las capas 2, 3 y 4 respectivamente. Cada función toma como argumento una estructura correspondiente (L2_info, L3_info o L4_info) y devuelve un arreglo de bytes que representa el encabezado. MPLS_slabel y MPLS_seqnum manejan las etiquetas y secuencias MPLS. MPLS_slabel crea una etiqueta MPLS basada en un valor dado, mientras que MPLS_seqnum genera un número de secuencia MPLS, asegurando que los paquetes se puedan rastrear de manera única. La función calculateChecksum calcula el checksum para los encabezados IP, sumando palabras de 16 bits y retornando el complemento de la suma. Esta función es crucial para asegurar la integridad de los datos transmitidos. Por último, parseIPv4addr convierte direcciones IP en su representación en bytes, dividiendo la dirección en sus componentes y convirtiéndolos a su forma binaria.

5.3.4. Funciones Complejas

Finalmente, se presentan las funciones complejas que manejan la lógica principal de procesamiento de paquetes y la inicialización de los nodos DetNet.

```
var seqNum uint32 = 0
var headers = make(map[[2]byte][]byte)
var actions = make(map[[2]byte]string)
var newFlows = make(map[[2]byte][2]byte)

func detnet(packet []byte) []byte {
    var slabel [2]byte
    var off int
    if packet[nxheadL2_offset_1] == byte{0x81} && packet[nxheadL2_offset_2] == byte{0x00} {
      copy(slabel[:], packet[slabel_offset:slabel_offset+2])
  off = 0
    } else {
  copy(slabel[:], packet[slabel_offset-4:slabel_offset-2])
  off = 4
    }

    switch actions[slabel] {
    case "forward":
  if len(headers[slabel]) == header_len {
          headers[slabel][iplen_offset_1] = packet[iplen_offset_1-off]
          headers[slabel][iplen_offset_2] = packet[iplen_offset_2-off]
          return append(headers[slabel][:slabel_offset], packet[slabel_offset-off:]...)
  } else {
              headers[slabel][iplen_offset_1-4] = packet[iplen_offset_1-off]
              headers[slabel][iplen_offset_2-4] = packet[iplen_offset_2-off]
              return append(headers[slabel][:slabel_offset-4], packet[slabel_offset-off:]...)
  }
    case "pop":
        return append(headers[slabel], packet[header_len:]...)
    default:
        var flow [2]byte
        copy(flow[:], packet[vlan_offset:vlan_offset+2])
        slabel, ok := newFlows[flow]
        if ok {
      if len(headers[slabel]) == header_len {
        ip_length := uint16(L3_size + L4_size + MPLS_size + len(packet[L2_size:]))
        bin_ip_length := make([]byte, 2)
```

```
38          binary.BigEndian.PutUint16(bin_ip_length, ip_length)
39          udp_length := uint16(L4_size + MPLS_size + len(packet[
    L2_size:]))
40          bin_udp_length := make([]byte, 2)
41          binary.BigEndian.PutUint16(bin_udp_length, udp_length)
42          out := append(headers[slabel], packet[L2_size:]...)
43          out[iplen_offset_1] = bin_ip_length[0]; out[iplen_offset_2]
    = bin_ip_length[1]
44                                  out[udplen_offset_1] =
    bin_udp_length[0]; out[udplen_offset_2] = bin_udp_length[1]
45          //out := append(headers[slabel][:udplen_offset_1], append(
    bin_udp_length, append(headers[slabel][udplen_offset_2+1:],
46          //  packet[L2_size:]...)...)...)
47          chkSm := calculateChecksum(out[L2_size:])
48          bin_chkSm := make([]byte, 2)
49          binary.BigEndian.PutUint16(bin_chkSm, chkSm)
50          out[ipChkSm_offset_1] = bin_chkSm[0]; out[ipChkSm_offset_2]
    = bin_chkSm[1]
51          //out = append(out[:ipChkSm_offset_1], append(bin_chkSm, out
    [ipChkSm_offset_2+1:]...)...)
52          return out
53        } else {
54                                  ip_length := uint16(L3_size +
    L4_size + MPLS_size + len(packet[L2_size:]))
55                                  bin_ip_length := make([]byte, 2)
56                                  binary.BigEndian.PutUint16(
    bin_ip_length, ip_length)
57          udp_length := uint16(L4_size + MPLS_size + len(packet[
    L2_size:]))
58          bin_udp_length := make([]byte, 2)
59          binary.BigEndian.PutUint16(bin_udp_length, udp_length)
60                                  out := append(headers[slabel],
    packet[L2_size:]...)
61                                  out[iplen_offset_1-4] =
    bin_ip_length[0]; out[iplen_offset_2-4] = bin_ip_length[1]
62                                  out[udplen_offset_1-4] =
    bin_udp_length[0]; out[udplen_offset_2-4] = bin_udp_length[1]
63          //out := append(headers[slabel][:udplen_offset_1-4], append(
    bin_udp_length, append(headers[slabel][udplen_offset_2-3:],
64          //  packet[L2_size:]...)...)...)
65          chkSm := calculateChecksum(out[L2_size-4:L2_size+L3_size-4])
66          bin_chkSm := make([]byte, 2) //[2]byte{0x24, 0x75} //make([]
    byte, 2)
67          binary.BigEndian.PutUint16(bin_chkSm, chkSm)
68                                  out[ipChkSm_offset_1-4] = bin_chkSm
    [0]; out[ipChkSm_offset_2-4] = bin_chkSm[1]
69          //out = append(out[:ipChkSm_offset_1-4], append(bin_chkSm,
    out[ipChkSm_offset_2-3:]...)...)
70          return out
71        }
72        }
```

```
73      }
74      return nil
75  }
76
77  func detnet_init(detnetConf_path string, newFlows_path string) bool
        {
78    detnetConf_file, err := os.ReadFile(detnetConf_path)
79    if err != nil {
80      fmt.Println(err)
81    }
82    fmt.Println("Successfully Opened " + detnetConf_path)
83    var services = make(map[string]Service)
84    json.Unmarshal(detnetConf_file, &services)
85    fmt.Println(services)
86    for k, v := range services {
87          var b [2]byte
88          c, _ := strconv.Atoi(k)
89          binary.BigEndian.PutUint16(b[:], uint16(c))
90      switch v.Action {
91      case "encapsulate":
92        headers[[2]byte(b)] = append(L2_header(v.L2), append(L3_header
      (v.L3),
93            append(L4_header(v.L4), append(MPLS_slabel(uint16(c)), []
      byte{0, 0, 0, 0}...)...)...)...)
94      case "forward":
95        actions[[2]byte(b)] = "forward"
96        headers[[2]byte(b)] = append(L2_header(v.L2), append(L3_header
      (v.L3),
97            append(L4_header(v.L4), append(MPLS_slabel(uint16(c)), []
      byte{0, 0, 0, 0}...)...)...)...)
98      case "pop":
99        actions[[2]byte(b)] = "pop"
100       headers[[2]byte(b)] = L2_header(v.L2)
101     }
102   }
103   flows_file, _ := os.ReadFile(newFlows_path)
104   flows := make(map[string]string)
105   fmt.Println("Successfully Opened " + detnetConf_path)
106   json.Unmarshal(flows_file, &flows)
107    for k, v := range flows {
108         var flow [2]byte
109         f, _ := strconv.Atoi(k)
110         binary.BigEndian.PutUint16(flow[:], uint16(f))
111         var slabel [2]byte
112         s, _ := strconv.Atoi(v)
113         binary.BigEndian.PutUint16(slabel[:], uint16(s))
114         newFlows[flow] = slabel
115     }
116
117   fmt.Println(headers)
118   fmt.Println(actions)
```

```
119     fmt.Println(newFlows)
120     return true
121 }
```

CÓDIGO 5.4. Definición de las funcionalidades más avanzadas.

La función detnet procesa los paquetes recibidos y aplica la acción correspondiente según la configuración (encapsular, reenviar o eliminar encabezados). Esta función comienza extrayendo la etiqueta MPLS del paquete y determinando el desplazamiento necesario en función de si el paquete contiene información de VLAN. Luego, dependiendo de la acción configurada (forward, pop o una acción por defecto), la función manipula los encabezados del paquete de manera adecuada, ya sea actualizando longitudes y checksums o ajustando las etiquetas MPLS.

Por otro lado, la función detnet_init inicializa las configuraciones del nodo, cargando las configuraciones desde archivos JSON y estableciendo los encabezados y acciones necesarios. Esta función lee los archivos de configuración que contienen detalles sobre los servicios y flujos de datos, deserializa esta información en estructuras de Go y luego configura los encabezados y las acciones basadas en esta información. Esto permite que los nodos DetNet se inicialicen con los parámetros correctos para manejar los paquetes de datos de manera determinista desde el inicio, asegurando que la red opere de acuerdo a las especificaciones necesarias para aplicaciones críticas.

En conclusión, el código anterior establece las bases para la implementación de los nodos DetNet, utilizando distintas estructuras y funciones para manejar los encabezados de capa 2, capa 3, capa 4 y MPLS.

Los componentes principales de este código son:

- **L2, L3 y L4 Headers:** Estas funciones crean los encabezados necesarios para las capas 2, 3 y 4, asegurando que los paquetes sean correctamente formateados y direccionados a través de la red.
- **MPLS S-Label y SeqNum:** Estas funciones manejan la creación y gestión de etiquetas MPLS y números de secuencia, esenciales para la operación de DetNet.
- **Funciones de Servicio DetNet:** La función `detnet` maneja la lógica principal de reenvío, encapsulación y decapsulación de paquetes, basándose en la configuración proporcionada.
- **Inicialización de DetNet:** La función `detnet_init` se encarga de inicializar la configuración de DetNet, leyendo los archivos de configuración y estableciendo las estructuras de datos necesarias para la operación.

Este código es fundamental para proporcionar las capacidades deterministas necesarias para la integración de múltiples dominios, asegurando una comunicación fiable y predecible a través de la infraestructura de red.

A continuación, se muestra el código Go utilizado para la inicialización y la ejecución de DetNet sobre XDP. Este código es esencial para interceptar y procesar los paquetes en el punto más temprano de la pila de software, asegurando un manejo de paquetes más predecible y determinista:

```
detnet_init("./config/detnetData.json", "./config/newFlows.json")
log.Printf("Detnet deployed...")

func forwardFrames(input *xdp.Socket, output *xdp.Socket) (numBytes
    uint64, numFrames uint64) {
    inDescs := input.Receive(input.NumReceived())
    outDescs := output.GetDescs(output.NumFreeTxSlots(), false)

    if len(inDescs) > len(outDescs) {
        inDescs = inDescs[:len(outDescs)]
    }
    numFrames = uint64(len(inDescs))

    for i := 0; i < len(inDescs); i++ {
        inFrame := input.GetFrame(inDescs[i])
        Frame2send := detnet(inFrame)
        outFrame := output.GetFrame(outDescs[i])
        numBytes += uint64(len(Frame2send))
        outDescs[i].Len = uint32(copy(outFrame, Frame2send))
    }
    outDescs = outDescs[:len(inDescs)]
    output.Transmit(outDescs)

    return
}
```

CÓDIGO 5.5. Código Go para la ejecución de DetNet sobre XDP.

El fragmento de código anterior muestra cómo se inicializa DetNet con la configuración proporcionada en los archivos ‘detnetData.json‘ y ‘newFlows.json‘. Además, se detalla la función ‘forwardFrames‘, la cual se encarga de recibir paquetes, procesarlos a través de DetNet y reenviarlos. Este proceso asegura que los paquetes sean manejados de manera eficiente y determinista.

A continuación, se presentan los archivos de configuración utilizados en la implementación. Estos archivos definen las acciones a realizar para cada flujo de datos, incluyendo el encapsulamiento, el reenvío y la eliminación de encabezados:

```
{
    "30": {
        "action": "forward",
        "L2": {
            "src_mac": "00301808c9f148",
            "dst_mac": "00301808afe0",
```

```
            "flow_id": 2,
            "pcp": 1,
            "vlan": "true"
        },
        "L3": {
            "src_ip": "10.10.10.4",
            "dst_ip": "10.10.10.5"
        },
        "L4": {
            "src_port": 4000,
            "dst_port": 4000
        }
    },
    "10": {
        "action": "encapsulate",
        "L2": {
            "src_mac": "70f8e7d02467",
            "dst_mac": "b44506aee3ac",
            "flow_id": 3,
            "pcp": 1,
            "vlan": "false"
        },
        "L3": {
            "src_ip": "10.10.10.3",
            "dst_ip": "10.10.10.2"
        },
        "L4": {
            "src_port": 4000,
            "dst_port": 4000
        }
    },
    "40": {
        "action": "encapsulate",
        "L2": {
            "src_mac": "00301808c9f148",
            "dst_mac": "00301808afe0",
            "flow_id": 3,
            "pcp": 1,
            "vlan": "true"
        },
        "L3": {
            "src_ip": "10.10.10.3",
            "dst_ip": "10.10.10.2"
        },
        "L4": {
            "src_port": 4000,
            "dst_port": 4000
        }
    },
    "20": {
        "action": "pop",
```

```
58            "L2": {
59                "src_mac": "00301808c9f148",
60                "dst_mac": "00301808afe0",
61                "flow_id": 1,
62                "pcp": 1,
63                "vlan": "true"
64            }
65        }
66    }
```

CÓDIGO 5.6. Archivo de configuración detnetData

Este archivo JSON detnetData.json define las acciones que se deben tomar para diferentes etiquetas de servicio (slabel) en el entorno DetNet. Por ejemplo, la etiqueta 30 especifica que los paquetes deben ser reenviados, mientras que las etiquetas 10 y 40 indican que los paquetes deben ser encapsulados con ciertas cabeceras de L2, L3 y L4. La etiqueta 20 especifica que los paquetes deben ser despojados de su encapsulamiento (pop).

La configuración precisa de estos archivos y las funciones de manejo de paquetes en el código Go son claves para garantizar la calidad de servicio y el rendimiento determinista de la red implementada.

6. EXPERIMENTACIÓN Y RESULTADOS

Este capítulo presenta los experimentos realizados y los resultados obtenidos a través del testbed multidominio diseñado para evaluar la viabilidad y eficacia de las soluciones de networking deterministas integradas en los dominios de TSN y 3GPP. Se detalla la estructura del testbed, los escenarios de prueba implementados, y los casos de uso específicos que demuestran la aplicabilidad de estas tecnologías en entornos críticos.

6.1. Configuración del Testbed

La configuración del testbed utilizada para los experimentos se diseñó para simular un entorno de red multidominio que integra tecnologías de TSN y 3GPP. Como se ilustra en la Figura 6.1, el testbed se compone de varios componentes críticos que facilitan la transmisión de tráfico determinista y Best-Effort (BE) a través de diferentes medios de transmisión y tecnologías de red.

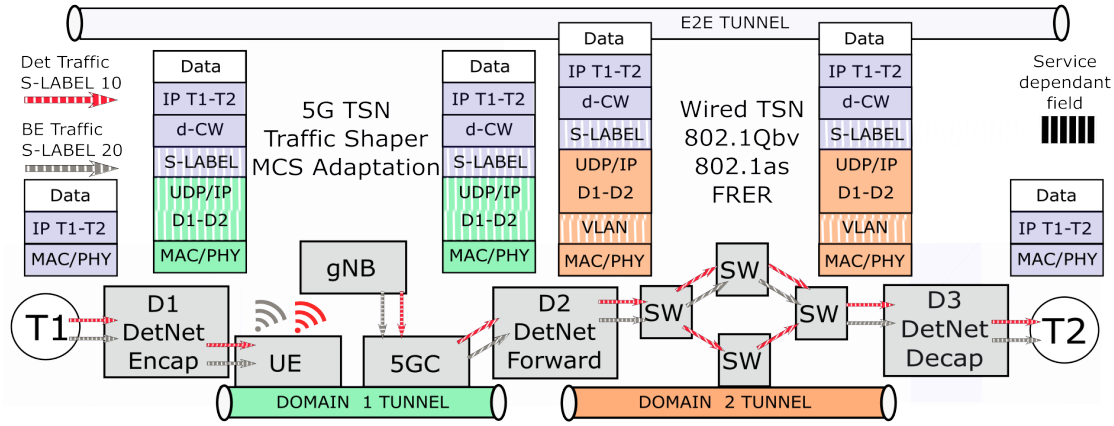


Fig. 6.1. Configuración general del testbed utilizado en los experimentos.

El esquema se divide en varias secciones críticas, detalladas a continuación:

1. **DetNet Encapsulation (D1 Encap)** y **DetNet Forwarding (D2 Forward)**: estos nodos son responsables de la encapsulación y reenvío de tráfico determinista. Utilizan etiquetas específicas (S-LABEL 10 para tráfico determinista y S-LABEL 20 para tráfico BE) para diferenciar y gestionar los flujos de datos a través de la red.

2. **T1 y T2**: representan los terminales de origen y destino, respectivamente, entre los cuales se transmite el tráfico. Estos terminales pueden ser dispositivos de usuario finales o gateways de red.

3. **5G TSN Traffic Shaper and MCS Adaptation**: este módulo ajusta la calidad de servicio mediante el modelado y la adaptación de los paquetes a través del shaper

de tráfico y algoritmos de adaptación de Modulation and Coding Scheme (MCS), garantizando así la integridad del tráfico crítico bajo diversas condiciones de red.

4. **E2E (End-to-End) Tunnel**: demuestra la capacidad del sistema para mantener una comunicación continua y segura entre los terminales a través de dominios múltiples, utilizando tecnologías como 802.1Qbv y Frame Replication and Elimination for Reliability (FRER) en redes TSN cableadas, junto con el encapsulamiento y manejo de tráfico en 5G.

5. **gNB y 5GC**: estos componentes del sistema 5G actúan como nodos de acceso y núcleo de red, respectivamente, facilitando la comunicación y el manejo de datos en el dominio móvil.

6. **DetNet Decapsulation (D3 Decap)**: este nodo es crucial para el proceso de desencapsulación del tráfico DetNet, asegurando que los datos se reintegren correctamente al llegar al destino, T2. D3 desempeña un papel fundamental en la restauración del formato original de los datos para su procesamiento final y entrega.

Esta configuración permite evaluar la interoperabilidad y el rendimiento de las soluciones de networking deterministas en un entorno controlado, que refleja una variedad de condiciones de red y carga de tráfico.

6.2. Escenarios y Casos de Uso

Los escenarios de prueba están diseñados para evaluar la eficacia de las soluciones implementadas en el manejo del tráfico crítico bajo diversas condiciones de red. Uno de los casos de uso destacados es el manejo autónomo de un perro robot, como se describe en la sección 6.2.1.

6.2.1. Caso de Uso: Control Autónomo de Perro Robot

Este caso de uso implica el control remoto de un perro robot en un entorno industrial donde la latencia y la fiabilidad son críticas. El perro robot, equipado con múltiples sensores y capacidad de respuesta en tiempo real, debe ejecutar tareas de transporte y mantenimiento sin intervención humana directa. Para que esto sea posible, se requiere una red que pueda asegurar la entrega oportuna y confiable de los comandos y datos de sensores entre los puntos de control y el robot.

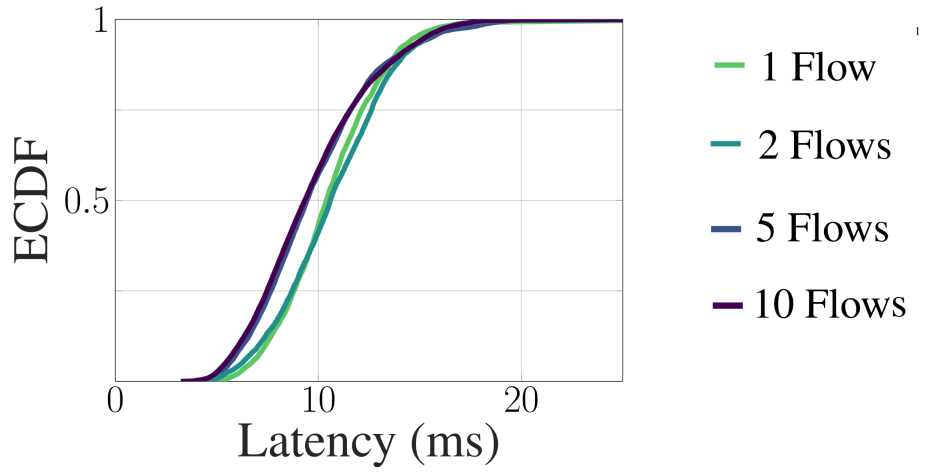
La Figura 6.2 muestra la implementación en hardware real de nuestro caso de uso. Aunque la imagen no presenta directamente al perro robot, ilustra la configuración de los componentes de la red utilizados para demostrar la viabilidad del tráfico determinista entre los nodos T1 y T2. Esta configuración es esencial para asegurar que, en un entorno real, un perro robot pueda operar eficientemente bajo las mismas condiciones de red.

Fig. 6.2. Implementación en hardware real del caso de uso en el entorno de prueba.

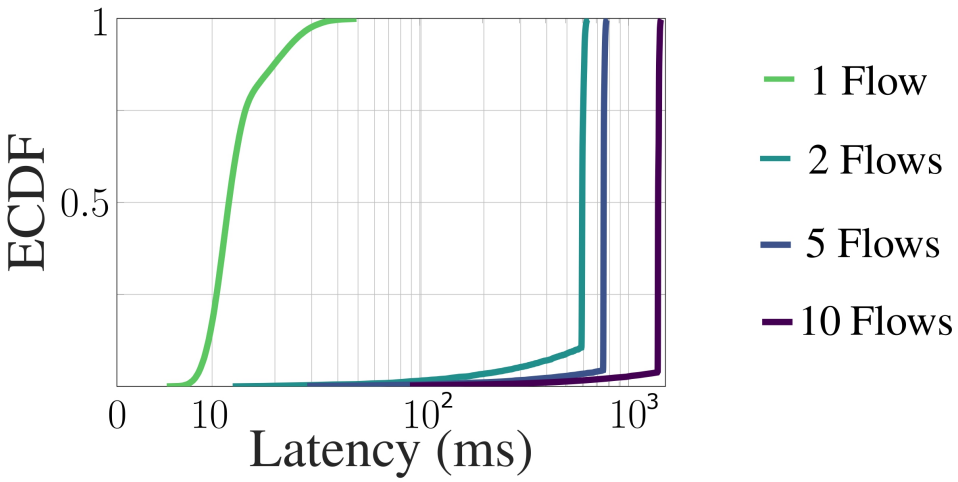
La interacción entre los dominios TSN y 3GPP garantiza que los comandos enviados al robot sean entregados de manera oportuna y confiable. Esto demuestra la viabilidad de las tecnologías de networking deterministas en aplicaciones críticas de la vida real. Aunque la imagen muestra la infraestructura de red y no el perro robot, esta configuración valida que el tráfico de red necesario para controlar el robot puede ser manejado eficientemente, asegurando así que el caso de uso del perro robot es alcanzable con nuestra implementación.

6.3. Análisis de Rendimiento y Resultados Obtenidos

En esta sección, se presenta un análisis detallado de los resultados obtenidos durante las pruebas de rendimiento del sistema multidominio. Las pruebas se realizaron con cargas de tráfico variables, que van desde 1 hasta 10 flujos de tráfico determinista definidos previamente, junto con tráfico de fondo aleatorio (BE). Esta configuración se utilizó para evaluar la robustez de la capacidad de la red para manejar variaciones significativas en la carga de tráfico mientras se priorizan las comunicaciones sensibles al tiempo, cruciales para la operación del perro robot en un entorno industrial en tiempo real.



(a) Retardo E2E para tráfico determinista



(b) Retardo E2E para tráfico BE

Fig. 6.3. Distribución del retardo por paquete de tráfico en el escenario E2E.

La Figura 6.3a muestra la distribución de latencia por paquete a través de 1, 2, 5 y 10 flujos deterministas. Los resultados indican que todos los flujos de TSN se entregan dentro de los 20 ms a lo largo del E2E, lo cual es coherente con la suma de las latencias de dominio único. Sin embargo, el rendimiento E2E está limitado por la red 3GPP. Este cuello de botella afecta significativamente la latencia del tráfico BE, como se muestra en la Figura 6.3b. La latencia aumenta notablemente cuando los volúmenes de tráfico superan la capacidad de la red y llenan las colas, demostrando el impacto crítico del dominio 3GPP en el rendimiento general de la red.

Durante las pruebas, se observó lo siguiente:

- **Estabilidad de la Latencia Determinista:** los flujos de tráfico determinista mostraron una latencia estable y predecible, manteniéndose dentro del rango de 20 ms incluso con un aumento en el número de flujos.

- **Impacto del Tráfico BE:** a medida que se incrementa el tráfico BE, se observa un aumento significativo en la latencia, especialmente cuando la capacidad de la red 3GPP se ve sobrepasada.
- **Cuellos de Botella en el Dominio 3GPP:** el dominio 3GPP fue identificado como el principal limitador del rendimiento E2E, afectando tanto la latencia como el jitter del tráfico BE.

Estas observaciones subrayan la importancia de gestionar adecuadamente la capacidad y el tráfico en el dominio 3GPP para mantener el rendimiento deseado en aplicaciones críticas. La implementación de técnicas de shaping y adaptación de MCS ha demostrado ser efectiva para mitigar algunos de estos problemas, garantizando que el tráfico determinista mantenga su calidad de servicio incluso en condiciones de alta carga.

La Figura 6.3 ilustra claramente cómo el tráfico BE experimenta latencias mucho mayores en comparación con el tráfico determinista, especialmente bajo condiciones de carga pesada. Esto destaca la necesidad de continuar optimizando los mecanismos de gestión de tráfico en redes multidominio para soportar aplicaciones sensibles al tiempo de manera eficiente y fiable.

7. IMPACTO SOCIO-ECONÓMICO

La implementación de soluciones multidominio para redes deterministas, como la desarrollada en este trabajo, tiene un impacto significativo en varios aspectos socioeconómicos. Este capítulo explora cómo estas innovaciones pueden transformar el mercado laboral, las consideraciones éticas y de privacidad, y la percepción pública de estas tecnologías.

7.1. Cambios en el Mercado Laboral

La integración de redes deterministas en sectores como la industria 4.0, la salud y el transporte autónomo tiene el potencial de transformar profundamente el mercado laboral. Las tecnologías que garantizan baja latencia y alta fiabilidad permiten la automatización de procesos críticos, lo cual, a su vez, influye en la demanda de diferentes habilidades laborales:

Automatización y Reducción de Tareas Manuales

- La automatización de tareas repetitivas y peligrosas, facilitada por las redes deterministas, puede reducir la necesidad de mano de obra en ciertas áreas, como la manufactura y la logística.
- Por otro lado, se incrementará la demanda de habilidades técnicas avanzadas para diseñar, implementar y mantener estos sistemas automatizados.

Nuevas Oportunidades Laborales

- La implementación de redes avanzadas crea nuevas oportunidades en campos como la ingeniería de redes, la ciberseguridad y el análisis de datos.
- Se espera un aumento en la demanda de especialistas en TSN y 5G, así como ingenieros de software y hardware capaces de trabajar con tecnologías emergentes.

Capacitación y Reentrenamiento

- La transformación del mercado laboral requerirá programas de reentrenamiento y capacitación continua para la fuerza laboral actual, adaptándose a las nuevas tecnologías y herramientas.
- Instituciones educativas y empresas deberán colaborar para desarrollar currículos que preparen a los trabajadores para los desafíos tecnológicos del futuro.

7.2. Ética y Privacidad de Datos

El despliegue de tecnologías de redes deterministas plantea importantes consideraciones éticas y de privacidad, especialmente en aplicaciones críticas como la atención médica y los vehículos autónomos.

Privacidad de Datos

- La recopilación y transmisión de grandes cantidades de datos sensibles en tiempo real implica riesgos significativos de privacidad. Es esencial implementar medidas de seguridad robustas para proteger la información personal y confidencial.
- Las políticas de gestión de datos deben ser transparentes y cumplir con las normativas de protección de datos, como el GDPR en Europa.

Ética en la Automatización

- La automatización de procesos puede llevar a la toma de decisiones autónomas por parte de máquinas, lo que plantea preguntas éticas sobre la responsabilidad y la equidad.
- Es crucial desarrollar marcos éticos que guíen el diseño y la implementación de sistemas autónomos, asegurando que las decisiones sean justas y responsables.

Transparencia y Consentimiento

- Los usuarios deben ser informados sobre cómo se recopilan y utilizan sus datos, y deben tener la opción de dar su consentimiento informado.
- Las organizaciones deben ser transparentes acerca de sus prácticas de datos y permitir a los usuarios controlar su información personal.

7.3. Impacto en la Percepción Pública

La percepción pública de las tecnologías avanzadas, como las redes deterministas y los sistemas autónomos, puede influir en su adopción y éxito.

Confianza en la Tecnología

- La fiabilidad y seguridad demostrada de las redes deterministas pueden aumentar la confianza del público en aplicaciones críticas, como la cirugía robótica y los vehículos autónomos.

- Sin embargo, cualquier fallo en estos sistemas puede erosionar rápidamente la confianza pública, subrayando la necesidad de rigurosos estándares de seguridad y calidad.

Aceptación Social

- La percepción de que la automatización puede llevar a la pérdida de empleos puede generar resistencia a la adopción de nuevas tecnologías. Es fundamental comunicar los beneficios de estas innovaciones, como la creación de nuevos tipos de empleos y la mejora de la eficiencia y seguridad.
- La educación y la sensibilización sobre cómo las tecnologías pueden mejorar la calidad de vida y crear oportunidades económicas son cruciales para la aceptación pública.

Responsabilidad y Transparencia

- Las empresas y los desarrolladores de tecnologías deben ser responsables y transparentes en sus operaciones. Esto incluye la divulgación de cómo funcionan los sistemas y cómo se mitigan los riesgos.
- La colaboración con reguladores y la adherencia a las normativas existentes ayudarán a construir y mantener la confianza pública.

En resumen, el impacto socioeconómico de las redes deterministas es amplio y multifacético. Mientras que estas tecnologías ofrecen numerosas ventajas y oportunidades, también plantean desafíos significativos que deben abordarse cuidadosamente. La clave para maximizar los beneficios y minimizar los riesgos reside en un enfoque equilibrado que incluya la innovación tecnológica, la capacitación laboral, la protección de datos y la transparencia ética.

8. PLANIFICACIÓN TEMPORAL Y PRESUPUESTO

En este capítulo, se presenta la planificación detallada seguida para la realización de este Trabajo de Fin de Máster, así como el presupuesto estimado correspondiente.

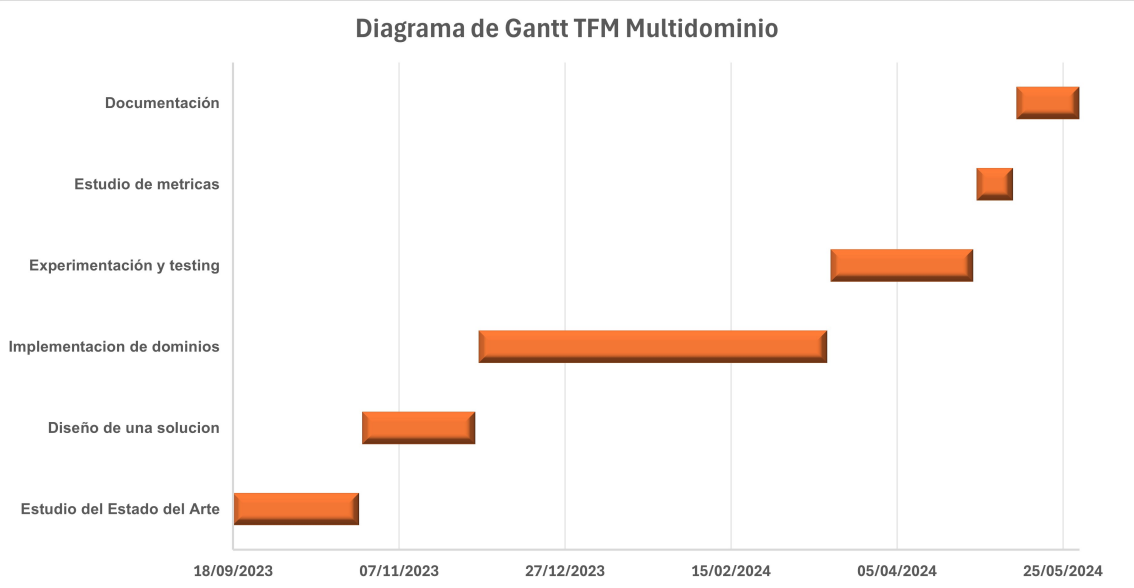
8.1. Planificación de Tareas del Proyecto

La ejecución del proyecto se dividió en varias etapas clave, siendo estas:

- **Estudio del Estado del Arte:** en esta fase inicial, se definieron claramente los objetivos del proyecto. Se realizó una revisión exhaustiva del estado actual de la investigación en el ámbito de las redes deterministas, analizando los avances y estudios más recientes tanto en el área de 3GPP como en TSN. También se recopilaron los requisitos necesarios para el desarrollo del sistema, teniendo en cuenta las limitaciones técnicas.
- **Diseño de una solución:** una vez establecidos los objetivos, se trabajó en el desarrollo de la arquitectura del sistema y se realizó un diseño preliminar del software, definiendo las principales funcionalidades y la estructura general del prototipo. Además, se configuraron todos los dispositivos involucrados y se creó el escenario de red necesario para la comunicación determinista.
- **Implementación de dominios:** en esta etapa, se llevó a cabo la implementación del software siguiendo el diseño establecido, y se realizó la integración con las tecnologías de TSN y 3GPP. También se realizaron pruebas iniciales de integración para verificar la comunicación y la interacción entre los distintos componentes del sistema.
- **Experimentación y testing:** se realizaron pruebas del sistema utilizando diversos casos de estudio para evaluar su rendimiento y comportamiento bajo diferentes condiciones de red. Posteriormente, se analizaron los resultados obtenidos y, con base en ellos, se identificaron los KPIs (Key Performance Indicators) de la aplicación desarrollada.
- **Estudio de métricas:** se extrajeron conclusiones finales del proyecto, contrastándolas con los objetivos iniciales, y se analizó el desempeño del sistema, incluyendo la interacción entre los diferentes dominios tecnológicos y la eficacia de la integración. Además, se sugirieron posibles líneas de investigación para el futuro en este ámbito.

- **Documentación:** la última etapa consistió en la redacción y revisión de la memoria del TFM. Esto incluyó la elaboración de la documentación técnica detallada del sistema desarrollado, así como la preparación de la presentación final del proyecto.

A continuación, se detalla en el diagrama de Gantt mostrado en la Figura 8.1a las fases previamente indicadas, así como sus tiempos, mientras que en la Figura 8.1b encontramos la gráfica de esfuerzo por fase.



(a) Diagrama de Gantt del TFM

(b) Gráfica de esfuerzo por fase

Fig. 8.1. Planificación temporal y distribución del esfuerzo del TFM Multidominio

8.2. Presupuesto de Elaboración

El presupuesto se ha desarrollado a partir de una estimación de los recursos necesarios para la realización del proyecto. Los costos solo incluyen los recursos de hardware, ya que todo el software utilizado es de código abierto y, por lo tanto, no genera costos adicionales. Los costos asociados al hardware se especifican en la Tabla 8.1, incluyendo todos los componentes físicos necesarios para la puesta en marcha del proyecto.

Concepto	Costo
USRP N310	17.407,00 €
DELL PowerEdge T550	4.652,00 €
Samsung Galaxy A33 5G	220,40 €
Servidor DELL Power Edge R7615	11.781,94 €
RELY-TSN-4 x 4	7.200,00 €
RELY-TSN-LAB	1.500,00 €
RELY-TRAF-GEN	990,00 €
PC Detnet x 3	1.800,00 €
MiniPCs Endpoints	800,00 €
Ordenador portátil Alienware M16	2.166,00 €
	46.351,34 €

TABLA 8.1. ESTIMACIÓN DE COSTES

Como se puede observar, el costo total del material físico necesario para la ejecución del proyecto asciende a **46.351,34 €**. Este presupuesto cubre todos los componentes necesarios para llevar a cabo las pruebas y la implementación del sistema multidominio propuesto en el proyecto, asegurando que se dispone de todo el equipamiento necesario para cumplir con los objetivos planteados.

A lo anterior, hay que sumarle el sueldo de un ingeniero de telecomunicaciones durante el periodo de ejecución del proyecto. Según Glassdoor [58], el salario anual de un ingeniero de telecomunicaciones oscila entre 24.000 € y 38.000 €. Calculando la media de estos valores, obtenemos un salario anual de 31.000 €. Dado que la duración del proyecto es de 250 días, se calcula el costo adicional correspondiente.

$$\text{Salario anual promedio} = \frac{24{,}000, + 38{,}000,}{2} = 31{,}000, \tag{8.1}$$

$$\text{Duración del proyecto} = 250, \text{días} \approx \frac{250}{365}, \text{años} = 0,685, \text{años} \tag{8.2}$$

$$\text{Costo del ingeniero} = 31{,}000, \times 0,685 = 21{,}235, \tag{8.3}$$

Por lo tanto, el costo añadido por la contratación de un ingeniero de telecomunicaciones durante el tiempo de ejecución del proyecto asciende a **21.235 €**.

Finalmente, el costo total del proyecto, incluyendo tanto el hardware como el salario del ingeniero, es:

$$\text{Costo total} = 46{,}351,34, + 21{,}235, = 67{,}586,34, \tag{8.4}$$

Concepto	Costo
Hardware	46.351,34 €
Salario del ingeniero	21.235,00 €
Total	67.586,34 €

TABLA 8.2. ESTIMACIÓN DE COSTES DE HARDWARE Y SALARIO DEL INGENIERO

9. CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se presentan las conclusiones derivadas del estudio y desarrollo de una solución multidominio para redes deterministas, así como las posibles direcciones futuras de investigación que podrían ampliar y mejorar los resultados obtenidos.

9.1. Síntesis de Resultados Clave

La implementación de una solución multidominio para redes deterministas ha demostrado ser efectiva en la integración de diferentes tecnologías de red, como IEEE 802.1 TSN y 3GPP, bajo una capa de DetNet. Los resultados obtenidos destacan varios puntos clave que merecen ser resaltados.

En primer lugar, la eficiencia de la integración multidominio se ha visto claramente demostrada. La solución propuesta ha permitido la coexistencia y cooperación sin fisuras entre dominios tecnológicos diferentes, garantizando el cumplimiento de los requisitos de latencia y fiabilidad en aplicaciones críticas. Este logro subraya la capacidad de la arquitectura desarrollada para soportar las demandas de tráfico determinista a través de múltiples dominios de red.

El desempeño del tráfico determinista fue otro aspecto crucial evaluado. Los experimentos realizados confirmaron que el tráfico determinista se mantiene dentro de los parámetros de latencia esperados, incluso bajo condiciones de alta carga de tráfico de mejor esfuerzo (BE). Esta robustez es fundamental para aplicaciones donde la puntualidad y la fiabilidad son esenciales.

Asimismo, la optimización del Modulation and Coding Scheme (MCS) en el dominio 3GPP y el uso de técnicas de traffic shaping han sido esenciales para reducir las retransmisiones y asegurar transmisiones sin errores en el enlace de radio. Estas adaptaciones han permitido que las comunicaciones críticas mantengan su integridad y calidad, en un dominio donde el trabajo en esta dirección es state of the art.

Finalmente, la validación en un entorno de prueba real E2E ha sido crucial para demostrar la viabilidad de la solución. La implementación mostró que es posible soportar aplicaciones como el control autónomo de un perro robot en un entorno industrial, validando así la aplicabilidad práctica de la arquitectura desarrollada.

9.2. Limitaciones y Desafíos

A pesar de los avances logrados, el estudio también ha identificado varias limitaciones y desafíos que deben ser abordados en futuras investigaciones. Uno de los desafíos más

destacados es la interferencia en redes 3GPP. Aunque se logró un buen desempeño en condiciones controladas, la interferencia en entornos 3GPP compartidos sigue siendo un obstáculo para mantener el determinismo.

La escalabilidad de la solución también requiere una evaluación más exhaustiva. Si bien la arquitectura ha mostrado ser efectiva en un entorno controlado, su capacidad para manejar un mayor número de dispositivos y tráfico en redes más grandes y complejas aún necesita ser probada, aunque la expectativa es positiva.

Otra área que requiere más desarrollo es la implementación de DetNet, donde una implementación y testeo de funcionalidades clave como es PREOF podría dar lugar a escenarios más interesantes, como lo son una replicación entre dos dominios, etc.

La gestión de la seguridad y la privacidad de los datos es crítica, especialmente en aplicaciones sensibles como la salud y la automatización industrial. La protección de los datos personales y confidenciales debe ser prioritaria, desarrollando medidas de seguridad robustas y cumpliendo con las normativas de protección de datos vigentes. Esto, al quedar fuera del scope de la línea de investigación de este trabajo, se deja abierto a trabajo futuro.

9.3. Direcciones Futuras de Investigación

Para seguir avanzando en el desarrollo de redes deterministas multidominio, se proponen varias direcciones futuras de investigación. Una de las áreas más prometedoras es la mejora de la interoperabilidad entre diferentes estándares y tecnologías de red. Desarrollar métodos que faciliten esta interoperabilidad será crucial para la expansión y adopción de soluciones multidominio.

Otra dirección importante es la optimización de algoritmos de control de tráfico que puedan adaptarse dinámicamente a las condiciones cambiantes de la red. Estos algoritmos deben ser capaces de gestionar eficientemente los recursos de la red, asegurando la calidad del servicio y minimizando la latencia.

El desarrollo de herramientas avanzadas de monitorización y diagnóstico es también una prioridad. Estas herramientas permitirán asegurar la calidad del servicio, detectar problemas en tiempo real y proporcionar soluciones rápidas y efectivas.

Ampliar la evaluación de la solución en escenarios del mundo real con diferentes tipos de tráfico y condiciones de red es fundamental para validar su robustez y fiabilidad. Esta evaluación permitirá identificar posibles mejoras y ajustar la arquitectura para satisfacer mejor las necesidades del entorno real.

Finalmente, y como se menciona en la sección anterior, la integración de medidas de seguridad robustas directamente en la arquitectura de las redes deterministas es esencial. Investigar métodos para asegurar la protección de datos sin comprometer el rendimiento será clave para la adopción generalizada de estas tecnologías.

En conclusión, este trabajo ha demostrado la viabilidad y las ventajas de una solu-

ción multidominio para redes deterministas, proporcionando una base sólida para futuras investigaciones y desarrollos en este campo emergente.

BIBLIOGRAFÍA

[1] M. Li, Q. Tan, Z. Li, K. Zhao, W. Li e Y.-k. Fang, "Topology Planning of Space Deterministic Networks," *2023 6th World Symposium on Communication Engineering (WSCE)*, pp. 68-73, 2023. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:266558857.

[2] L. Barrière, F. Comellas, C. Dalfó y M. A. Fiol, "Deterministic hierarchical networks," *Journal of Physics A: Mathematical and Theoretical*, vol. 49, 2015. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:14511794.

[3] P. R. Kshirsagar, D. H. Reddy, M. Dhingra, D. Dhabliya y A. Gupta, "A Review on Comparative study of 4G, 5G and 6G Networks," en *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, IEEE, 2022, pp. 1830-1833.

[4] F. Comellas y M. Sampels, "Deterministic small-world networks," *Physica A-statistical Mechanics and Its Applications*, vol. 309, pp. 231-235, 2001. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:119370397.

[5] A. H. Madsen, "Deterministic Capacity of Networks," *2007 IEEE Information Theory Workshop*, pp. 601-606, 2007. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:8963885.

[6] E. Hajlaoui, A. Zaier, A. Khlifi, J. Ghodhbane, M. B. Hamed y L. Sbita, "4G and 5G technologies: A Comparative Study," en *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2020, pp. 1-6. DOI: 10.1109/ATSIP49331.2020.9231605.

[7] R.-A. Koutsiamanis, G. Z. Papadopoulos, X. Fafoutis, J. M. D. Fiore, P. Thubert y N. Montavont, "From Best Effort to Deterministic Packet Delivery for Wireless Industrial IoT Networks," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 4468-4480, 2018. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:52925962.

[8] S. H. Lim, S.-Y. Chung e Y.-H. Kim, "Deterministic relay networks with state information," *2009 IEEE International Symposium on Information Theory*, pp. 36-40, 2009. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:1486312.

[9] F. Song, L. Li, I. You y H. Zhang, "Enabling Heterogeneous Deterministic Networks with Smart Collaborative Theory," *IEEE Network*, vol. 35, pp. 64-71, 2021. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:235455791.

[10] J.-F. Tsai, M.-H. Lin y P.-C. Wang, "An Efficient Deterministic Approach to Optimal Design of Reliable Networks," *IEEE Transactions on Reliability*, vol. 67, pp. 598-608, 2018. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:44184974.

[11] L. Silva, P. Pedreiras, P. Fonseca y L. Almeida, "On the adequacy of SDN and TSN for Industry 4.0," en *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)*, 2019, pp. 43-51. DOI: 10.1109/ISORC.2019.00017.

[12] J. Lee y S. Park, "Time-Sensitive Network (TSN) Experiment in Sensor-Based Integrated Environment for Autonomous Driving," *Sensors*, vol. 19, n.º 5, 2019. DOI: 10.3390/s19051111. [En línea]. Disponible en: https://www.mdpi.com/1424-8220/19/5/1111.

[13] B. Rother, M. Kasparick, E. Schweißguth, F. Golatowski y D. Timmermann, "Automatic Configuration of a TSN Network for SDC-based Medical Device Networks," en *2020 16th IEEE International Conference on Factory Communication Systems (WFCS)*, 2020, pp. 1-8. DOI: 10.1109/WFCS47810.2020.9114471.

[14] J. Sanchez-Garrido, A. Jurado, L. Medina, R. Rodriguez, E. Ros y J. Diaz, "Digital Electrical Substation Communications Based on Deterministic Time-Sensitive Networking Over Ethernet," *IEEE Access*, vol. 8, pp. 93 621-93 634, 2020. DOI: 10.1109/ACCESS.2020.2995189.

[15] H. Trifonov y D. Heffernan, "OPC UA TSN: a next-generation network for Industry 4.0 and IIoT," *International Journal of Pervasive Computing and Communications*, vol. 19, n.º 3, pp. 386-411, 2023.

[16] C. Mannweiler et al., "Reliable and Deterministic Mobile Communications for Industry 4.0: Key Challenges and Solutions for the Integration of the 3GPP 5G System with IEEE," en *Mobile Communication - Technologies and Applications; 24. ITG-Symposium*, 2019, pp. 1-6.

[17] M.-T. Thi et al., "IEEE 802.1 TSN Time Synchronization over Wi-Fi and 5G Mobile Networks," *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pp. 1-7, 2022. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:255995503.

[18] D. Ergenç, R. Schenderlein y M. Fischer, "TSNZeek: An Open-source Intrusion Detection System for IEEE 802.1 Time-sensitive Networking," *2023 IFIP Networking Conference (IFIP Networking)*, pp. 1-6, 2023. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:257636790.

[19] A. Nasrallah et al., "Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 88-145, 2018. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:52169298.

[20] P. Pop, M. Raagaard, M. Gutiérrez y W. Steiner, "Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN)," *IEEE Communications Standards Magazine*, vol. 2, pp. 55-61, 2018. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:49888418.

[21] J. Falk et al., "NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++," *2019 International Conference on Networked Systems (NetSys)*, pp. 1-8, 2019. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:203655317.

[22] D. Ergenç, C. Brülhart, J. Neumann, L. Krüger y M. Fischer, "On the Security of IEEE 802.1 Time-Sensitive Networking," *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1-6, 2021. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:235813977.

[23] S. Avallone, P. Imputato y D. Magrin, "Controlled Channel Access for IEEE 802.11-Based Wireless TSN Networks," *IEEE Internet of Things Magazine*, vol. 6, pp. 90-95, 2023. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:257536157.

[24] "IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks," *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)*, pp. 1-1993, 2018. DOI: 10.1109/IEEESTD.2018.8403927.

[25] "IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications," *IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011)*, pp. 1-421, 2020. DOI: 10.1109/IEEESTD.2020.9121845.

[26] "IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic," *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)*, pp. 1-57, 2016. DOI: 10.1109/IEEESTD.2016.8613095.

[27] "IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability," *IEEE Std 802.1CB-2017*, pp. 1-102, 2017. DOI: 10.1109/IEEESTD.2017.8091139.

[28] "IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 28: Per-Stream Filtering and Policing," *IEEE Std 802.1Qci-2017 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv-2015, IEEE Std 802.1Qbu-2016, and IEEE Std 802.1Qbz-2016)*, pp. 1-65, 2017. DOI: 10.1109/IEEESTD.2017.8064221.

[29] L. Kong, P. Wu, J. Chu, W. Xu y L. Zhou, "A Deterministic Communication Technique in the 5G-Adv/6G Access Network Systems," en *2022 14th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2022. DOI: 10.1109/WCSP55476.2022.10039118. [En línea]. Disponible en: https://dx.doi.org/10.1109/WCSP55476.2022.10039118.

[30] M. C. Lucas-Estañ, A. Larrañaga, J. Gozalvez e I. Martinez, "Configured Grant Scheduling for the Support of TSN Traffic in 5G and Beyond Industrial Networks," en *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 2023. DOI: 10.1109/VTC2023-Fall60731.2023.10333523. [En línea]. Disponible en: https://dx.doi.org/10.1109/VTC2023-Fall60731.2023.10333523.

[31] J. Prados-Garzon y T. Taleb, "Asynchronous Time-Sensitive Networking for 5G Backhauling," *IEEE Network*, 2021. DOI: 10.1109/MNET.011.2000402. [En línea]. Disponible en: https://dx.doi.org/10.1109/MNET.011.2000402.

[32] R. Debnath, M. Akinci, D. Ajith y S. Steinhorst, "5GTQ: QoS-Aware 5G-TSN Simulation Framework," en *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 2023. DOI: 10.1109/VTC2023-Fall60731.2023.10333533. [En línea]. Disponible en: https://dx.doi.org/10.1109/VTC2023-Fall60731.2023.10333533.

[33] D. Krummacker, B. Veith, C. Fischer y H. Schotten, "Analysis of 5G Channel Access for Collaboration with TSN Concluding at a 5G Scheduling Mechanism," *Network*, vol. 2, n.º 3, 2022. DOI: 10.3390/network2030027. [En línea]. Disponible en: https://dx.doi.org/10.3390/network2030027.

[34] A. Larrañaga, M. C. Lucas-Estañ, I. Martinez, I. Val y J. Gozálvez, "Analysis of 5G-TSN Integration to Support Industry 4.0," en *2020 IEEE 25th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020. DOI: 10.1109/ETFA46521.2020.9212141. [En línea]. Disponible en: http://dspace.umh.es/bitstream/11000/7119/1/1-ETFA_Lucas_5G_TSN_web.pdf.

[35] Y. Zhu, L. Sun, J. Wang, R. Huang y X. Jia, "Deep Reinforcement Learning-Based Joint Scheduling of 5G and TSN in Industrial Networks," *Electronics*, vol. 12, n.º 12, 2023. DOI: 10.3390/electronics12122686. [En línea]. Disponible en: https://www.mdpi.com/2079-9292/12/12/2686.

[36] IETF, *DETNET WORKING GROUP*. [En línea]. Disponible en: https://datatracker.ietf.org/wg/detnet/about/.

[37] IETF, *Home*. [En línea]. Disponible en: https://www.ietf.org/.

[38] N. Finn, P. Thubert, B. Varga y J. Farkas, *Deterministic Networking Architecture*, RFC 8655, oct. de 2019. DOI: 10.17487/RFC8655. [En línea]. Disponible en: https://www.rfc-editor.org/info/rfc8655.

[39] B. Varga, J. Farkas, L. Berger, A. G. Malis, S. Bryant y J. Korhonen, *Deterministic Networking (DetNet) Data Plane: MPLS*, RFC 8964, ene. de 2021. DOI: 10.17487/RFC8964. [En línea]. Disponible en: https://www.rfc-editor.org/info/rfc8964.

[40] B. Varga, J. Farkas y A. G. Malis, *Deterministic Networking (DetNet) Packet Replication, Elimination, and Ordering Functions (PREOF) via MPLS over UDP/IP*, RFC 9566, abr. de 2024. DOI: 10.17487/RFC9566. [En línea]. Disponible en: https://www.rfc-editor.org/info/rfc9566.

[41] J.-Y. Le Boudec, "Modeling and Control of a Network of Open Systems with Application to Real-Time Networks," *IEEE Transactions on Networking*, vol. 6, n.º 4, pp. 485-498, 1998. DOI: 10.1109/90.7095804. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/7095804.

[42] L. Zhang y L. Thiele, "Scheduling for Fault-Tolerant Communication on Real-Time Ethernet Networks," en *Proceedings of the IEEE Real-Time Systems Symposium*, IEEE, 2015, pp. 125-134. DOI: 10.1109/RTSS.2015.7113243. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/7113243.

[43] L. Ferranti y J. Van Der Veen, "Design and Implementation of Deterministic Network Solutions for Industrial Robotic Systems," en *2019 IEEE International Conference on Robotic Automation*, IEEE, 2019, pp. 4210-4216. DOI: 10.1109/ICRA.2019.8768549. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/8768549.

[44] E. Ezhilarasan y M. Dinakaran, "A review on mobile technologies: 3G, 4G and 5G," en *2017 second international conference on recent trends and challenges in computational models (ICRTCCM)*, IEEE, 2017, pp. 369-373.

[45] R. Zeqiri, F. Idrizi y H. Halimi, "Comparison of Algorithms and Technologies 2G, 3G, 4G and 5G," en *2019 3rd International symposium on Multidisciplinary studies and Innovative Technologies (ISMSIT)*, IEEE, 2019, pp. 1-4.

[46] A. Adamowicz, *5G non-stand alone vs. 5G stand alone: Esta es la diferencia*, ene. de 2023. [En línea]. Disponible en: https://www.gsma.com/latinamerica/es/5g-non-stand-alone-vs-5g-stand-alone-esta-es-la-diferencia/.

[47] P. Picazo Martínez, "Despliegue de redes 5G Non-Standaone basadas en OpenAirInterface.," Tesis doct., Universitat Politècnica de València, 2021.

[48] Relyum, *RELY-TSN-BRIDGE*, Accessed: 2024-06-13, 2024. [En línea]. Disponible en: https://www.relyum.com/web/rely-tsn-bridge/.

[49] RealTime at Work, *RTAW PEGASE*, Accessed: 2024-06-13, 2024. [En línea]. Disponible en: https://www.realtimeatwork.com/rtaw-pegase/.

[50] OpenAirInterface, *OpenAirInterface*, Accessed: 2024-06-13, 2024. [En línea]. Disponible en: https://openairinterface.org/.

[51] OpenAirInterface CN5G, *OpenAirInterface 5G Core Network (CN5G)*, Accessed: 2024-06-13, 2024. [En línea]. Disponible en: https://gitlab.eurecom.fr/oai/cn5g.

[52] OpenAirInterface 5G, *OpenAirInterface 5G RAN*, Accessed: 2024-06-13, 2024. [En línea]. Disponible en: https://gitlab.eurecom.fr/oai/openairinterface5g/.

[53] J. Korhonen et al., "DetNet Data Plane Encapsulation," Internet Engineering Task Force, Internet-Draft draft-dt-detnet-dp-sol-02, sep. de 2017, Work in Progress, 36 págs. [En línea]. Disponible en: https://datatracker.ietf.org/doc/draft-dt-detnet-dp-sol/02/.

[54] B. Varga, J. Farkas, L. Berger, A. G. Malis y S. Bryant, *Deterministic Networking (DetNet) Data Plane: MPLS over UDP/IP*, RFC 9025, abr. de 2021. DOI: 10.17487/RFC9025. [En línea]. Disponible en: https://www.rfc-editor.org/info/rfc9025.

[55] XDP Project, *XDP Project on GitHub*, Accessed: 2024-06-13, 2024. [En línea]. Disponible en: https://github.com/xdp-project.

[56] eBPF, *eBPF - Extended Berkeley Packet Filter*, Accessed: 2024-06-13, 2024. [En línea]. Disponible en: https://ebpf.io/.

[57] Kernel.org, *AF_XDP – LinuxKernelDocumentation*, Accessed: 2024-06-13, 2024. [En línea]. Disponible en: https://www.kernel.org/doc/html/next/networking/af_xdp.html.

[58] Glassdoor, *Sueldo: Ingeniero de Telecomunicaciones (Junio, 2023) | glassdoor*. [En línea]. Disponible en: https://www.glassdoor.es/Sueldos/ingeniero-de-telecomunicaciones-sueldo-SRCH_KO0,31.htm.