

La investigación reportada en esta tesis es parte de los programas de investigación del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California).

La investigación fue financiada por el CONAHCYT (Consejo Nacional de Humanidades, Ciencias y Tecnologías).

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México). El uso de imágenes, fragmentos de videos y material protegido de los derechos de autor será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo o titular de los Derechos de Autor.

CICESE© 2024. Todos los derechos reservados

**Centro de Investigación Científica y de Educación
Superior de Ensenada, Baja California**



**Maestría en Ciencias
en Electrónica y Telecomunicaciones
con orientación en Telecomunicaciones**

**Sistema de gestión de monitorización de violaciones de
contratos de niveles de servicio (SLA, *Service Level Agreement*)
en entorno de redes 5G/B5G**

Tesis
para cubrir parcialmente los requisitos necesarios para obtener el grado de
Maestro en Ciencias

Presenta:

Yenia Leyva Labrador

Ensenada, Baja California, México
2024

Tesis defendida por
Yenia Leyva Labrador
y aprobada por el siguiente Comité

Dr. Jorge Enrique Preciado Velasco
Codirector de tesis

Dr. José Eleno Lozano Rizk
Codirector de tesis

Dr. Jorge Torres Rodríguez

Dr. Raúl Rivera Rodríguez

Dr. Miguel Ángel Alonso Arévalo



Dra. María del Carmen Maya Sánchez
Coordinadora del Posgrado en Electrónica y Telecomunicaciones

Dra. Ana Denise Re Araujo
Directora de Estudios de Posgrado

Resumen de la tesis que presenta **Yenia Leyva Labrador** como requisito parcial para la obtención del grado de Maestro en Ciencias en Electrónica y Telecomunicaciones con orientación en Telecomunicaciones.

Sistema de gestión de monitorización de violaciones de contratos de niveles de servicio (SLA, Service Level Agreement) en entorno de redes 5G/B5G

Resumen aprobado por:

Dr. Jorge Enrique Preciado Velasco
Codirector de tesis

Dr. José Eleno Lozano Rizk
Codirector de tesis

Esta investigación aborda el desafío de garantizar niveles de servicio (SLA) en redes 5G/B5G, caracterizadas por tráfico dinámico y heterogéneo. Se propone un sistema innovador para la gestión y monitorización predictiva de SLA, que integra monitorización en tiempo real, redireccionamiento dinámico de flujos y análisis predictivo mediante *Machine Learning* en entornos de redes definidas por software (SDN). El sistema opera detectando incumplimientos de métricas de SLA, implementando estrategias de redireccionamiento de tráfico que priorizan la baja latencia y adaptando proactivamente los recursos de red para optimizar el cumplimiento de los SLA. La arquitectura propuesta aprovecha las capacidades de programabilidad, control centralizado y abstracción de recursos que ofrecen las SDN, junto con la flexibilidad y escalabilidad de Mininet para simular entornos de red complejos. Incorpora un sistema de monitorización para verificar el cumplimiento de los SLA y una API RESTful para facilitar el redireccionamiento de tráfico de manera ágil y estandarizada. Lo más destacado es la integración de técnicas de *Machine Learning*, que permiten al sistema aprender de los patrones de tráfico y comportamiento de la red, anticipar posibles violaciones de SLA y redireccionar dinámicamente la asignación de recursos para cumplir con los requisitos de calidad de servicio (QoS) acordados. Los resultados preliminares sugieren una mejora significativa en la gestión de SLA comparado con enfoques tradicionales, especialmente en escenarios de alta variabilidad de tráfico. Aunque se requiere una implementación y evaluación exhaustiva en entornos reales, la propuesta representa una solución prometedora para los desafíos de aseguramiento de calidad de servicio en redes de nueva generación. Este enfoque integrado y predictivo ofrece ventajas sobre métodos reactivos convencionales, contribuyendo al campo de la gestión de redes 5G/B5G con un marco innovador para la optimización dinámica de SLA. Se espera que el sistema propuesto facilite la provisión de servicios de alta calidad en futuras redes móviles, caracterizadas por demandas de tráfico cada vez más complejas y diversas.

Palabras clave: Redes móviles 5G/B5G, Monitorización de SLA, Redes SDN, Aprendizaje de Máquina.

Abstract of the thesis presented by **Yenia Leyva Labrador** as a partial requirement to obtain the Master of Science degree in Electronics and Telecommunications with orientation in Telecommunications.

SLA (Service Level Agreement) violation management of monitoring system in 5G/B5G network environments

Abstract approved by:

Dr. Jorge Enrique Preciado Velasco
Thesis Co-director

Dr. José Eleno Lozano Rizk
Thesis Co-director

This research addresses the challenge of ensuring Service Level Agreements (SLAs) in 5G/B5G networks characterized by dynamic and heterogeneous traffic. We proposed an innovative system for predictive SLA management and monitoring, integrating real-time monitoring, dynamic flow redirection, and predictive analysis using Machine Learning in Software-Defined Networking (SDN) environments. The system detects SLA metric violations, implements traffic redirection strategies that prioritize low latency, and proactively adapts network resources to optimize SLA compliance. The proposed architecture leverages the programmability, centralized control, and resource abstraction capabilities offered by SDN, along with the flexibility and scalability of Mininet, to simulate complex network environments. It incorporates a monitoring system to verify SLA compliance and a RESTful API to facilitate agile and standardized traffic redirection. The highlight is the integration of Machine Learning techniques, which allow the system to learn from traffic patterns and network behavior, anticipate potential SLA violations, and dynamically redirect resource allocation to meet agreed Quality of Service (QoS) requirements. Preliminary results suggest a significant improvement in SLA management compared to traditional approaches, especially in scenarios with high traffic variability. Although comprehensive implementation and evaluation in existing 5G/B5G network environments are required, the proposal represents a promising solution for quality of service assurance challenges in next-generation networks. This integrated and predictive approach offers advantages over conventional reactive methods, contributing to the 5G/B5G network management with an innovative framework for dynamic SLA optimization. With the proposed system, we expect to facilitate high-quality services in future mobile networks characterized by increasingly complex and diverse traffic demands.

Keywords: 5G/B5G Mobile Networks, SLA Monitoring, SDN Networks, Machine Learning.

Dedicatoria

A Mi

A Tan

A Mi Familia

Agradecimientos

Quiero agradecer a todas esas personas que de una forma u otra hicieron posible este logro:

A mi Hijo, mi motor, mi sostén y mi vida entera.

A mis padres por su apoyo incondicional.

A Yan y Daylin por ser los protagonistas de este periodo.

A Lili por ser parte de mi familia.

A mis Cubanichis que han estado acompañándome en esta aventura.

A mis amigos que son un pilar fundamental en mi vida.

A mis tutores por su motivación y enseñanzas en el desarrollo de esta investigación.

A mi comité de tesis por sus observaciones y recomendaciones que contribuyeron a mejorar mi investigación.

A CICESE.

A CONACHYT.

A todos los profesores, compañeros y conocidos en esta experiencia.

A un ser muy especial, a quien tuve el honor de conocer, y cuyo apoyo, confianza y motivación fueron esenciales para que pudiera llegar hasta aquí.

Muchas Gracias....

Los llevaré por siempre en mi corazón.

Tabla de contenido

	Página
Resumen en español.....	ii
Resumen en inglés.....	iii
Dedicatoria.....	iv
Agradecimientos.....	v
Lista de figuras.....	ix
Lista de tablas.....	xi
Capítulo 1. Introducción.....	1
1.1 Planteamiento del problema.....	3
1.2 Antecedentes y justificación.....	4
1.3 Hipótesis.....	5
1.4 Objetivos.....	6
1.4.1 Objetivo general.....	6
1.4.2 Objetivos específicos.....	6
1.5 Estructura de la tesis.....	7
Capítulo 2. Marco teórico.....	9
2.1 Segmentación de la red.....	9
2.2 Acuerdos de Nivel de Servicio (SLA).....	10
2.2.1 Estructura de un SLA.....	11
2.2.2 Ciclo de vida de un SLA.....	12
2.3 Arquitectura Basada en Servicios (SBA).....	13
2.4 Virtualización de Funciones de Red (NFV).....	16
2.4.1 Arquitectura NFV.....	16

2.5	Redes Definidas por Software (SDN).....	18
2.5.1	Arquitectura SDN	19
2.6	API REST.....	21
2.7	Aprendizaje de máquina o <i>Machine Learning</i>	22
2.7.1	Clasificación tipos de <i>Machine Learning</i>	23
2.7.1.1	Aprendizaje Profundo o <i>Deep Learning</i>	23
2.7.2	Importancia del uso de <i>Machine Learning</i> en la predicción de SLA	24
2.8	Conclusiones parciales	26
Capítulo 3. Propuesta del sistema de gestión y monitorización de SLA		28
3.1	Esquema general de la propuesta.....	28
3.2	Desarrollo de la propuesta	30
3.2.1	Plataforma de simulación.....	30
3.2.2	Controlador SDN	32
3.2.2.1	Asignación de prioridades en ODL	34
3.3	Monitorización y enrutamiento	35
3.3.1	Parámetros de monitorización para evaluar el cumplimiento del SLA.....	38
3.3.2	Comportamiento del tráfico por aplicación	39
3.3.3	Generación de tráfico y obtención de métricas.....	41
3.3.4	API de enrutamiento	43
3.4	Integración de la monitorización y la API de enrutamiento	44
3.5	Predicción de incumplimientos de SLA mediante <i>Machine Learning</i>	47
3.6	Implementación en la nube y selección de proveedor de servicio	50
3.7	Conclusiones parciales	52
Capítulo 4. Validación de la propuesta		54
4.1	Escenario de validación	54

4.2	Topología de red propuesta	57
4.3	Simulaciones del sistema	59
4.3.1	Simulación y monitorización	60
4.3.1.1	Parámetros de generación de tráfico.....	65
4.3.2	Desarrollo de la API y configuración de flujos.....	68
4.3.3	Análisis de datos y predicción de SLA	71
4.4	Resultados obtenidos.....	73
4.4.1	Comportamiento del SLA	73
4.4.2	Comportamiento de las pérdidas de paquetes.....	75
4.4.3	Comportamiento de la latencia.....	76
4.4.4	Resultados de la predicción de SLA.....	77
4.5	Conclusiones parciales	79
Capítulo 5. Conclusiones y recomendaciones		81
5.1	Contribuciones al conocimiento	83
5.2	Limitaciones de la investigación.....	85
5.3	Trabajos futuros	85
Literatura citada		87
Anexos		93

Lista de figuras

Figura	Página
1. Escenarios de uso 5G. Tomada de (Recommendation ITU-R M.2083-0, 2015).	2
2. Segmentos de red. Tomada de (Jalalian et al., 2023).	10
3. Estructura de un SLA. Adaptada de (Ramos-García et al., 2021).	11
4. Ciclo de vida de un SLA. Adaptada de (Ramos-García et al., 2021).	12
5. Arquitectura SBA en 5G. Tomada de (3GPP Release, 2018).	14
6. Arquitectura NFV. Adaptada de (Juniper Networks, 2023).	17
7. Arquitectura SDN. Tomada de (L. Wang & Yang, 2018).	19
8. Funcionamiento de una API REST. Adaptada de (Gonzalez-Franco et al., 2023).	21
9. Diferencias entre <i>Machine Learning</i> y <i>Deep Learning</i> . Tomada de (Rafael Zambrano, 2019).	24
10. Esquema general propuesto.	29
11. Estrategia de monitorización.	36
12. Proceso de monitorización y ajuste dinámico de rutas para cumplimiento de SLA en redes 5G/B5G.	45
13. Redes Neuronales Recurrentes. Tomada de (León et al., 2022).	47
14. Modelo LSTM. Tomada de (Alom et al., 2019).	49
15. Topología de red propuesta.	58
16. Topología de la red en Mininet.	64
17. Imagen de Postman.	69
18. Porcentaje de cumplimiento de SLA.	74
19. Porcentaje de cumplimiento de SLA por aplicación.	74
20. Porcentaje de pérdida de paquetes.	75
21. Porcentaje de pérdidas de paquetes por aplicación.	76
22. Latencia promedio por tipo de medición.	76
23. Latencia promedio por aplicación.	77
24. Log de los resultados del Sistema de Monitorización.	93

25. Ejemplo de la Bitácora de las mediciones.....	94
26. Configuración regla del flujo en el controlador ODL.....	96
27. Reglas de Flujo en el switch s0.....	96
28. Código predicción con LSTM.	97

Lista de tablas

Tabla	Página
1. Listado de Servicios/Aplicaciones por nodo.....	61
2. Características de QoS de cada enlace de los switches al servicio o UE de la topología.	62
3. Características de QoS de cada enlace de los switches de la topología.	62
4. Parámetros de SLA por aplicación. Tomada de (González-Franco et al., 2023).	65
5. Prioridad de los flujos por aplicación.	69
6. Resultados de la monitorización.	73
7. Resultado de la predicción.	78
8. Porcentaje de SLA por nodo.....	95
9. Latencia promedio por nodo.....	95
10. Porcentaje de Pérdidas de paquetes por nodo.....	95

Capítulo 1. Introducción

A medida que el mundo se sumerge en una transformación digital, las redes móviles continúan siendo el epicentro de la innovación. La introducción de las redes móviles 5G/B5G (por las siglas de los términos en inglés, *5th Generation / Beyond 5G*) marca un punto de inflexión en la evolución tecnológica, con un impacto profundo en la sociedad. Estas redes se diseñaron para manejar velocidades ultrarrápidas, con baja latencia y una conectividad más confiable. Estos avances permiten una amplia gama de nuevos casos de uso, servicios, aplicaciones y escenarios.

La Unión Internacional de Telecomunicaciones (ITU, por las siglas del término en inglés, *International Telecommunications Union*) estableció los siguientes tres casos de uso genéricos para garantizar que las redes 5G/B5G puedan satisfacer las diversas demandas emergentes de conectividad inalámbrica (International Telecommunication Union, 2015).

- ✓ Banda Ancha Móvil Mejorada (eMBB, por las siglas del término en inglés, *Enhanced Mobile Broadband*): se enfoca en mejorar la velocidad, capacidad y cobertura de redes móviles para admitir aplicaciones avanzadas como realidad virtual, video *streaming* y juegos avanzados, ofreciendo velocidades gigabit, baja latencia y alta confiabilidad.
- ✓ Comunicación Ultra confiable y de Baja Latencia (urLLC, por las siglas del término en inglés, *Ultra-reliable and Low Latency Communication*): garantiza baja latencia y alta confiabilidad para aplicaciones críticas como la automatización industrial y vehículos autónomos.
- ✓ Comunicación Masiva Tipo Máquina (mMTC, por las siglas del término en inglés, *Massive Machine Type Communication*): aborda la conectividad masiva de dispositivos (IoT, por las siglas del término en inglés, *Internet of Things*), proporcionando conectividad para miles de millones de dispositivos con requisitos diversos.

Con el despliegue de estas capacidades innovadoras, emergen aplicaciones revolucionarias en múltiples sectores industriales, desde la agricultura de precisión, que facilita un monitoreo detallado de los cultivos, hasta la fabricación automatizada, que permite un análisis profundo de los procesos industriales (Docomo, 2020). Otros ámbitos beneficiados incluyen el comercio minorista con sistemas sin cajeros, la atención

médica remota, la optimización de la logística de flotas, la inteligencia artificial avanzada, la realidad virtual envolvente, los juegos en tiempo real y la computación en la nube distribuida. La baja latencia y la conectividad masiva de Internet de las cosas habilitan estos nuevos casos de uso, procesando datos más cerca de los dispositivos según se requiera y permitiendo la toma de decisiones casi en tiempo real. En la Figura 1 se ilustran las diferentes aplicaciones que se incluyen en 5G (Recommendation ITU-R M.2083-0, 2015).

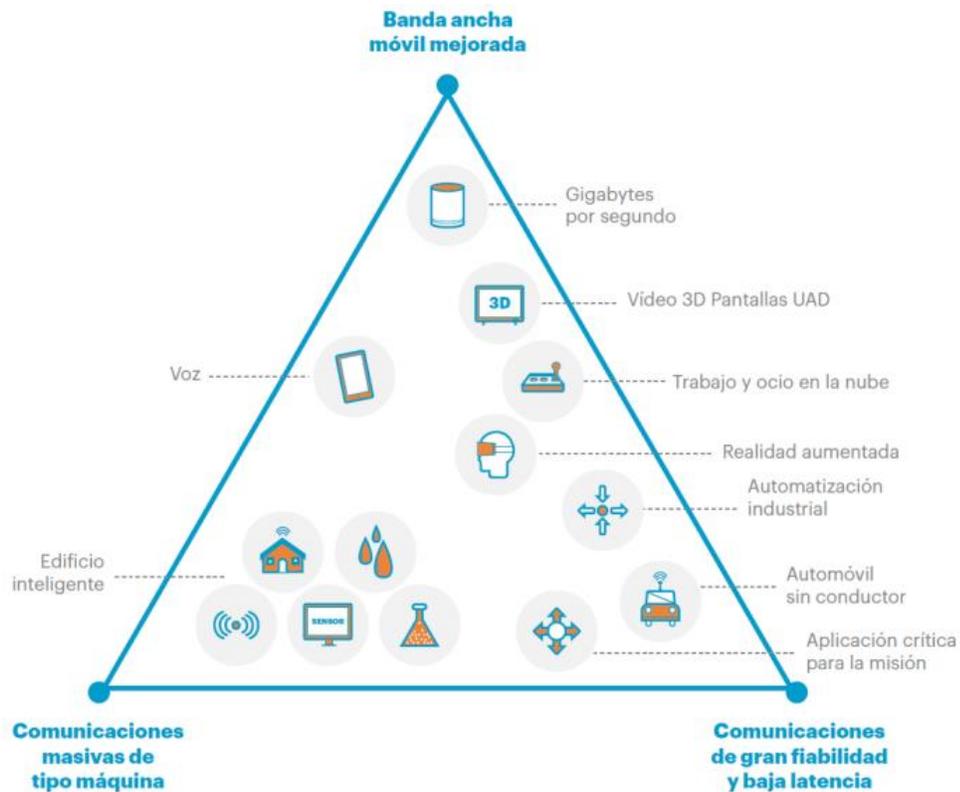


Figura 1. Escenarios de uso 5G. Tomada de (Recommendation ITU-R M.2083-0, 2015).

Dentro del ecosistema de las futuras redes móviles, la creación y gestión de aplicaciones y servicios requieren recursos de red cada vez más especializados y potentes para satisfacer demandas que evolucionan rápidamente. Esto plantea desafíos significativos, ya que cada aplicación o servicio puede tener requisitos específicos en términos de ancho de banda, latencia y fiabilidad. Por lo tanto, es crucial establecer una alineación óptima entre los requerimientos de los usuarios y las capacidades de los proveedores de recursos de red, lo que exige una adaptación continua y dinámica de la red a las necesidades cambiantes (5G Americas Whitepaper, 2017).

La creciente complejidad, el aumento constante del tráfico y la necesidad de flexibilidad en las redes 5G/B5G hacen que sea necesario automatizar la gestión de la red, ya que los métodos de gestión tradicionales no son suficientes para lidiar con la complejidad de los parámetros en estas redes avanzadas y su constante evolución. Por eso, se hace imperativo la utilización de técnicas avanzadas como la Inteligencia Artificial (AI, por las siglas del término en inglés, *Artificial Intelligence*), el Aprendizaje de Máquina (ML, por las siglas del término en inglés, *Machine Learning*), las Redes Definidas por Software (SDN, por las siglas del término en inglés, *Software Defined Networking*) y la Virtualización de Funciones de Red (NFV, por las siglas del término en inglés, *Network Functions Virtualization*) para gestionar eficazmente los servicios en tiempo real (Docomo, 2020) .

Por los motivos arriba mencionados, una de las estrategias introducidas en las redes 5G es implementar lo conocido como Segmentación de Red, que permite adaptar mediante la creación de múltiples enlaces lógicos (segmentos) punto a punto sobre una misma infraestructura física de la red y ofrecer atención especializada a diversas aplicaciones y servicios. Cada segmento de red (NS, por sus siglas en inglés *Network Slices*) tendrá asociado un Acuerdo de Nivel de Servicio (SLA, por las siglas del término en inglés, *Service Level Agreement*) específico, adecuado a los requisitos inherentes de cada caso de uso o servicio que cursa en el NS (5G Americas Whitepaper, 2017).

En este contexto, se propone llevar a cabo un seguimiento exhaustivo del funcionamiento y rendimiento de los segmentos de red (NS) creados específicamente para atender los servicios y aplicaciones requeridos por los usuarios. Este monitoreo implica supervisar los parámetros de calidad de servicio (QoS, por las siglas del término en inglés, *Quality of Service*) como la latencia, *jitter*, ancho de banda y pérdida de paquetes, para garantizar el cumplimiento de los SLA correspondientes, asignar los recursos necesarios de manera adecuada y establecer una gestión automatizada que prevenga las posibles violaciones a los SLA y optimice la asignación de recursos por parte de los proveedores de servicios (Saad et al, 2021).

1.1 Planteamiento del problema

El rápido desarrollo y la creciente complejidad de las redes móviles modernas, con su flexibilidad, la heterogeneidad de los servicios, la expansión del tráfico, la aparición de más parámetros que influyen en la dinámica de red, requieren de una infraestructura de red avanzada que sostendría y mejoraría la calidad de servicio (QoS).

La integración de servicios heterogéneos y el rápido crecimiento del tráfico plantean desafíos significativos en la gestión de redes. La necesidad de adaptación y reacción inmediata se ha convertido en un requisito fundamental en este contexto. Sin embargo, las técnicas de gestión actuales, limitadas a controles y medidas en tiempo real, a menudo resultan insuficientes. Esta limitación puede conducir a un uso ineficiente de los recursos, a una disminución en la calidad de los servicios prestados y, en consecuencia, al incumplimiento de los acuerdos de nivel de servicio (SLA) para los diversos casos de uso.

Frente a estos desafíos, se hace evidente y urgente la necesidad de desarrollar nuevas estrategias que aumenten las capacidades de monitorización y gestión en las redes móviles 5G/B5G. Un problema fundamental en este contexto es la ausencia de un sistema automatizado capaz de reaccionar eficazmente ante las demandas cambiantes del entorno de red. Dicho sistema debería ser capaz de gestionar los recursos de red de manera óptima, maximizando su eficacia y optimizando la QoS de forma dinámica y adaptativa.

El núcleo del problema de investigación radica, por tanto, en la carencia de un sistema de gestión y monitorización automatizado para las redes móviles 5G/B5G. Este sistema debería abordar dos aspectos críticos: por un lado, garantizar una asignación eficiente de los recursos de red, y por otro, lograr una mejora significativa en la calidad de servicio. Dicho sistema debe de ser capaz de evitar tanto la subasignación como la sobreasignación de recursos, asegurando al mismo tiempo el cumplimiento riguroso de los SLA para cada caso de uso o servicio específico.

El desafío central consiste en desarrollar un sistema inteligente y autónomo que pueda anticiparse a las necesidades de la red, optimizar el uso de recursos en tiempo real y mantener un alto nivel de QoS en un entorno de red cada vez más complejo y dinámico. La solución a este problema no solo mejoraría la eficiencia operativa de las redes 5G/B5G, sino que también sentaría las bases para una gestión de red más robusta y adaptable en el futuro de las telecomunicaciones.

1.2 Antecedentes y justificación

A medida que las redes se vuelven más complejas y heterogéneas, la gestión efectiva de los Acuerdos de Nivel de Servicio (SLA) emerge como un desafío clave. Los estudios de Saad et al. (2021) y Habibi et al. (2018) han explorado arquitecturas de gestión automatizada y el uso del *Network Slicing* respectivamente,

destacando la necesidad de adaptar los SLA para garantizar el cumplimiento efectivo de las promesas de servicio frente a demandas dinámicas. Adicionalmente, el papel de la Inteligencia Artificial (IA) y el Aprendizaje Automático (ML) en esta área se ha vuelto indispensable, como lo indican los trabajos de (Ramos-García et al., 2021; Rekkas et al., 2021) quienes han demostrado cómo estas tecnologías pueden ser aplicadas para mejorar la clasificación y predicción de servicios en redes 5G/B5G, asegurando una asignación de recursos más eficiente y una mejor supervisión de los parámetros críticos de QoS. Este contexto tecnológico y académico basa la construcción de la investigación, subrayando la importancia de desarrollar sistemas más robustos y dinámicos para la gestión de SLA en entornos de telecomunicaciones cada vez más complejos.

Algunos trabajos han abordado aspectos relacionados, como el marco integrado de gestión de SLA en entornos 5G presentado en (Touloupou et al., 2019), donde se implementa en un ambiente 5G real y se demuestra la detección de una violación de SLA al desplegar un servicio de red en un NS. En (Ramos-García et al., 2021), los autores generaron un modelo basado en el comportamiento del SLA para la predicción del *throughput*. Además, (Kosiński et al., 2008) presenta un marco de trabajo para el monitoreo y gestión de SLA en servicios de telecomunicaciones, y propone una arquitectura de sistema que puede interoperar con sistemas de monitoreo de desempeño y herramientas de gestión. Por otro lado, en (Gonzalez-Franco et al., 2023), se presenta una propuesta para el reconocimiento de patrones en métricas de tráfico para el cumplimiento de los SLA en entornos de redes 5G/B5G.

Si bien se han realizado varios aportes al tema, no se ha implementado un mecanismo de gestión y monitorización en tiempo real que intervenga directamente en la violación de los contratos de SLA en redes de nueva generación. En este trabajo, se presenta una propuesta de un sistema de monitorización automatizado de SLA en redes 5G/B5G, vigilando la no existencia de violaciones de los SLA, con el objetivo de mejorar la eficiencia, la calidad de servicio, el rendimiento y el aprovisionamiento óptimo de los recursos de la red. Esta propuesta es significativa y pertinente, ya que aborda un desafío clave en la gestión de redes avanzadas, contribuyendo a garantizar una experiencia de usuario óptima y el cumplimiento de los acuerdos de nivel de servicio acordados.

1.3 Hipótesis

La implementación de un sistema de gestión y monitorización automatizada en las redes móviles de quinta

generación 5G/B5G permitirá una asignación más eficiente de recursos de red y una mejora significativa en la calidad de servicio (QoS) para los usuarios, al tiempo que garantizará el cumplimiento de los acuerdos de nivel de servicio (SLA) establecidos entre los proveedores de red y los usuarios.

1.4 Objetivos

En este proyecto multidisciplinario, se propone utilizar herramientas de computación, así como las distintas técnicas que ofrece la inteligencia artificial en la solución de una problemática en redes de telecomunicaciones avanzadas. Gestionar las características y parámetros dinámicos y heterogéneos (disponibilidad, confiabilidad, densidad, tráfico, latencia, pérdida de paquetes) que puntualizan su comportamiento y desempeño, que definen la asignación de recursos de red. Por lo que se propone, utilizar lo arriba indicado para diseñar un sistema de monitoreo de redes móviles avanzadas (5G/B5G), y dar seguimiento del comportamiento y desempeño de los NS creados, vigilando la no existencia de violaciones a los SLA, por lo tanto, el aprovisionamiento óptimo de recursos de red.

1.4.1 Objetivo general

Proponer un sistema de gestión de monitorización sistematizado, para vigilancia del cumplimiento de los parámetros de desempeño de los SLA inherentes al *Network Slices* en redes inalámbricas móviles 5G/B5G.

1.4.2 Objetivos específicos

1. Utilizar el reconocimiento de parámetros de servicios y aplicaciones y los *Network Slices* del servicio acorde a sus parámetros.
2. Diseñar un sistema de monitorización del comportamiento de parámetros QoS en los escenarios (casos de uso) elegidos (*testbed*, cama de pruebas).
3. Simular y monitorizar el comportamiento de los NS en dicha cama de pruebas.

1.5 Estructura de la tesis

Esta sección detalla cómo se estructura el presente trabajo de tesis. A continuación, se describen brevemente los aspectos tratados en cada capítulo.

En el Capítulo 2, se presenta el estado del arte y los fundamentos teóricos que sustentan esta investigación, se aborda en detalle la segmentación de redes 5G (*Network Slicing*), las Redes Definidas por Software (SDN), la Arquitectura basada en Servicios de 5G (SBA, por las siglas del término en inglés, *Service Based Architecture*), la gestión de Acuerdos de Nivel de Servicio (SLA) y la aplicación de técnicas de Aprendizaje de Máquina (ML) en la gestión de redes 5G. Además, se examinan trabajos relacionados y se exploran en profundidad la importancia del *Network Slicing* para satisfacer requisitos específicos de QoS, la estructura y ciclo de vida de los SLA, y los diferentes enfoques de ML como el aprendizaje supervisado, no supervisado, por refuerzo y profundo, destacando su potencial para optimizar y personalizar dinámicamente la prestación de servicios en redes 5G.

En el Capítulo 3, se presenta una propuesta integral para un sistema de gestión y monitorización de Acuerdos de Nivel de Servicio (SLA) en entornos de redes 5G/B5G. La propuesta se basa en la integración de varias tecnologías y conceptos avanzados, como las Redes Definidas por Software (SDN), la herramienta de simulación Mininet, un sistema de monitorización y una Interfaz de Programación de Aplicaciones Transferencia de Estado Representacional (API RESTful, por las siglas del término en inglés, *Application Programming Interface Representational State Transfer*) para redireccionamiento de tráfico, y técnicas de vanguardia de Aprendizaje de Máquina (ML). La arquitectura aprovecha las capacidades de programabilidad, control centralizado y abstracción de recursos de las SDN, junto con la flexibilidad de Mininet para simular entornos complejos. Además, incorpora la API RESTful para facilitar el redireccionamiento ágil de tráfico, y destaca la integración de ML para aprender patrones, anticipar violaciones de SLA y optimizar dinámicamente la asignación de recursos cumpliendo con los requisitos de calidad de servicio acordados.

En el Capítulo 4, se lleva a cabo la validación del modelo propuesto, se presentan las simulaciones realizadas y se analizan detalladamente los resultados obtenidos. Se describen las configuraciones experimentales, se incluyen los parámetros y métricas utilizadas, y se realiza una exhaustiva discusión de los hallazgos clave, se evalúan tanto el desempeño como la eficacia del enfoque propuesto en escenarios prácticos.

En el Capítulo 5, se presentan las conclusiones del trabajo de investigación, destacando las contribuciones más significativas al conocimiento en el área de gestión y monitorización de Acuerdos de Nivel de Servicio (SLA) en entornos de redes avanzadas 5G/B5G. Se resumen los hallazgos clave y se discuten las implicaciones prácticas de la investigación realizada. Además, se abordan las limitaciones encontradas durante el estudio y se plantean oportunidades y direcciones prometedoras para trabajo futuro, con el objetivo de seguir avanzando en el desarrollo de soluciones más eficientes y adaptativas que garanticen el cumplimiento de los SLA en las redes de próxima generación.

Capítulo 2. Marco teórico

Este capítulo establece las bases teóricas fundamentales para comprender la gestión de Acuerdos de Nivel de Servicio (SLA) en el contexto de las redes 5G. Se abordan conceptos clave que son esenciales para el desarrollo de esta investigación, incluyendo *Network Slicing*, Acuerdos de Nivel de Servicio, Arquitectura Basada en Servicios (SBA), Virtualización de Funciones de Red (NFV), Redes Definidas por Software (SDN), API REST y *Machine Learning*.

Las redes 5G representan un avance significativo en la arquitectura de las telecomunicaciones, impulsando la adopción de tecnologías como la Arquitectura Basada en Servicios (SBA) y el *Network Slicing*. Estas innovaciones permiten una gestión más eficiente y escalable de los recursos de red. La SBA, caracterizada por el uso de Interfaces de Programación de Aplicaciones (API) estandarizadas, facilita la interacción y automatización ágil de servicios. Por su parte, el *Network Slicing* aprovecha la orquestación dinámica de funciones de red virtualizadas (VNF) y recursos físicos, implementada a través de NFV y SDN, para crear segmentos de red personalizados que se adaptan a diversos servicios y casos de uso (Jalalian et al., 2023).

La integración de estas tecnologías en 5G no solo mejora la adaptabilidad a los requerimientos tecnológicos emergentes, sino que también optimiza el rendimiento y la eficiencia de la infraestructura de comunicaciones. Este marco teórico proporciona el fundamento necesario para explorar cómo estas tecnologías confluyen en la gestión efectiva de los SLA en las redes 5G, tema central de los capítulos subsiguientes.

2.1 Segmentación de la red

El Segmento de Red (NS) constituye una de las novedades más importantes que introduce la tecnología 5G, tiene por objetivo cubrir uno de los principales requisitos que los organismos de estandarización imponen a los Sistemas de Quinta Generación: poder proporcionar los niveles de calidad requeridos por una gran variedad de servicios de naturaleza muy distinta, de la forma más eficiente y escalable posible (Ordonez-Lucena et al., 2017). La segmentación de red 5G es una arquitectura de red que proporciona una forma de dividir una red para proporcionar redes lógicas independientes punto a punto sobre recursos y funcionalidades de red física (Zhang, 2019). Esto puede ayudar a los operadores a proporcionar servicios diferenciados y desplegar más rápidamente casos nuevos. Un operador puede utilizar la fragmentación de

red para asignar lógicamente recursos físicos a través de una o más segmentos, donde cada uno de ellos puede tener una calidad de servicio (QoS) diferente y otras características de rendimiento, así como configuraciones y políticas, para cumplir con una variedad de casos de uso y posibles acuerdos de nivel de servicio (SLA). Por ejemplo, como se ilustra en la Figura 2, un segmento que soporta usuarios de banda ancha móvil requiere altas tasas de datos y volúmenes de tráfico, otra que soporta dispositivos de Internet de las cosas puede optimizar dispositivos de alta densidad y consumo de energía, y finalmente la que soporta conducción autónoma puede proporcionar comunicaciones de alta confiabilidad y baja latencia (Jalalian et al., 2023). La segmentación mejora el rendimiento de la red, no proporciona recursos innecesarios y permite la personalización de recursos, la priorización de tráfico y la adaptación a las necesidades de servicios específicos. Lo cual conduce a una red más eficiente, escalable y capaz de proporcionar un rendimiento óptimo para una amplia variedad de servicios y aplicaciones en entornos de redes avanzados como 5G/B5G (Zhang, 2019).

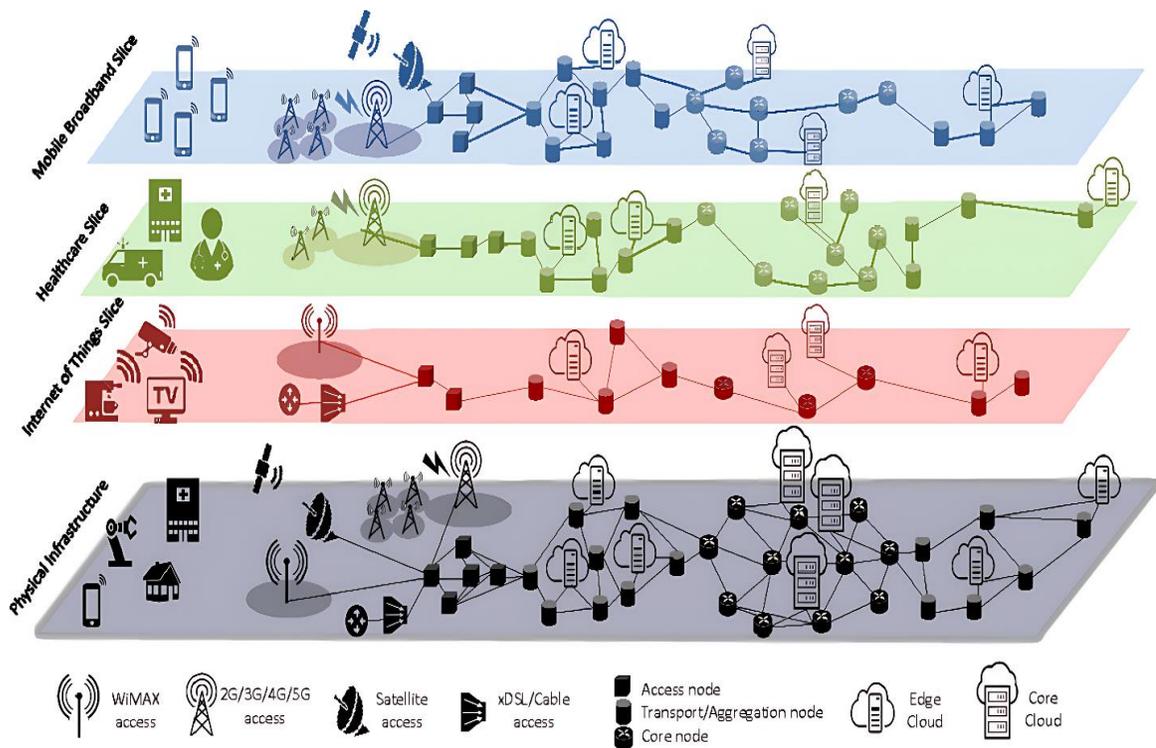


Figura 2. Segmentos de red. Tomada de (Jalalian et al., 2023).

2.2 Acuerdos de Nivel de Servicio (SLA)

De acuerdo con (Kosiński et al., 2008) un SLA, Acuerdo de Nivel de Servicio, es un contrato formal o acuerdo

entre un proveedor de servicios y un cliente que establece los niveles de calidad y rendimiento esperados para un servicio específico. Los principales componentes de un SLA suelen incluir:

- ✓ **Objetivos de rendimiento:** disponibilidad del servicio, velocidad de respuesta, velocidad de transferencia de datos, la latencia y otros parámetros relevantes para el servicio.
- ✓ **Responsabilidades:** tanto del proveedor de servicios como del cliente en términos de mantener y cumplir los niveles de rendimiento acordados.
- ✓ **Métricas de medición:** define cómo se medirán los niveles de rendimiento, procedimientos de seguimiento, indica cómo se llevará a cabo el seguimiento de los niveles de servicio y cómo se reportarán los resultados.
- ✓ **Y reportes y consecuencias por incumplimiento:** especifica las consecuencias o penalizaciones que pueden aplicarse si el proveedor no cumple con los niveles acordados (Kosiński et al., 2008).

2.2.1 Estructura de un SLA

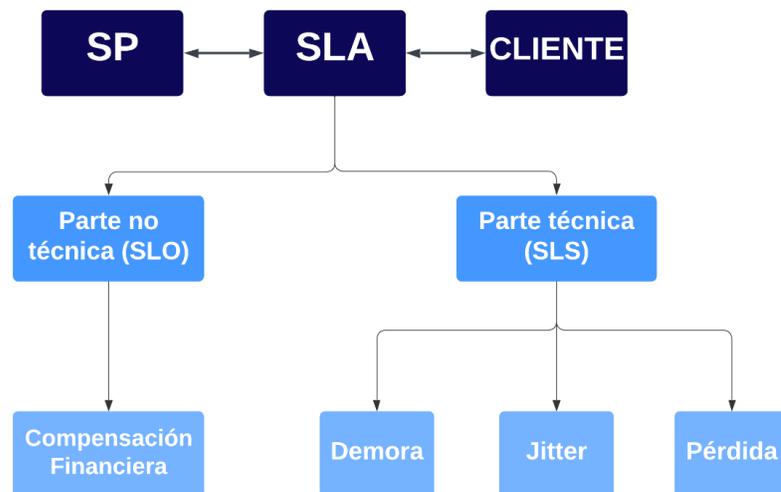


Figura 3. Estructura de un SLA. Adaptada de (Ramos-García et al., 2021).

Un SLA típicamente consta de dos partes: una técnica y otra no técnica, como se ilustra en la Figura. 3, la parte no técnica está conformada por los Objetivos de Nivel de Servicio (SLO, por las siglas del término en

inglés, *Service Level Objectives*), que desglosan el SLA en metas individuales. Estos definen métricas para hacer cumplir o supervisar el SLA, y evaluar si se alcanzan los objetivos. Algunos ejemplos incluyen el tiempo de actividad o el tiempo medio entre fallas (MTBF, por las siglas del término en inglés, *Mean Time Between Failures*), el tiempo de respuesta y el tiempo medio de reparación (MTTR, por las siglas del término en inglés, *Mean Time To Repair*). También se incluyen en los parámetros no técnicos los procedimientos de violación, que se activan en caso de incumplimiento de las Especificaciones del Nivel de Servicio (SLS, por las siglas del término en inglés, *Service Level Specifications*)(Ramos-García et al., 2021). Por otro lado, las Especificaciones del Nivel de Servicio (SLS) constituyen la parte técnica de un SLA e incorporan métricas de calidad de servicio (QoS), como latencia, rendimiento, variabilidad del retardo (*jitter*), pérdida de paquetes y disponibilidad, entre otras. Las SLS contribuyen al control integral de la QoS de extremo a extremo, utilizando métricas cuantitativas.

2.2.2 Ciclo de vida de un SLA



Figura 4. Ciclo de vida de un SLA. Adaptada de (Ramos-García et al., 2021).

La gestión del SLA en el contexto del dominio 5G es una faceta integral de la prestación de servicios, dirigida por las plataformas de servicios 5G. Este proceso dinámico se despliega a través de cuatro fases fundamentales: arquitectura, compromiso, operación y terminación, como ilustra la Figura 4. El ciclo de vida del SLA se alinea con los principios centrales de 5G y opera junto al del servicio de red (Touloupou et al., 2019).

Como se ilustra en la Figura 4 la primera etapa inicia con la elección de un servicio de red y la definición de requisitos por parte del desarrollador. El operador, generalmente, revisa y considera estos requisitos comerciales fundamentales, implementando plantillas de SLA como una oferta inicial para los clientes del servicio. En la segunda fase, la selección de servicios de red se basa en consideraciones comerciales, con restricciones de Calidad de Servicio (QoS) que también pueden ser definidas como requisitos del acuerdo. La preferencia de un operador o proveedor de servicios de red depende del servicio de red deseado, sus características y restricciones presupuestarias. Las expectativas de QoS llevan a los usuarios a negociar con sus operadores o proveedores de servicios los niveles acordados. Tras una negociación exitosa, se genera un SLA para describir los parámetros de QoS acordados (Kosiński et al., 2008).

Posteriormente, la fase de operación implica el despliegue real del servicio, la población de servicios con datos en ejecución, el establecimiento de canales de comunicación y otras actividades operativas. Esta fase monitorea datos en tiempo real para evitar o gestionar violaciones inesperadas de los SLA. En la fase de terminación, se maneja el fin de la relación entre el operador o proveedor de servicios y el cliente, incluyendo la evaluación de alternativas, compromisos de liquidación y terminación, exportación de datos, atención al cliente y diligencia, así como la supresión de datos. Todos estos aspectos deben considerarse si el servicio de red se rescinde o se viola el SLA (Kapassa et al., 2019).

2.3 Arquitectura Basada en Servicios (SBA)

Las redes 5G aportan cambios transformadores al núcleo de la red (CN, por las siglas del término en inglés, *Core Network*) con un enfoque modular y nativo de la nube. Un avance clave es que actualiza la arquitectura tradicional de telecomunicaciones a la arquitectura basada en servicios (SBA) definida por (3rd Generation Partnership Project, 2018), lo que permite un desarrollo de servicios más flexible. Introducido para mejorar el modularidad del sistema de red. Como se ilustra en la Figura 5, la SBA permite que los elementos de red o las funciones de red (NF, por las siglas del término en inglés, *Network Function*) en 5G se comuniquen entre sí a través de una interfaz basada en servicios SBI. Permite el desacoplamiento de NF con funcionalidades más precisas. Cada NF está en contenedores y funciona de forma independiente, y expone su funcionalidad a otros NF a través de un SBI (3rd Generation Partnership Project, 2018).

La SBA ofrece una serie de beneficios como el despliegue de contenedores orquestados por *Kubernetes* (un sistema de código abierto para automatizar la implementación, el escalado y la gestión de aplicaciones en

contenedores), lo que permite que el núcleo se ejecute en una infraestructura no propietaria, permite a los nuevos proveedores de software conectar y usar sus NF para obtener el mejor enfoque de su clase, además permite la segmentación de la red, con una utilización dinámica y eficiente de los recursos, simplifica las operaciones mediante la interfaz de programación de aplicaciones (API), así como, aprovecha el uso de protocolos armonizados como HTTP/2 y sus mecanismos de seguridad bien desarrollados por lo que facilita la integración perfecta de aplicaciones de terceros con la red central (3GPP Release, 2018).

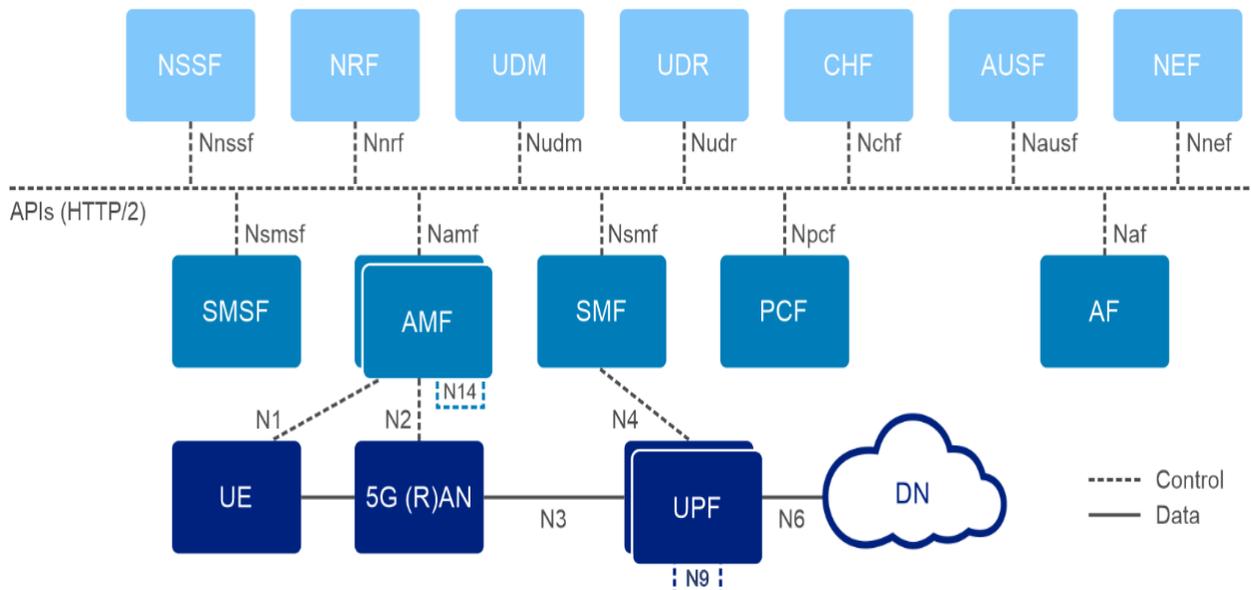


Figura 5. Arquitectura SBA en 5G. Tomada de (3GPP Release, 2018).

Funcionalidad principal de las Funciones de red (3rd Generation Partnership Project, 2018, 2019a):

- ✓ Función de Gestión de Acceso y Movilidad (AMF, por las siglas del término en inglés, *Access and Mobility Management Function*): gestiona el registro y la movilidad de UE. Por ejemplo, rastrea la ubicación del UE y proporciona eventos de movilidad a otras NF.
- ✓ Función de Servidor de Autenticación (AUSF, por las siglas del término en inglés, *Authentication Server Function*): servidor que atiende las funciones de Autenticación.
- ✓ Función de Aplicación (AF, por las siglas del término en inglés, *Application Function*): funciones generales o específicas para las aplicaciones.

- ✓ Función de Control de Políticas (PCF, por las siglas del término en inglés, *Policy Control Function*): gestión de políticas para suscriptores y para segmentos de red. Esto incluye la selección de rutas de tráfico, reenvío de tráfico, políticas de itinerancia, tarifas y políticas de QoS.
- ✓ Función de Exposición de Red (NEF, por las siglas del término en inglés, *Network Exposure Function*): proporciona un puente hacia las verticales (sectores específicos de la industria, como la salud, la automoción, la manufactura, etc.) para acceso a los datos y funcionalidades del núcleo 5G. Expone datos a terceros, asigna identificadores y convierte protocolos.
- ✓ Función de Repositorio de Red (NRF, por las siglas del término en inglés, *Network Repository Function*): admite el descubrimiento de servicios, la autorización del consumidor de NF y mantiene los perfiles de NF, los servicios y la lista de instancias de NF. Actúa como el servidor de autorización.
- ✓ Función de Selección de Segmento de Red (NSSF, por las siglas del término en inglés, *Network Slice Selection Function*): ayuda en la selección de instancias de segmento de red adecuadas para los usuarios y en la asignación de las funciones de gestión de acceso (AMF) necesarias.
- ✓ Función de Gestión de Sesiones (SMF, por las siglas del término en inglés, *Session Management Function*): realiza el establecimiento, modificación y liberación de sesiones. La asignación de la dirección IP y la gestión de la UE hace que la recopilación de datos de carga, la dirección del tráfico y el enrutamiento en la UPF.
- ✓ Gestión Unificada de Datos (UDM, por las siglas del término en inglés, *Unified Data Management*): admite el acceso al almacenamiento de datos. Por ejemplo, para la gestión de suscripción, autorización de acceso y servicio, almacenamiento y gestión de identificación de usuario, autenticación de usuario.
- ✓ Función del Plano de Usuario (UPF, por las siglas del término en inglés, *User Plane Function*): proporciona el punto de sesión de la PDU externa de interconexión a las redes de datos. Realiza inspección de paquetes y aplicación de políticas y reglas de QoS para el plano del usuario.

La SBA no solo optimiza la gestión técnica y operativa de las redes 5G, sino que también impulsa un ecosistema de red más inclusivo y adaptable. Estas características aseguran que la red 5G pueda sostener no sólo las demandas actuales sino también las futuras innovaciones en servicios y aplicaciones, marcando

un hito en la evolución de las telecomunicaciones hacia sistemas más abiertos y flexibles. Esta arquitectura facilita la interoperabilidad entre diferentes proveedores y plataformas, fomentando la colaboración en un entorno multioperador que promueve la innovación a través del uso eficiente de recursos y servicios. (3GPP Release, 2018, 2019).

2.4 Virtualización de Funciones de Red (NFV)

La Virtualización de Funciones de Red (NFV) representa un cambio de paradigma en la forma en que se implementan y gestionan las funciones de red tradicionalmente. NFV propone virtualizar diversas funciones de red como firewalls, Traducción de Direcciones de Red (NAT, por las siglas del término en inglés, *Network Address Translation*), Red Privada Virtual (VPN, por las siglas del término en inglés, *Virtual Private Network*), optimizadores TCP, Inspección Profunda de Paquetes (DPI, por las siglas del término en inglés, *Deep Packet Inspection*), etc. sobre hardware básico y genérico en lugar de utilizar dispositivos dedicados y propietarios (Yousaf et al., 2017). La principal premisa de NFV es separar las funciones de red del hardware subyacente en el que se ejecutan. De esta manera, las funciones de red pueden implementarse como software ejecutándose sobre infraestructura virtualizada estándar, desacoplando así el plano de control del plano de datos. Este desacoplamiento trae varias ventajas clave:

- ✓ Permite evolucionar el software y hardware por separado, sin la dependencia mutua existente en los enfoques tradicionales.
- ✓ Brinda mayor flexibilidad, ya que el mismo hardware puede ejecutar diferentes funciones de red virtualizadas (VNF) según las necesidades cambiantes.
- ✓ Posibilita la operación dinámica de la red y el aprovisionamiento ágil de servicios mediante el escalado elástico de las VNF según la demanda. (Yousaf et al., 2017)

2.4.1 Arquitectura NFV

El marco de referencia del Instituto Europeo de Normas de Telecomunicaciones (ETSI, por las siglas del término en inglés, *European Telecommunications Standards Institute*) para NFV define una arquitectura

modular como se ilustra en la Figura 6 con tres componentes principales: la Infraestructura NFV (NFVI), las Funciones de Red Virtualizadas (VNF) y los sistemas de Gestión y Orquestación (MANO, por las siglas del término en inglés, *Management and Orchestration*) (Juniper Networks, 2023).

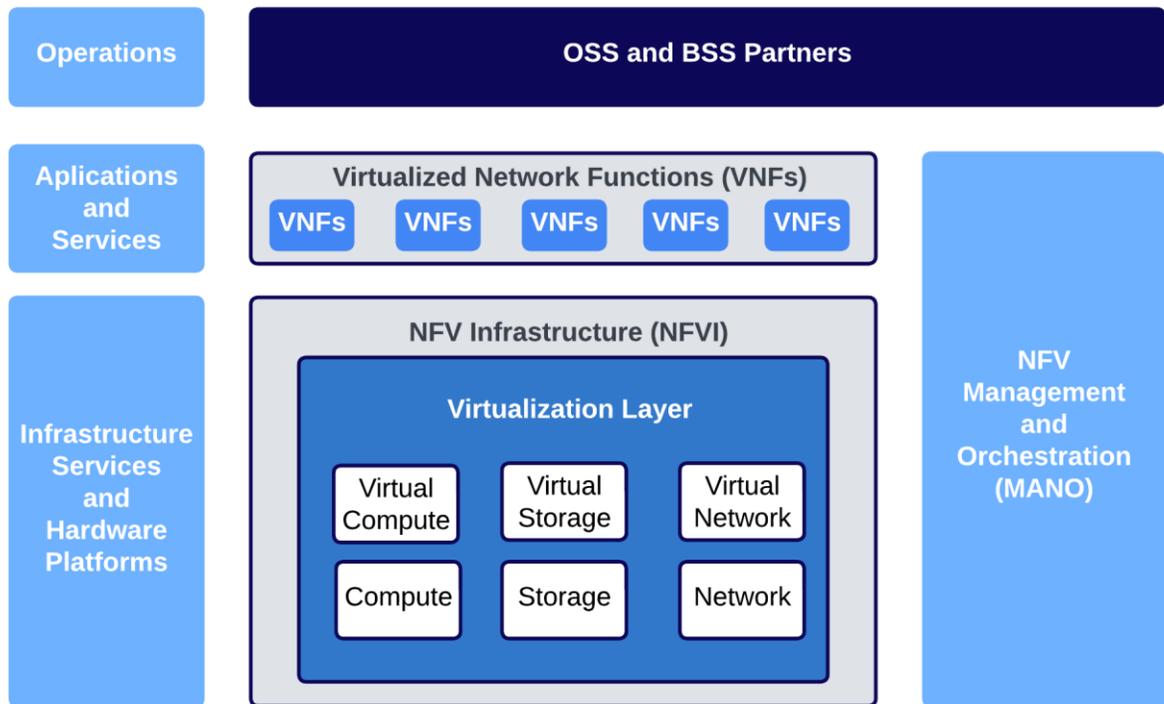


Figura 6. Arquitectura NFV. Adaptada de (Juniper Networks, 2023).

- ✓ La infraestructura de virtualización de funciones de red (NFVI) es la base de la arquitectura NFV. Esta capa proporciona los recursos físicos de cómputo, almacenamiento y redes, así como el entorno virtualizado (hipervisores o gestores de contenedores como *Kubernetes*) donde se implementan las funciones de red virtualizadas (VNF). El gestor de infraestructura se encarga de administrar y asignar estos recursos a las VNF (Barakabitze et al., 2020).
- ✓ Las VNF son aplicaciones basadas en software que ofrecen servicios de red específicos. Estas funciones utilizan la infraestructura virtualizada proporcionada por la NFVI para conectarse a la red y brindar servicios escalables y programables. Los gestores de VNF gestionan el ciclo de vida y el software de las instancias VNF (Barakabitze et al., 2020).
- ✓ La gestión y orquestación (MANO) coordina y automatiza el aprovisionamiento y encadenamiento de las VNF para crear servicios de red completos. MANO interactúa con los gestores de VNF y

orquestra los flujos de trabajo necesarios para instanciar y ensamblar las distintas funciones virtualizadas. Además, MANO conecta la arquitectura NFV con los sistemas OSS/BSS existentes (Juniper Networks, 2023).

- ✓ Los Sistemas de Soporte de Operaciones (OSS, por las siglas del término en inglés, *Operations Support Systems*) se utilizan para monitorear, administrar y optimizar el rendimiento de la red de telecomunicaciones, así como para brindar soporte a los servicios ofrecidos a los clientes. Incluyen sistemas de gestión de red, inventario, planificación y diseño, y gestión de servicios (Juniper Networks, 2023).
- ✓ Los sistemas Sistemas de Soporte Empresarial (BSS, por las siglas del término en inglés, *Business Support Systems*) gestionan los procesos comerciales de los proveedores de servicios, como la facturación, pedidos, tarifas, campañas y relaciones con clientes. Comprenden sistemas de facturación, gestión de pedidos, tarifas y Gestión de Relaciones con los Clientes (CRM por las siglas del término en inglés, *Customer Relationship Management*) (Bouras et al., 2017).

2.5 Redes Definidas por Software (SDN)

Las redes definidas por software (SDN) introducen un nuevo paradigma en la gestión y el control de redes, con cinco características principales (Lozano-Rizk et al., 2022):

- ✓ Separación del plano de control y el plano de datos. SDN separa las decisiones de control y enrutamiento (plano de control) de la transferencia real de datos (plano de datos). Esto posibilita la gestión centralizada y el control programable de la red.
- ✓ Programabilidad y automatización: permite programar el comportamiento de red a través de API (Interfaz de Programación de Aplicaciones), facilitando la automatización de tareas y la creación de políticas.
- ✓ Abstracción de recursos: las aplicaciones interactúan con interfaces abstractas sin necesidad de entender la complejidad de hardware.
- ✓ Gestión centralizada: el controlador SDN centralizado recopila información de red y toma

decisiones de enrutamiento y configuración. Esto habilita visibilidad completa y control efectivo desde un único punto.

- ✓ Flexibilidad y adaptabilidad: la programabilidad de SDN permite responder ágilmente a nuevas necesidades y aplicaciones sin realizar cambios físicos en la infraestructura. SDN es altamente adaptable (Ordóñez-Lucena et al., 2017).

2.5.1 Arquitectura SDN

Los principales componentes de una Red Definida por Software se ilustran en la Figura 7, y se describen líneas abajo:

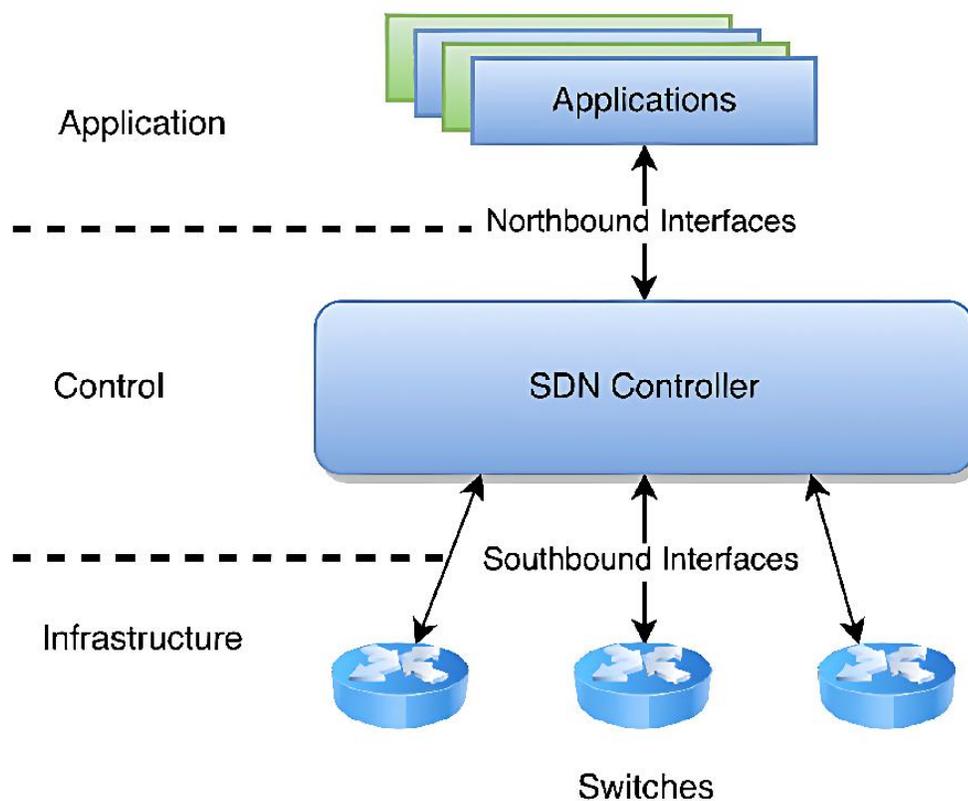


Figura 7. Arquitectura SDN. Tomada de (L. Wang & Yang, 2018).

El Plano de Datos o Infraestructura se compone de los dispositivos de red físicos que conforman el plano de datos de la arquitectura SDN. Estos dispositivos incluyen conmutadores y enrutadores ubicados

típicamente en los centros de datos. Su función principal es el reenvío de paquetes de datos y el procesamiento de tráfico de red (Ordóñez-Lucena et al., 2017) Además, estos dispositivos recopilan información crítica sobre el uso, topología y estado de la red, que es remitida a la capa de control. Los dispositivos de la capa de infraestructura pueden ser puramente dispositivos SDN, donde toda la inteligencia de control proviene del controlador centralizado, o dispositivos híbridos que combinan capacidades SDN con protocolos de redes tradicionales. A pesar de ser el plano de datos, estos dispositivos siguen siendo programables gracias a la interfaz *southbound* abierta como *OpenFlow*, a través de la cual el controlador SDN puede configurar dinámicamente su comportamiento de reenvío según las necesidades de la red y aplicaciones (Distrital Francisco Jose de Caldas et al., 2021).

El Plano de Control está representado por el controlador SDN, que es una plataforma de software centralizada que actúa como cerebro de la red. Mantiene una vista holística de toda la infraestructura y es responsable de tomar decisiones inteligentes de enrutamiento y control de tráfico, gestionar dinámicamente los recursos de red, comunicarse con los dispositivos de la capa de infraestructura a través de API *southbound* como *OpenFlow* para programar su comportamiento de reenvío, e interactuar con las aplicaciones y servicios de la capa superior mediante API *northbound* para facilitar la entrega de servicios de red personalizados de manera programable y automatizada. Existen diferentes tipos de controladores como son OpenDaylight (ODL), ONOS, RYU, POX, entre otros (Yousaf et al., 2017).

El Plano de Aplicación constituye la capa superior donde residen las aplicaciones y servicios de red que consumen y solicitan recursos y capacidades específicas de la red subyacente de manera programática. A través de interfaces *northbound* abiertas, típicamente APIs RESTful, estas aplicaciones se comunican con el controlador SDN para expresar sus requerimientos, los cuales se traducen por el controlador en instrucciones y reglas que se aplican dinámicamente en los dispositivos de la infraestructura. Esta capa representa a los consumidores finales de los servicios de red programables y automatizados habilitados por la separación de los planos de control y datos en SDN, permitiendo un aprovisionamiento ágil, flexible y eficiente de recursos de red personalizados según las necesidades cambiantes de cada aplicación (Distrital Francisco Jose de Caldas et al., 2021).

El Protocolo *OpenFlow* es el principal protocolo estándar que habilita la comunicación entre el controlador SDN y los dispositivos de red de la capa de infraestructura en la arquitectura SDN. Propuesto originalmente en 2008 y estandarizado por la Fundación de Redes Abiertas (ONF, por las siglas del término en inglés, *Open Networking Foundation*), *OpenFlow* permite que el controlador SDN programe de manera remota las tablas

de flujo en los dispositivos *OpenFlow* compatibles, como conmutadores y enrutadores. De esta forma, el controlador puede definir reglas de reenvío de paquetes personalizadas en la red física subyacente de acuerdo con los requerimientos de las aplicaciones y servicios en el plano de aplicación. *OpenFlow* establece una interfaz abierta entre los planos de control y datos separados en SDN, habilita la programabilidad, automatización y control centralizado de la red (Lozano-Rizk et al., 2022).

2.6 API REST

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas y definiciones que permite a diferentes aplicaciones o sistemas comunicarse entre sí, facilitando el intercambio de datos y la ejecución de funciones específicas.

Una API REST, o RESTful (por las siglas del término en inglés, *Representational State Transfer Application Programming Interface*) se distingue por varias características esenciales. En primer lugar, adopta una arquitectura basada en estándares Web, utiliza protocolos HTTP para la transferencia de datos. Además, sigue el principio de representación de recursos, lo que significa que los datos están organizados y accesibles de manera jerárquica, lo cual permite una fácil navegación. La comunicación se realiza mediante operaciones HTTP estándar, a saber, GET para obtener información, POST para crear nuevos recursos, PUT para actualizarlos y DELETE para eliminarlos. La API REST es *stateless*, lo que implica que cada solicitud del cliente al servidor contiene toda la información necesaria para comprender y procesar la solicitud, sin necesidad de mantener el estado de la sesión. Esto resulta en una arquitectura simple, escalable y fácil de mantener. Además, la API REST utiliza formatos de representación comunes como JSON o XML, lo que facilita la interoperabilidad y la integración con diversas aplicaciones y sistemas (Ferdian Thung, 2016).

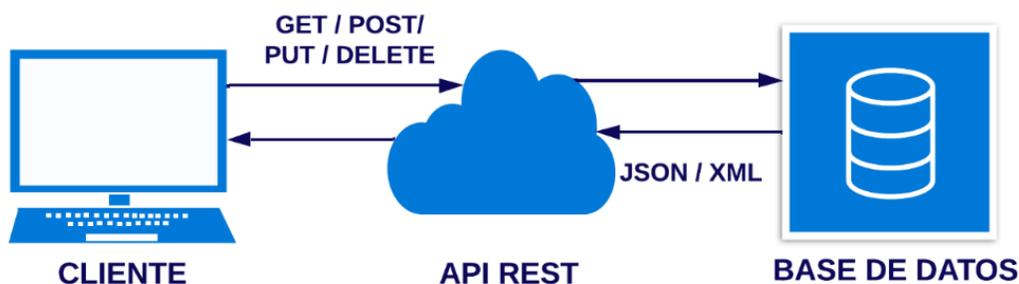


Figura 8. Funcionamiento de una API REST. Adaptada de (Gonzalez-Franco et al., 2023).

Las API RESTful incluyen los siguientes beneficios:

- ✓ Escalabilidad: los sistemas que implementan API REST pueden escalar de forma eficiente porque REST optimiza las interacciones entre el cliente y el servidor.
- ✓ Flexibilidad: los servicios web RESTful admiten una separación total entre el cliente y el servidor. Simplifican y desacoplan varios componentes del servidor, de manera que cada parte pueda evolucionar de manera independiente. Los cambios de la plataforma o la tecnología en la aplicación del servidor no afectan la aplicación del cliente. La capacidad de ordenar en capas las funciones de la aplicación aumenta la flexibilidad aún más. Por ejemplo, los desarrolladores pueden efectuar cambios en la capa de la base de datos sin tener que volver a escribir la lógica de la aplicación (Ferdian Thung, 2016).
- ✓ Independencia: las API REST son independientes de la tecnología que se utiliza. Puede escribir aplicaciones del lado del cliente y del servidor en diversos lenguajes de programación, sin afectar el diseño de la API. También puede cambiar la tecnología subyacente en cualquiera de los lados sin que se vea afectada la comunicación.

Las API RESTful facilitan el desarrollo ágil y escalable de aplicaciones web y servicios distribuidos al promover el desacoplamiento, modularidad, portabilidad y optimización del rendimiento en los flujos de comunicación cliente-servidor (Lozano-Rizk et al., 2022).

2.7 Aprendizaje de máquina o *Machine Learning*

El Aprendizaje de Máquina o *Machine Learning*, se enmarca en la inteligencia artificial, un campo de estudio que busca dotar a las máquinas de habilidades similares a las de los seres humanos. Desarrolla métodos que permiten a las máquinas aprender a partir de muchos ejemplos. En términos computacionales, se puede entender que un programa aprende a medida que realiza una tarea específica y mejora su desempeño con la experiencia adquirida (Khan y Adholiya, 2023).

2.7.1 Clasificación tipos de *Machine Learning*

El campo del *Machine Learning* abarca diversas metodologías que se emplean para entrenar a los modelos a realizar predicciones y tomar decisiones basadas en datos. Entre los tipos más comunes se encuentran el aprendizaje supervisado y no supervisado, pero también existen otros enfoques importantes como el aprendizaje por refuerzo, el aprendizaje semi supervisado y el aprendizaje profundo (*Deep Learning*).

El aprendizaje automático se divide principalmente en dos enfoques: supervisado y no supervisado. El aprendizaje supervisado utiliza datos etiquetados para entrenar modelos que puedan hacer predicciones o clasificaciones precisas sobre nuevos datos. Por otro lado, el aprendizaje no supervisado trabaja con datos no etiquetados para descubrir patrones, agrupar información y extraer conocimientos útiles sin tener acceso a respuestas predefinidas (López Telenchana et al., 2024) .

Además de estos dos enfoques principales, existen otras clasificaciones importantes en el aprendizaje automático. El aprendizaje por refuerzo combina elementos supervisados y no supervisados para que los modelos aprendan a tomar decisiones que maximicen las recompensas en un entorno dado. Por último, el aprendizaje semi supervisado utiliza una combinación de datos etiquetados y no etiquetados, aprovechando la información disponible en una cantidad limitada de datos etiquetados para mejorar el rendimiento del modelo (Ramos-García et al., 2021).

2.7.1.1 Aprendizaje Profundo o *Deep Learning*

El aprendizaje profundo, por las siglas del término en inglés, *Deep Learning*, es un enfoque del aprendizaje de máquina que se centra en el uso de algoritmos para abordar tareas altamente complejas mediante el análisis de grandes volúmenes de datos. Este enfoque aprovecha redes neuronales artificiales jerarquizadas que se organizan en múltiples capas o niveles. Cada capa tiene una función específica en el procesamiento de datos y, a medida que avanzamos en la jerarquía, la complejidad de las representaciones aumenta (Ramos-García et al., 2021).

Un aspecto crucial en el aprendizaje profundo es el proceso de entrenamiento del modelo para minimizar la función de pérdida. Este proceso implica ajustar los pesos de las conexiones neuronales a través de iteraciones, de manera que el modelo pueda hacer predicciones cada vez más precisas en nuevos datos

una vez que ha completado su entrenamiento. El aprendizaje profundo ha demostrado ser efectivo en aplicaciones, desde el procesamiento de lenguaje natural hasta la visión por computadora y la toma de decisiones en tiempo real.

Deep Learning en comparación con los métodos tradicionales de *Machine Learning*, como se ilustra en la Figura 9, tiene la ventaja de realizar la extracción de características y la clasificación o regresión en un único modelo, lo que simplifica considerablemente el diseño y desarrollo de sistemas. Esto significa que podemos abordar tareas complejas de manera más eficiente, lo que ha impulsado el uso generalizado del *Deep Learning* en una amplia variedad de aplicaciones (Rafael Zambrano, 2019).

La elección entre *Machine Learning* y *Deep Learning* depende de la naturaleza del problema, la disponibilidad de datos y recursos informáticos, y los objetivos específicos de la aplicación. Ambos enfoques tienen su lugar y aplicaciones adecuadas en el campo de la inteligencia artificial (Rafael Zambrano, 2019).

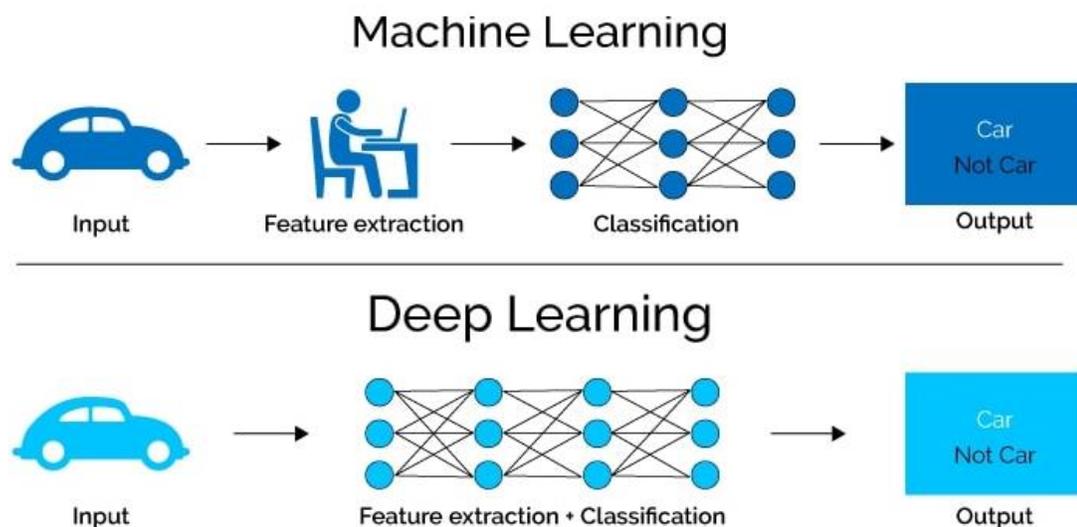


Figura 9. Diferencias entre *Machine Learning* y *Deep Learning*. Tomada de (Rafael Zambrano, 2019).

2.7.2 Importancia del uso de *Machine Learning* en la predicción de SLA

La utilización de técnicas de *Machine Learning* para predecir el cumplimiento de los SLA en redes 5G es de gran importancia por varias razones (Khan y Adholiya, 2023):

- ✓ Complejidad de las redes 5G: las redes 5G son significativamente más complejas que las generaciones anteriores debido a características como *network slicing*, virtualización de funciones de red (NFV), y la diversidad de servicios que soportan (eMBB, mMTC, URLLC). Esta complejidad hace que sea muy difícil predecir el rendimiento y cumplimiento de SLA utilizando métodos tradicionales.
- ✓ Naturaleza dinámica: estas redes son altamente dinámicas, con recursos que se asignan y reasignan constantemente según la demanda. El *machine learning* puede adaptarse mejor a estos cambios rápidos que los métodos estáticos (Rekkas et al., 2021; Khan y Adholiya, 2023).
- ✓ Gran volumen de datos: generan enormes cantidades de datos operativos. Las técnicas de ML son ideales para procesar y extraer información útil de estos grandes volúmenes de datos.
- ✓ Requisitos estrictos de SLA: muchos servicios exigen condiciones de SLA extremadamente rigurosas, especialmente en casos de uso críticos. Predecir con precisión el cumplimiento de estos SLA es crucial para garantizar la calidad del servicio.
- ✓ Prevención proactiva: el uso de ML permite una gestión proactiva de la red, prediciendo potenciales violaciones de SLA antes de que ocurran, lo que permite tomar medidas preventivas.
- ✓ Optimización de recursos: al predecir el cumplimiento de SLA, se pueden asignar recursos de manera más eficiente, evitando tanto el sobredimensionamiento como la escasez de recursos (Pavani y Kathirvel, 2023) .
- ✓ Personalización de servicios: puede ayudar a personalizar los SLA para diferentes tipos de servicios y clientes basándose en patrones de uso y requisitos específicos.
- ✓ Automatización: la predicción automatizada del cumplimiento de SLA mediante ML es clave para redes en 5G.
- ✓ Manejo de múltiples variables: el ML puede manejar eficazmente múltiples variables que afectan el cumplimiento de SLA y sus complejas interrelaciones.

- ✓ Adaptabilidad: los modelos pueden adaptarse y mejorar continuamente a medida que obtienen más datos, lo que los hace ideales para el entorno en constante evolución de las redes 5G (Pavani y Kathirvel, 2023).

El uso de técnicas de *machine learning* para predecir el cumplimiento de SLA en redes 5G es crucial para garantizar la calidad del servicio, optimizar el uso de recursos y gestionar eficazmente la complejidad y dinamismo inherentes a estas redes avanzadas (Rekkas et al., 2021).

2.8 Conclusiones parciales

En este capítulo, se han explorado los conceptos clave que sustentan la gestión de Acuerdos de Nivel de Servicio (SLA) en las redes 5G/B5G, destacando la importancia de la arquitectura basada en servicios, el *network slicing*, las tecnologías habilitadoras como NFV y SDN, y el papel de las API y el aprendizaje de máquina en la provisión eficiente y personalizada de servicios de red. Se puede concluir lo siguiente:

- ✓ La arquitectura basada en servicios (SBA) y las redes de servicios desempeñan un papel crucial en la gestión eficiente y escalable de los recursos de red en 5G, permitiendo el cumplimiento de los Acuerdos de Nivel de Servicio (SLA) mediante la interacción y automatización ágil de servicios a través de Interfaces de Programación de Aplicaciones (API) estandarizadas.
- ✓ El *Network Slicing* (NS) es una característica clave de 5G que permite dividir la red física en redes lógicas independientes, cada una optimizada para satisfacer los requisitos específicos de calidad de servicio (QoS) y rendimiento de diferentes casos de uso y aplicaciones.
- ✓ Los SLA son contratos formales que establecen los niveles de calidad y rendimiento esperados para un servicio específico, y su gestión en 5G sigue un ciclo de vida que incluye fases de arquitectura, compromiso, operación y terminación.
- ✓ La virtualización de funciones de red (NFV) y las redes definidas por software (SDN) son tecnologías clave que brindan flexibilidad, escalabilidad y eficiencia en la asignación de recursos de red en 5G, al desacoplar las funciones de red del hardware subyacente y permitir el control programable y centralizado de la red.

Las API RESTful y las técnicas de *Machine Learning*, como el aprendizaje supervisado, no supervisado, por refuerzo y profundo, desempeñan un papel importante en la automatización, personalización y optimización de los servicios de red en 5G, aprovechando la modularidad y programabilidad de la arquitectura SBA.

Capítulo 3. Propuesta del sistema de gestión y monitorización de SLA

En los sistemas de comunicaciones modernos, garantizar el cumplimiento de los Acuerdos de Nivel de Servicio (SLA) establecidos con los usuarios es un desafío crítico. A medida que se avanza hacia entornos de redes más avanzados como 5G/B5G, la complejidad y la naturaleza dinámica de las aplicaciones emergentes exigen soluciones innovadoras y adaptativas para la gestión y monitorización de SLA de manera eficiente y escalable.

En este capítulo, se presenta una propuesta integral para un sistema de gestión y monitorización de SLA en entornos de redes 5G/B5G. La propuesta se basa en la integración sinérgica de varias tecnologías y conceptos avanzados, incluyendo las Redes Definidas por Software (SDN), la herramienta de simulación Mininet, un sistema de monitorización y una API de redireccionamiento, así como técnicas de vanguardia de Aprendizaje de Máquina (ML).

La arquitectura propuesta aprovecha las capacidades de programabilidad, control centralizado y abstracción de recursos que ofrecen las SDN, junto con la flexibilidad y escalabilidad de Mininet para simular entornos de red complejos. Además, incorpora un sistema de monitorización para verificar el cumplimiento de los SLA y una API RESTful para facilitar el redireccionamiento de tráfico de manera ágil y estandarizada. Pero lo más destacado es la integración de técnicas de ML, que permiten al sistema aprender de los patrones de tráfico y comportamiento de la red, anticipar posibles violaciones de SLA y optimizar dinámicamente la asignación de recursos para cumplir con los requisitos de calidad de servicio (QoS) acordados.

3.1 Esquema general de la propuesta

El esquema que se describe en la Figura 10 propone un enfoque integrado que combina varios conceptos avanzados para lograr un sistema de gestión y monitorización de SLA (Acuerdo de Nivel de Servicio) adaptativo y eficiente.

Las Redes Definidas por Software (SDN) permiten la programación y control centralizado de la red al separar el plano de control del plano de datos, lo que facilita la implementación de políticas de gestión más flexibles

y adaptables (Bouras et al., 2017). Mininet, es una herramienta de emulación de redes que permite crear redes virtuales realistas en un solo sistema y se utiliza como entorno de prueba para evaluar el sistema de gestión y monitorización de SLA propuesto. Se propone un sistema de monitorización en conjunto con una API de enrutamiento o redireccionamiento del tráfico de red, que recopila datos y estadísticas, optimiza el rendimiento y asegura el cumplimiento de los SLA. Además, se incorporan técnicas de *Machine Learning* para analizar los datos de la bitácora de la red, anticipar posibles incumplimientos de SLA y optimizar proactivamente la asignación de recursos. Estos componentes se integran para crear un sistema dinámico que se adapta a las demandas cambiantes del entorno de red, con el objetivo de anticipar y evitar incumplimientos de SLA y optimizar la asignación de recursos de manera eficiente.

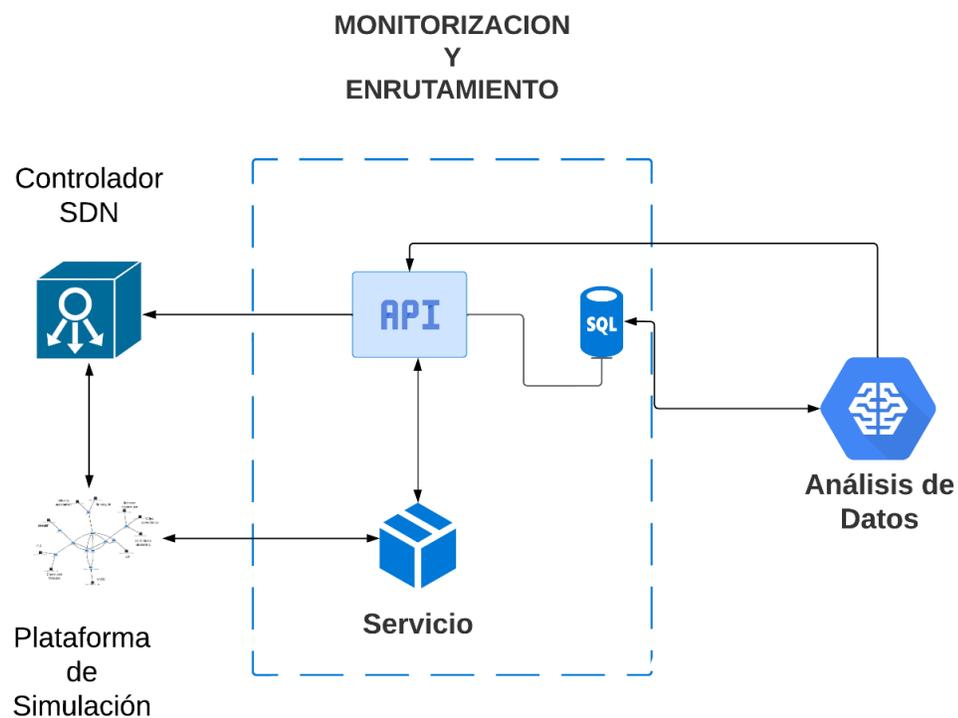


Figura 10. Esquema general propuesto.

La combinación de SDN, Mininet, el sistema de monitorización, la API de enrutamiento, y las técnicas de ML permiten al sistema recopilar datos de la red, analizarlos, tomar decisiones informadas y ajustar la configuración de la red en consecuencia, todo de manera automatizada y adaptativa. Este enfoque integrado crea un ciclo de retroalimentación continuo que mejora constantemente el rendimiento y la eficiencia de la red, reduciendo la necesidad de intervención manual y permitiendo una respuesta más rápida a las condiciones cambiantes. La naturaleza adaptativa del sistema le permite aprender de sus propias acciones y resultados, refinando sus estrategias de gestión con el tiempo. Además, esta integración

tecnológica facilita una gestión proactiva de los SLA, anticipando posibles problemas antes de que afecten el servicio. Por ejemplo, el ML puede identificar patrones precursores de incumplimientos de SLA, permitiendo que el sistema tome medidas preventivas, las cuales pueden ser implementadas rápidamente gracias a la flexibilidad de SDN y probadas de manera segura en el entorno Mininet. Esta integración no solo mejora el cumplimiento de los SLA, sino que también optimiza el uso de recursos de red, reduce los costos operativos y mejora la experiencia del usuario.

3.2 Desarrollo de la propuesta

3.2.1 Plataforma de simulación

Las plataformas de simulación desempeñan un papel crucial al permitir a los investigadores y desarrolladores evaluar y probar nuevas tecnologías, protocolos y arquitecturas de forma controlada y sin afectar los entornos de producción. Existen diversas plataformas de simulación disponibles, cada una con sus propias características, capacidades y enfoques. A continuación, se presentan algunas de las plataformas más populares y se justifica la elección de Mininet para la simulación de la propuesta (Gonzalez-Franco et al., 2023).

Plataformas de simulación:

- ✓ NS-3 (*Network Simulator 3*): es un simulador de redes de código abierto ampliamente utilizado en investigación y educación. Ofrece un conjunto completo de modelos de protocolos y dispositivos de red, lo que lo convierte en una herramienta poderosa para simular redes cableadas e inalámbricas (Network Simulator 3, 2024)
- ✓ OMNET++: es un entorno de simulación modular y basado en componentes, principalmente utilizado para simular redes de comunicaciones, sistemas distribuidos y sistemas de colas. Ofrece una interfaz gráfica de usuario (GUI) y es compatible con varios *frameworks*, como INET(OMNet, 2024).
- ✓ GNS3 (*Graphical Network Simulator 3*): es una plataforma de simulación gráfica que permite emular

redes completas, incluyendo enrutadores, conmutadores y otros dispositivos de red reales. Es ampliamente utilizado por profesionales y estudiantes para practicar configuraciones de red (GNS3, 2024).

- ✓ Mininet: es una plataforma de emulación de redes de código abierto que permite crear redes virtuales realistas en una sola máquina. Utiliza el kernel de Linux y el hipervisor de red *Open vSwitch* para emular hosts, conmutadores, enrutadores y enlaces (Mininet GitHub, 2021).

Para la simulación de la propuesta, se eligió Mininet debido a varias razones clave:

- ✓ Integración con SDN: se integra de manera fluida con controladores de Redes Definidas por Software (SDN), lo que lo convierte en una herramienta ideal para probar y evaluar arquitecturas SDN (Mininet, 2024).
- ✓ Entorno realista: utiliza el kernel de Linux y OVS para crear un entorno de red virtual realista, lo que permite evaluar el comportamiento de protocolos y aplicaciones de manera precisa.
- ✓ Facilidad de uso: ofrece una interfaz de programación de aplicaciones (API) desarrollada en Python, lo que facilita la creación, configuración y personalización de topologías de red complejas.
- ✓ Baja sobrecarga: al ejecutarse en una sola máquina, Mininet tiene una sobrecarga de recursos relativamente baja en comparación con otras plataformas de simulación, lo que lo hace adecuado para pruebas y experimentación rápida.
- ✓ Flexibilidad: permite emular diferentes tipos de dispositivos de red, como hosts, conmutadores, enrutadores y enlaces, lo que brinda una gran flexibilidad para simular una amplia gama de escenarios de red (Mininet, 2024).
- ✓ Compatibilidad con sistemas operativos basados en Linux: es compatible con la mayoría de las distribuciones de Linux, lo que facilita su implementación en diversos entornos.

La elección de Mininet como plataforma de simulación se basa en su integración con SDN, su capacidad para proporcionar un entorno de red realista, su facilidad de uso, su baja sobrecarga de recursos y su flexibilidad para simular diversos escenarios de red. Estas características hacen de Mininet una herramienta

poderosa y adecuada para la simulación y evaluación de la propuesta presentada (Mininet GitHub, 2021). Además, su amplia comunidad de usuarios y desarrolladores garantiza un soporte continuo y actualizaciones frecuentes, lo que contribuye a su evolución y mejora constante.

3.2.2 Controlador SDN

En el contexto de las Redes Definidas por Software (SDN), la elección del controlador adecuado es un factor crítico para una gestión efectiva de la infraestructura de red. Existen varios controladores SDN prominentes en el mercado, cada uno con sus propias fortalezas y debilidades. A continuación, se presenta una exploración de los principales controladores y se justifica la elección del controlador OpenDaylight (ODL) para la implementación de la propuesta (Ryait y Sharma, 2023).

- ✓ *Ryu* es un controlador SDN de código abierto desarrollado en Python. Si bien es versátil y permite diseñar aplicaciones de control de red, presenta un rendimiento inferior en comparación con otros controladores. Además, carece de independencia respecto a hardware específico y no se utiliza ampliamente en aplicaciones del mundo real (Ryu, 2024).
- ✓ OpenDaylight (ODL) es una iniciativa de código abierto que actúa como una plataforma colaborativa para mejorar SDN. Es un controlador modular y altamente disponible que soporta múltiples estándares y protocolos SDN. Facilita el desarrollo de aplicaciones de análisis y la definición de políticas, siendo ampliamente utilizado en centros de datos (OpenDaylight, 2024)
- ✓ ONOS (por las siglas del término en inglés, *Open Network Operating System*) es un controlador de alto rendimiento diseñado para redes de telecomunicaciones. Cuenta con un núcleo distribuido con replicación, lo que le permite escalar de manera eficiente en entornos grandes y complejos. ONOS admite la interacción a través de CLI, GUI y API REST, y es independiente de protocolos específicos. Además, es compatible con múltiples versiones de OpenFlow (Open Network Operating System, 2024).
- ✓ *POX* es una plataforma SDN desarrollada en Python que permite conectar conmutadores mediante *OpenFlow* y *OVSDB* (por las siglas del término en inglés, *Open vSwitch Database*). Facilita el desarrollo de controladores en Python y cuenta con un servidor RADIUS y un controlador de

autenticación, lo que la hace útil para prevenir ataques como DoS (Denegación de Servicio) y suplantación de identidad (POX, 2024).

- ✓ *Floodlight* es un controlador SDN de código abierto basado en Java que regula el tráfico utilizando *OpenFlow*. Sin embargo, carece de autenticación en la migración de hosts, lo que puede permitir la suplantación de identidad. Para mitigar este problema, utiliza perfiles de host con índices de direcciones Control de Acceso al Medio (MAC, por las siglas del término en inglés, *Media Access Control*), Protocolo de internet (IP, por las siglas del término en inglés, *Internet Protocol*), Red Virtual de Área Local (VLAN, por las siglas del término en inglés *Virtual Local Area Network*), entre otros (Floodlight, 2024)

Al comparar los controladores analizados, todos los proyectos son de código abierto. Sin embargo, ODL y ONOS son controladores distribuidos, lo que les permite escalar de manera más eficiente en entornos de mayor tamaño y complejidad. Además, presentan una mejor disponibilidad, documentación, frecuencia de actualización y una comunidad más sólida (OpenDaylight, 2024).

La elección de OpenDaylight (ODL) como plataforma para implementaciones de Redes Definidas por Software (SDN) se justifica por varias razones clave:

- ✓ **Modularidad:** la arquitectura modular de ODL permite una adaptación específica a las necesidades del usuario al proporcionar módulos independientes y personalizables.
- ✓ **Flexibilidad:** ofrece la capacidad de ajustar y configurar estos módulos según los requisitos del entorno de red, lo que brinda una gran flexibilidad.
- ✓ **Interoperabilidad:** destaca por su fuerte interoperabilidad, al ser compatible con diversos controladores SDN y protocolos estándar de la industria.
- ✓ **Versatilidad:** la plataforma tiene características, que abarcan desde la programación de flujos hasta la gestión de políticas, convirtiéndolo en una opción integral para varias aplicaciones SDN.
- ✓ **Solidez:** se respalda en una comunidad activa y en su cumplimiento de estándares abiertos, garantizando un rendimiento confiable y sostenible en entornos de red de distintos tamaños y complejidades.

En conjunto, estas cualidades hacen de ODL una elección estratégica para aquellos que buscan una plataforma SDN robusta y adaptable, capaz de satisfacer las demandas de entornos de red complejos y en constante evolución. Además, su arquitectura modular permite una integración fluida con diversas herramientas y tecnologías emergentes, proporcionando una mayor flexibilidad y escalabilidad en la gestión de redes (Shaho Ismael Hassen, 2023).

3.2.2.1 Asignación de prioridades en ODL

La asignación de prioridades en OpenDaylight (ODL) desempeña un papel fundamental en la gestión eficiente del tráfico de red para diversas aplicaciones en entornos de red avanzados. ODL utiliza estas prioridades para determinar cómo se procesarán los paquetes de diferentes aplicaciones en la red SDN. OpenDaylight maneja las prioridades de flujo en un rango de 0 a 65535, donde 0 es la prioridad más baja y 65535 la más alta. En el contexto de una implementación típica, las prioridades asignadas a diferentes aplicaciones suelen situarse en un rango específico dentro de este espectro, lo que permite flexibilidad para futuras adiciones tanto de mayor como de menor prioridad (Nam Nguyen et al., 2016) .

Cuando se configuran las rutas en el controlador ODL, las prioridades asignadas se traducen directamente en reglas de flujo *OpenFlow*. Esto significa que cada aplicación o tipo de tráfico puede tener una prioridad específica en la red, influyendo en cómo se manejan sus paquetes.

La asignación de prioridades influye directamente en cómo ODL maneja el tráfico:

- ✓ Orden de evaluación: ODL evalúa las reglas de flujo en orden descendente de prioridad. El tráfico con mayor prioridad se evalúa antes que el de menor prioridad.
- ✓ Resolución de conflictos: si un paquete coincide con múltiples reglas de flujo, se aplicará la regla con la prioridad más alta. Esto asegura que el tráfico más crítico reciba el tratamiento adecuado (Nam Nguyen et al., 2016).
- ✓ Gestión de recursos: en situaciones de congestión, ODL utilizará estas prioridades para asignar recursos de red. Las aplicaciones con prioridades más altas recibirán preferencia en términos de ancho de banda y procesamiento.

Un sistema bien diseñado de asignación de prioridades proporciona varias ventajas:

- ✓ Espacio para expansión: al utilizar un rango específico dentro del espectro total de ODL, se deja espacio para agregar nuevas aplicaciones con prioridades más altas o bajas en el futuro.
- ✓ Granularidad: una diferenciación adecuada entre niveles de prioridad permite una distinción clara entre aplicaciones, facilitando la gestión y el ajuste fino de las políticas de red.
- ✓ Adaptabilidad: las prioridades pueden ajustarse fácilmente dentro del rango establecido para reflejar cambios en las necesidades del negocio o en la importancia relativa de las aplicaciones (OpenDaylight, 2024).

Esta implementación de prioridades en ODL, combinada con la funcionalidad de API para configurar dinámicamente las rutas, permite una gestión de red altamente flexible y eficiente. Es capaz de adaptarse a las demandas cambiantes de las diversas aplicaciones en un entorno de red moderno, asegurando que los servicios más críticos reciban los recursos necesarios mientras se mantiene un funcionamiento eficiente de aplicaciones menos críticas (OpenDaylight, 2024).

3.3 Monitorización y enrutamiento

El servicio de monitorización es un componente fundamental del sistema propuesto para la gestión de Acuerdos de Nivel de Servicio (SLA). Su principal objetivo es monitorizar de forma continua el rendimiento de la red y recopilar métricas para verificar el cumplimiento de los SLA.

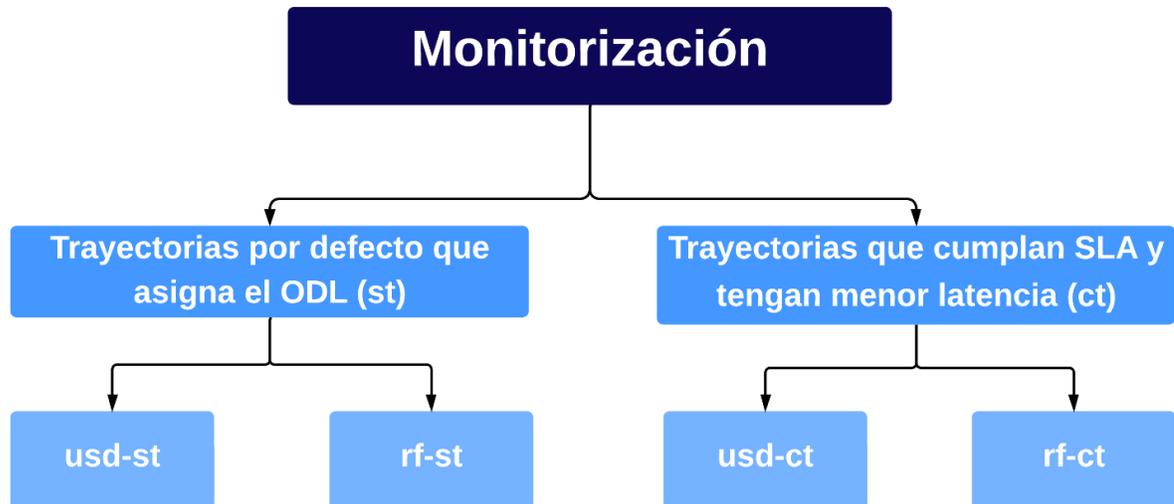


Figura 11. Estrategia de monitorización.

Como se ilustra en la Figura 11, el sistema implementa un enfoque de monitorización dual, diseñado para proporcionar una evaluación exhaustiva del rendimiento de la red. Este enfoque se divide en dos fases distintas:

1. Trayectorias por defecto que asigna el ODL (st): representa las rutas de red asignadas automáticamente por el controlador OpenDaylight (ODL).
2. Trayectorias que cumplan SLA y tengan menor latencia (ct): ilustra las rutas optimizadas que cumplen con los acuerdos de nivel de servicio (SLA) y ofrecen la menor latencia.

Cada rama se subdivide en dos categorías cruciales:

- ✓ usd: corresponde a la Monitorización Individual. En esta fase, el sistema evalúa meticulosamente el rendimiento de cada aplicación de manera secuencial. Se genera tráfico desde el UE individualmente a cada aplicación, con un intervalo de 1 minuto entre cada usuario. Este enfoque permite un análisis detallado y específico de cómo la red responde a las demandas particulares de cada servicio.
- ✓ rf: representa la Monitorización Grupal. Esta fase implica un análisis simultáneo de todas las aplicaciones. El sistema simula el tráfico de nueve usuarios (UE-9), uno por aplicación, operando

concurrentemente. Este método proporciona una visión holística del comportamiento de la red bajo condiciones de carga múltiple y variada.

Para simplificar cada una de estas monitorizaciones en los resultados se identifican de la siguiente forma:

- ✓ Monitorización individual a cada aplicación con las trayectorias por defecto asignadas por el controlador: usd-st.
- ✓ Monitorización individual a cada aplicación con la trayectoria que cumple con el SLA y presenta menor latencia: usd-ct.
- ✓ Monitorización grupal o en ráfagas a todas las aplicaciones con las trayectorias por defecto asignadas por el controlador: rf-st.
- ✓ Monitorización grupal o en ráfagas a todas las aplicaciones con la trayectoria que cumple con el SLA y presenta menor latencia: rf-ct.

La estructura simétrica de la Figura 11 indica un enfoque equilibrado y exhaustivo en la recolección de datos. Se recopilaban múltiples muestras en cada escenario de simulación, lo que resulta en un número significativo de mediciones en nuestra bitácora, distribuidas equitativamente entre los tipos de simulación y las fases de monitorización. Esta extensa recopilación de datos proporciona una base estadística robusta para el análisis. Esto permite una comparación directa entre las trayectorias por defecto y las optimizadas, así como entre los métodos de monitorización individual y grupal.

Este diseño de monitorización dual (individual y grupal) se aplica tanto a las trayectorias por defecto como a las optimizadas, ofreciendo una evaluación completa del rendimiento de la red en diversos escenarios. La combinación de estos enfoques proporciona una comprensión profunda de cómo la red maneja tanto las demandas específicas de aplicaciones individuales como las cargas de trabajo complejas y variadas en un entorno multiusuario. Este enfoque es esencial para mantener servicios de alta calidad en redes complejas, proporcionando información crucial para la optimización y gestión continua de la red, lo que permite una adaptación eficaz a las cambiantes demandas de los usuarios y aplicaciones.

3.3.1 Parámetros de monitorización para evaluar el cumplimiento del SLA

Se extraen de la monitorización cuatro parámetros fundamentales para verificar el cumplimiento de los SLA:

- ✓ **Latencia:** es el tiempo que tarda un paquete de datos en viajar desde su origen hasta su destino en una red. Se mide generalmente en milisegundos (ms) y es crucial para aplicaciones en tiempo real como la videoconferencia, el *gaming* en línea y la transmisión de voz sobre IP (VoIP). Una latencia baja, generalmente menor a 50 ms, es necesaria para garantizar una comunicación fluida y rápida. Latencias entre 50 ms y 100 ms se consideran medias y pueden ser aceptables para la mayoría de las aplicaciones, aunque pueden afectar la experiencia en tiempo real. Una latencia superior a 100 ms se considera alta y puede causar retrasos perceptibles que afectan negativamente la fluidez de la comunicación (Kurose y Ross, 2017).
- ✓ **Jitter:** es la variabilidad en el tiempo de llegada de los paquetes de datos, es decir, la fluctuación en la latencia de los paquetes que se envían a través de la red. Un *jitter* bajo, de menos de 20 ms, es ideal para aplicaciones en tiempo real, asegurando una experiencia estable. Entre 20 ms y 50 ms es medio y puede causar leves fluctuaciones en la calidad del servicio y superior a 50 ms se considera alto y puede provocar interrupciones notables en aplicaciones como videollamadas o VoIP, afectando la consistencia de la calidad del servicio (Kurose y Ross, 2017).
- ✓ **Ancho de banda:** es la capacidad máxima de transferencia de datos de una red, generalmente medida en bits por segundo (bps), megabits por segundo (Mbps) o gigabits por segundo (Gbps). Un ancho de banda bajo, por debajo de 10 Mbps, es adecuado para navegación web básica o transmisión de videos de baja resolución. Anchos de banda entre 10 Mbps y 100 Mbps son considerados medios, suficientes para transmisión de video HD y múltiples usuarios conectados. Un ancho de banda superior a 100 Mbps se considera alto, ideal para transmisión de video 4K, grandes transferencias de datos y aplicaciones que requieren un tráfico de red intenso (Kurose y Ross, 2017).
- ✓ **Pérdida de paquetes:** ocurre cuando uno o más paquetes de datos que se transmiten a través de una red no llegan a su destino. Se mide como un porcentaje respecto al total de paquetes enviados. Una pérdida de paquetes baja, inferior al 1%, generalmente es imperceptible en la mayoría de las aplicaciones, incluida la transmisión de video y audio. Una pérdida entre el 1% y el 5% se considera

media y puede afectar la calidad de aplicaciones en tiempo real, provocando pequeños cortes o degradación de la calidad. Una pérdida superior al 5% es alta y puede causar una degradación significativa, con cortes frecuentes en audio y video, y problemas de estabilidad en aplicaciones (Kurose y Ross, 2017).

Estos parámetros son fundamentales para entender y optimizar el rendimiento de la red, y buscar se cumplan los requisitos de calidad de servicio, los SLA y se mantenga una experiencia de usuario satisfactoria.

3.3.2 Comportamiento del tráfico por aplicación

1. **Experiencia Inmersiva:** el tráfico de experiencias inmersivas en 5G se caracteriza por flujos de datos intensivos y constantes, requiriendo un alto ancho de banda (50-55 Mbps) para transmitir contenido de realidad virtual y aumentada de alta calidad. Este tráfico es sensible a la latencia, requiere de valores muy bajos (alrededor de 7 ms) para mantener la sincronización y prevenir mareos en los usuarios. Utiliza paquetes grandes (1200 bytes) y el protocolo UDP para optimizar la transmisión en tiempo real. El tráfico puede ser altamente variable, con picos repentinos durante cambios de escena o interacciones del usuario, lo que demanda una red adaptable y con capacidad de priorización dinámica (Siddiqui et al., 2023).
2. **Videovigilancia:** este tipo de tráfico en redes 5G se caracteriza por un flujo constante de datos de video, con un ancho de banda moderado pero estable (10-12 Mbps). Este tráfico requiere una pérdida de paquetes extremadamente baja (0.001%) para mantener la calidad de imagen y evitar la pérdida de información crítica. La latencia moderada (10 ms) es importante para permitir el monitoreo en tiempo real y la respuesta rápida a incidentes. El tráfico puede experimentar picos repentinos durante eventos de interés o en situaciones de alerta, lo que requiere que la red tenga la capacidad de escalar rápidamente el ancho de banda asignado (Usman et al., 2015).
3. **eHealth:** este tráfico en 5G es crítico y diverso, abarcando desde la transmisión constante de signos vitales hasta grandes volúmenes de datos de imágenes médicas. Requiere un ancho de banda moderado pero consistente (16-20 Mbps) y una fiabilidad extremadamente alta, con una pérdida de paquetes prácticamente nula (0.00000001%). La latencia ultra baja (1 ms) es crucial para

aplicaciones como la telemedicina y la cirugía remota. El tráfico puede ser impredecible, con ráfagas intensas durante emergencias médicas, lo que exige una red altamente adaptable y con mecanismos de priorización instantánea (Ahad et al., 2019).

4. Smart Grid: este tráfico se caracteriza por ser ligero pero crítico, con un ancho de banda relativamente bajo (1-1.5 Mbps) pero con necesidad de alta confiabilidad. Requiere una pérdida de paquetes muy baja (0.0001%) y una latencia reducida (5ms) para permitir el control en tiempo real de la infraestructura eléctrica. El tráfico típicamente consiste en pequeños paquetes de datos (64 bytes) enviados con alta frecuencia, representando mediciones y comandos de control. Puede haber picos de tráfico durante eventos de la red eléctrica, como cortes o cambios repentinos en la demanda de energía (Wang et al., 2018).
5. Automatización Industrial: el tráfico de automatización industrial en 5G se caracteriza por ser constante, pero de bajo volumen, con un ancho de banda de 1-3 Mbps. Requiere una fiabilidad extremadamente alta, con una pérdida de paquetes prácticamente nula (0.0000001%) y una latencia ultra baja (1 ms) para soportar procesos industriales críticos y control en tiempo real. El tráfico típicamente consiste en paquetes pequeños a medianos (500 bytes) enviados a intervalos regulares. Puede haber ráfagas de tráfico durante eventos específicos de producción o en respuesta a anomalías detectadas en los procesos industriales (Li et al., 2017).
6. Vehículos Conectados: este tráfico es dinámico y crítico para la seguridad, requiere un ancho de banda moderado (10-15 Mbps) con baja latencia (3 ms) y pérdida de paquetes (0.001%). Se caracteriza por transmisiones frecuentes de paquetes pequeños a medianos (300 bytes) que contienen información de posición, velocidad y estado del vehículo. El tráfico puede experimentar picos repentinos durante situaciones de emergencia o en áreas de alta densidad de vehículos. La naturaleza móvil de este tráfico requiere una gestión de red altamente dinámica y eficiente en el traspaso entre células (Storck y Duarte-Figueiredo, 2019).
7. Sistemas de Transporte Inteligente (ITS): se caracteriza por ser intermitente pero crucial, con un ancho de banda relativamente bajo (0.5-1 Mbps). Requiere una latencia moderada (10 ms) y una pérdida de paquetes aceptable (0.1%). El tráfico típicamente consiste en paquetes pequeños (64 bytes) enviados a intervalos regulares, representando datos de sensores de tráfico, señalización y

gestión de infraestructura. Puede haber aumentos significativos en el volumen de tráfico durante horas pico o eventos especiales que afecten al transporte urbano (Talavera et al., 2018).

8. **Voz sobre 5G (Vo5G):** este tráfico se caracteriza por ser constante durante las llamadas, con ráfagas de actividad seguidas por períodos de silencio. Típicamente, requiere un ancho de banda mucho menor, alrededor de 64-128 Kbps para voz de alta calidad, dependiendo del códec utilizado. Puede tolerar cierta latencia (preferiblemente menos de 150 ms) y *jitter* (hasta 30 ms) sin afectar significativamente la calidad de la llamada. Utiliza paquetes pequeños (generalmente de 20 a 160 bytes) enviados a intervalos regulares. La pérdida de paquetes debe mantenerse muy baja (preferiblemente menos del 1%) para preservar la inteligibilidad. El tráfico de voz es sensible a la congestión de la red y requiere mecanismos de priorización para mantener la calidad de servicio durante períodos de alta demanda (Pocovi et al., 2018).

9. **Streaming de Video UHD:** este tráfico de video UHD en 5G se caracteriza por flujos de datos intensivos y constantes, requiriendo un alto ancho de banda (10-20 Mbps) para mantener la calidad de imagen. Tolera cierta pérdida de paquetes (1%) debido a los mecanismos de corrección de errores incorporados en los *codecs* de video modernos. Requiere una latencia relativamente baja (4 ms) para permitir una experiencia de visualización fluida, especialmente en aplicaciones interactivas. Utiliza paquetes grandes (1200 bytes) para optimizar la eficiencia de la transmisión. El tráfico puede experimentar variaciones significativas dependiendo del contenido del video, con picos durante escenas de alta acción o complejidad visual (Nightingale et al., 2018).

3.3.3 Generación de tráfico y obtención de métricas

La generación de tráfico en el sistema de monitorización es un componente crucial para evaluar el rendimiento de la red y asegurar que se cumplan los Acuerdos de Nivel de Servicio (SLA). Este proceso implica simular el comportamiento de diversas aplicaciones y servicios a través de herramientas como *Ping* e *Iperf*. *Ping*, una herramienta de diagnóstico en redes de computadoras, se utiliza para probar la conectividad entre dos nodos en una red. Utiliza el protocolo ICMP (por las siglas en inglés, *Internet Control Message Protocol*) para enviar mensajes de solicitud de eco y recibir mensajes de respuesta de eco (Paessler, 2024). El proceso de *Ping* incluye el envío de una solicitud de eco desde un nodo usuario a los nodos destino, la recepción de respuestas de eco por parte de estos nodos y la medición del tiempo de ida

y vuelta (RTT, por las siglas del término en inglés, *Round Trip Time*) para determinar la latencia, el *jitter* y la pérdida de paquetes. Estos resultados se registran en un archivo de log para su análisis.

Por otro lado, *Iperf* es una herramienta de medición de ancho de banda en redes que puede usar tanto los protocolos de transporte TCP (por las siglas del término en inglés, *Transmission Control Protocol*) como UDP (por las siglas del término en inglés, *User Datagram Protocol*) para evaluar el rendimiento de la red entre dos nodos (Iperf, 2024). El proceso con *Iperf* incluye la configuración de los nodos destino como servidores y el nodo usuario como cliente, la transferencia de datos para medir la capacidad del enlace de red, y la medición y análisis del ancho de banda, cuyos resultados también se registran en un archivo de log para su análisis.

El sistema de monitorización está diseñado para simular tráfico para diversas aplicaciones y servicios típicos en redes avanzadas como 5G y B5G. Para cada tipo de aplicación, se definen perfiles de tráfico específicos que incluyen parámetros como el tamaño de los paquetes, la frecuencia de envío y el ancho de banda requerido. Las pruebas se ejecutan desde un nodo central (cliente UE) hacia los demás nodos (servidores) en la red, que utilizan las herramientas *Ping* e *Iperf* para generar el tráfico y medir los parámetros de rendimiento. Los resultados de estas pruebas se almacenan en archivos de texto y se registran en un archivo de log, que incluye datos sobre latencia, *jitter*, ancho de banda y pérdida de paquetes. Los datos recopilados se comparan con los requisitos del SLA para asegurar el rendimiento adecuado de la red. La generación de tráfico realista en sistemas de monitorización de redes es crucial para evaluar con precisión el rendimiento y asegurar el cumplimiento de los SLA. Si los resultados no cumplen con los estándares, se realizan ajustes en la configuración de la red para mejorar el rendimiento y buscar asegurar el cumplimiento de los SLA.

Este proceso simula condiciones operativas reales, permitiendo identificar problemas potenciales, optimizar la configuración de la red y validar su desempeño bajo diferentes cargas de trabajo. Además, facilita la detección de cuellos de botella, la optimización de recursos y la preparación para escenarios reales. En última instancia, esta práctica es esencial para mantener una alta calidad de servicio, cumplir con los estándares establecidos y realizar ajustes informados que mejoren la eficiencia y capacidad de respuesta de la red (Karagiannis et al., 2004).

3.3.4 API de enrutamiento

La API de enrutamiento permite una gestión eficiente del tráfico de red a través de la comunicación con el controlador OpenDaylight mediante sus API RESTful. Estas API exponen diferentes *endpoints* que interactúan con distintos servicios y módulos del controlador, realizando solicitudes HTTP (GET, PUT, POST, DELETE) y utiliza formatos como JSON o XML para el intercambio de datos. La API de enrutamiento recolecta y analiza datos de monitorización de la red, optimiza dinámicamente las rutas de tráfico para asegurar el cumplimiento de los acuerdos de nivel de servicio (SLA) establecidos. Mediante ajustes en tiempo real de las rutas, la API garantiza que los flujos de tráfico se encaminen por las vías que satisfagan los requisitos de los SLA y presenten menor latencia, maximizando así el rendimiento y la eficiencia de la red.

Proceso de Enrutamiento:

- ✓ Carga de Datos: la API carga la bitácora obtenida del sistema de monitorización en formato JSON en una base de datos SQL, permite un almacenamiento estructurado y accesible de los datos de rendimiento.
- ✓ Revisión de Datos: la API realiza una exploración de los datos almacenados en la base de datos y busca entradas donde SLA=NO, identificando los nodos de la red que no cumplen con los SLA.
- ✓ Selección de Rutas: la API revisa los datos de rendimiento de todas las trayectorias y selecciona, de entre todas las rutas disponibles que cumplen con los SLA, aquella con la menor latencia para cada aplicación específica. Esto se realiza mediante el método *épsilon-restricción* que se describe en el artículo (Lozano-Rizk et al., 2022), el cual filtra primero el conjunto de rutas factibles que satisfacen los requisitos de SLA establecidos para esa aplicación. Luego, del subconjunto de rutas que pasaron el filtro, se elige la de menor retardo o latencia. Seleccionar la ruta con la menor latencia es crucial para mejorar la calidad de la experiencia del usuario (QoE) en aplicaciones sensibles al tiempo, como el *Streaming* de Video UHD, videollamadas y juegos en línea. La latencia afecta directamente la rapidez y respuesta de estas aplicaciones, garantizando una transmisión más fluida y sincronizada. Aunque el *jitter*, la pérdida de paquetes y el ancho de banda también deben cumplir con los valores del SLA, una ruta con menor latencia generalmente reduce el *jitter* y maximiza el uso eficiente del ancho de banda. Por lo tanto, priorizar la menor latencia asegura una experiencia de usuario óptima y minimiza interrupciones y retardos perceptibles. De esta manera, se mejora el

rendimiento de la red escogiendo la ruta más rápida que a la vez cumple con los umbrales definidos por los SLA para cada aplicación.

- ✓ Configuración de Rutas: en primer lugar, la API utiliza un método DELETE para eliminar las rutas configuradas de los flujos de tráfico en los switches. La API mediante un método POST configura las nuevas rutas seleccionadas del controlador SDN, que instala estas reglas de reenvío en las tablas de flujo de los switches correspondientes. De esta manera, se asegura que el tráfico de red se encamine automáticamente por las nuevas rutas, las cuales cumplen con los SLA establecidos y tienen una menor latencia punto a punto.

3.4 Integración de la monitorización y la API de enrutamiento

El esquema proporcionado en la Figura 12 muestra un flujo claro de cómo se integra la monitorización de red con la API de enrutamiento para asegurar el cumplimiento de los Acuerdos de Nivel de Servicio (SLA).

A continuación, se describe paso a paso cómo se lleva a cabo esta integración de forma general para el ejemplo de que el usuario solicite una aplicación de *Streaming* de Video UHD:

1. Solicitud de aplicación/servicio

Un usuario (UE) ha solicitado el servicio de *Streaming* de Video UHD. Esta solicitud inicia el proceso de creación de un segmento de red específico (NS) adaptado a las características y requisitos de este servicio, que incluyen un ancho de banda adecuado para la transmisión de video en alta definición.

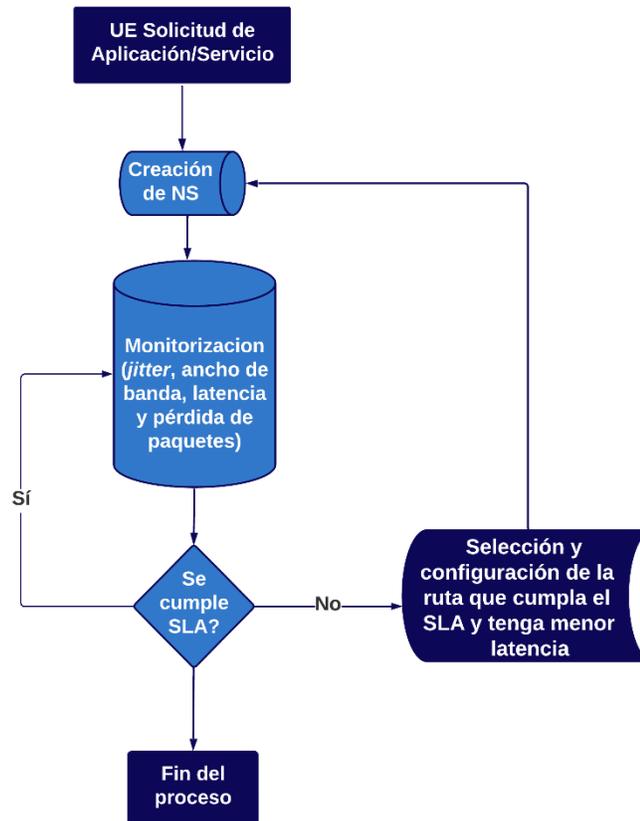


Figura 12. Proceso de monitorización y ajuste dinámico de rutas para cumplimiento de SLA en redes 5G/B5G.

2. Creación de NS (*Network Slice*)

En respuesta a la solicitud del usuario, se crea un segmento de red (NS) específico para *Streaming* de Video UHD. Este segmento se diseña para proporcionar los recursos necesarios que soporten la transmisión continua de video UHD con calidad adecuada.

3. Monitorización del tráfico

Una vez creado el NS, se inicia la monitorización del tráfico para el servicio solicitado. Se utilizan herramientas como *Ping* e *Iperf* para medir parámetros clave como son *jitter*, ancho de banda, latencia y pérdida de paquetes.

4. Evaluación del cumplimiento de SLA

Los datos recopilados durante la monitorización se comparan con los requisitos establecidos en el SLA para

el servicio de *Streaming* de Video UHD. Todos los parámetros deben cumplir con los valores límites definidos:

- Pérdida de paquetes: 1% máximo

- Latencia: 4 ms máximo

- *Jitter*: 5.84 ms máximo

- BitRate (Ancho de banda): 10 Mbps mínimo

5. Selección y configuración de rutas

Si alguno de los parámetros medidos no cumple con los requisitos del SLA, se activa la API de enrutamiento para seleccionar y configurar una nueva ruta. Esta API busca una ruta alterna que no solo cumpla con el SLA, sino que también ofrezca la menor latencia posible. En este caso, si la evaluación muestra que los valores actuales no son aceptables, se ajustarán las rutas de la red para mejorar los parámetros de rendimiento.

6. Fin del proceso

Una vez ajustadas las rutas y asegurado el cumplimiento de los SLA, el proceso de monitorización y ajuste continúa hasta que el usuario finaliza el servicio. Esto garantiza que el servicio de *Streaming* de Video UHD se esté entregando con la calidad requerida, proporcionando una experiencia de usuario óptima.

El proceso descrito gestiona eficazmente las solicitudes de *Streaming* de video UHD, abarcando desde la creación de segmentos de red hasta la monitorización y ajuste dinámico de rutas. Esta aproximación integra la monitorización continua con API de enrutamiento, asegurando el cumplimiento de los SLAs y una calidad de servicio óptima. El sistema evalúa constantemente los parámetros críticos para mantener los estándares de calidad definidos. Los beneficios clave incluyen un rendimiento mejorado de la red, una calidad de servicio superior que realza la experiencia del usuario, una gran adaptabilidad a las condiciones cambiantes de la red. En conjunto, este enfoque garantiza una entrega eficiente y de alta calidad de contenido de video UHD, adaptándose ágilmente a las demandas del servicio y las fluctuaciones de la red.

3.5 Predicción de incumplimientos de SLA mediante *Machine Learning*

Para la predicción de incumplimiento de SLA uno de los modelos más utilizados son las Redes Neuronales recurrentes (RNN, por las siglas del término en inglés, *Recurrent Neural Networks*) (Sayed et al., 2022).

Las Redes Neuronales Recurrentes (RNN) son un tipo de arquitectura de aprendizaje profundo diseñada para procesar datos secuenciales. Como se muestra en la Figura 13, una RNN consiste en una serie de unidades interconectadas que procesan información a lo largo del tiempo. Cada unidad RNN recibe una entrada (X) en un paso de tiempo específico y produce una salida (h), que también sirve como estado oculto. Lo crucial de las RNN es su capacidad para mantener información a través del tiempo, transmitiendo este estado oculto de un paso al siguiente.

Esto permite a la red recordar información previa y utilizarla para influir en las predicciones futuras. La imagen de la Figura 13, ilustra cómo esta estructura secuencial puede desplegarse en el tiempo, muestra múltiples pasos, pero también representa de forma compacta como un solo bloque RNN con una conexión recurrente. Esta arquitectura hace que las RNN sean efectivas para tareas con datos secuenciales, como procesamiento de lenguaje natural, reconocimiento de voz o predicción de series temporales.

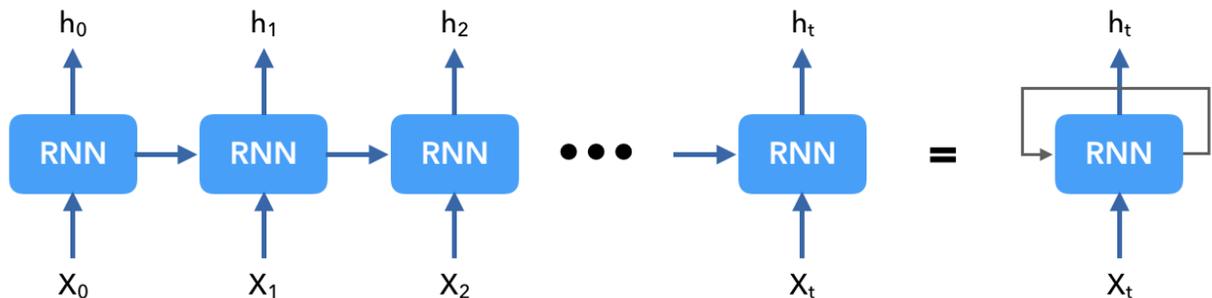


Figura 13. Redes Neuronales Recurrentes. Tomada de (León et al., 2022).

RNN es una herramienta valiosa en la predicción de patrones de tráfico en redes 5G debido a su capacidad para modelar secuencias temporales, adaptarse a datos cambiantes y optimizar recursos (León et al., 2022). Sus principales ventajas son:

- ✓ Modelado de secuencias temporales: son especialmente adecuadas para modelar datos secuenciales, como series temporales de tráfico en redes. En el caso de redes 5G, el tráfico de datos

puede variar con el tiempo, y las RNN pueden capturar patrones y tendencias en estas secuencias temporales.

- ✓ Memoria a corto y largo plazo: tienen la capacidad de mantener una memoria de lo que ha sucedido en las secuencias anteriores. Esto es esencial para predecir el tráfico en redes 5G, ya que el rendimiento de la red puede depender de eventos pasados, como congestiones o fluctuaciones en la demanda de datos (Karpathy et al., 2015).
- ✓ Flexibilidad en la arquitectura: permiten una mayor flexibilidad en la arquitectura de la red. Puedes diseñar RNN simples o más complejas, como las LSTM (por las siglas del término en inglés, *Long Short-Term Memory*) o las GRU (por las siglas del término en inglés, *Gated Recurrent Unit*), según las necesidades específicas del problema.
- ✓ Adaptabilidad a datos cambiantes: las redes 5G son redes altamente dinámicas con un tráfico que puede variar rápidamente. Pueden adaptarse a estos cambios y actualizar sus predicciones a medida que fluyen nuevos datos (Karpathy et al., 2015).
- ✓ Escalabilidad: se pueden entrenar y utilizar en escenarios de gran escala, lo que es importante en redes 5G, que deben manejar un gran volumen de datos y dispositivos conectados.
- ✓ Aplicaciones de optimización de recursos: además de predecir el tráfico, se pueden utilizar en la optimización de recursos en redes 5G. Por ejemplo, pueden ayudar a asignar de manera eficiente el ancho de banda y los recursos de la red en función de las predicciones de tráfico (Karpathy et al., 2015).

En los últimos años, la aplicación de redes LSTM para la predicción del tráfico ha sido objeto de numerosos estudios. Desde el trabajo pionero de (Lv et al., 2015) , que introdujo un enfoque de aprendizaje profundo basado en LSTM para predecir el flujo de tráfico, hasta investigaciones más recientes como la de (Wu et al., 2016), que incorpora información espacio-temporal en modelos LSTM para la predicción del tráfico urbano, se ha observado una evolución significativa en este campo. Los investigadores han abordado diversos aspectos, incluyendo la optimización de recursos en redes de telecomunicaciones (Troia et al., 2018), la predicción del tráfico en centros de datos (Fu et al., 2019), y la integración de técnicas de transferencia de aprendizaje (Vinayakumar et al., 2017). Además, revisiones exhaustivas como la de (Yuan y Li, 2021)han

proporcionado una visión general de los diversos métodos basados en LSTM utilizados en la predicción del tráfico, demostrando la creciente importancia y eficacia de esta técnica en el campo.

Las LSTM son una variante avanzada de las redes neuronales recurrentes (RNN) diseñadas para manejar eficazmente datos secuenciales y temporales. A diferencia de las RNN simples, como se ilustra en la Figura 14, las LSTM pueden recordar información relevante durante largos períodos de tiempo, lo que las hace adecuadas para procesar y predecir series temporales. Esta capacidad se debe a su arquitectura única que incluye celdas de memoria y tres puertas principales (*input*, *forget* y *output*). La estructura de una LSTM incluye cuatro redes neuronales y bloques de memoria llamados células. La *Forget Gate* decide qué información se descarta del estado de la célula; la *Input Gate* determina qué nueva información se añade; y la *Output Gate* extrae y presenta la información útil como salida. Estas puertas permiten a la red decidir qué información almacenar, olvidar o utilizar en cada paso de tiempo, lo que les confiere la capacidad de capturar dependencias a largo plazo en los datos y gestionar eficientemente la información, reteniendo lo importante y descartando lo innecesario. Además, las LSTM son menos sensibles a la longitud de las secuencias de datos en comparación con las RNN tradicionales (Alom et al., 2019).

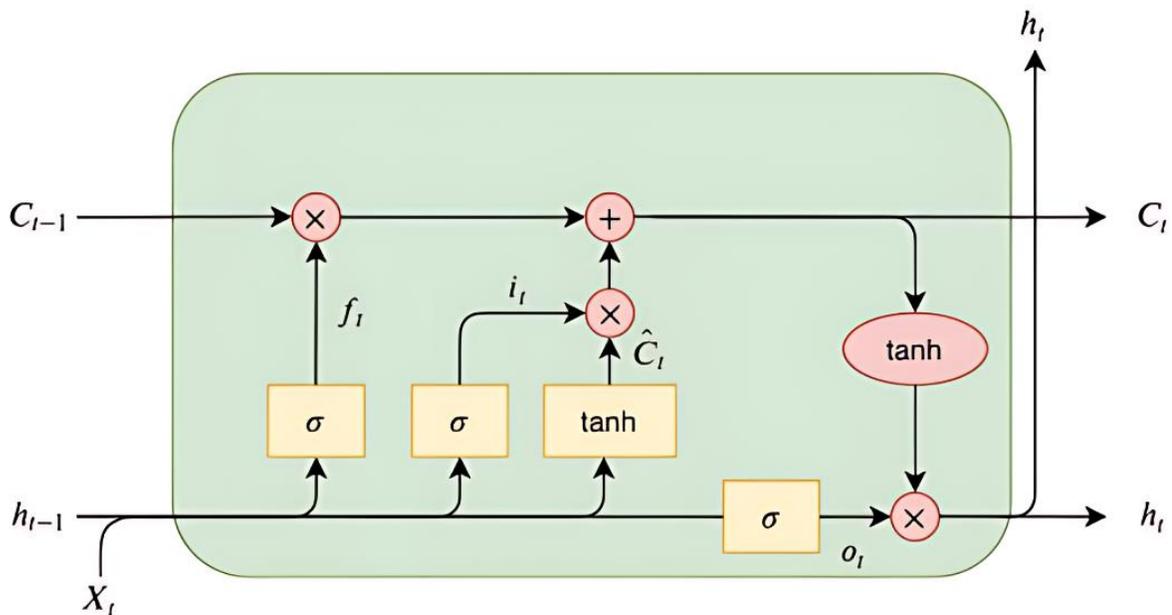


Figura 14. Modelo LSTM. Tomada de (Alom et al., 2019).

Para predecir el comportamiento de los SLA en nuestra propuesta, las LSTM son una excelente elección debido a su capacidad para manejar series temporales y patrones complejos a lo largo del tiempo. Los SLA suelen involucrar métricas que varían en el tiempo y pueden tener dependencias tanto a corto como a largo

plazo. Las LSTM pueden capturar estas relaciones temporales, aprender patrones estacionales, tendencias y anomalías en el rendimiento del sistema. Además, pueden manejar múltiples variables de entrada y salida, útil para sistemas complejos con indicadores de rendimiento. Su robustez frente al problema del desvanecimiento del gradiente las hace especialmente adecuadas para predecir comportamientos a largo plazo, crucial en la gestión y cumplimiento de SLA (Greff et al., 2017).

3.6 Implementación en la nube y selección de proveedor de servicio

Para que esta propuesta de sistema de monitorización y gestión de SLA en entornos de redes 5G/B5G pueda ser implementada en cualquier lugar y tenga la capacidad de escalarse de manera efectiva, es fundamental aprovechar las ventajas de la computación en la nube (*cloud computing*). Actualmente, existen varios proveedores líderes en el mercado de servicios de nube, cada uno con fortalezas y características distintivas. *Amazon Web Services (AWS)*, *Microsoft Azure* y *Google Cloud* se posicionan como los principales actores en este ámbito.

- ✓ AWS destaca por su extensa gama de servicios en la nube, su infraestructura global distribuida y su sólido conjunto de herramientas para el desarrollo, implementación y gestión de aplicaciones en la nube (Amazon Web Services, 2024)
- ✓ *Microsoft Azure* sobresale por su integración con otros productos de Microsoft, lo que facilita el desarrollo y la implementación de soluciones empresariales, además de ofrecer opciones de implementación híbrida (Azzure, 2024).
- ✓ *Google Cloud*, por otro lado, se distingue por su infraestructura de alto rendimiento y sus capacidades avanzadas en inteligencia artificial, aprendizaje de máquina y análisis de datos. Además, se integra de manera fluida con otras herramientas y servicios de Google, lo que puede ser beneficioso si se utilizan estos productos en el entorno de trabajo (Google Cloud, 2024).

Independientemente del proveedor de servicios de nube seleccionado, la implementación en la nube permitirá que nuestra propuesta sea escalable, flexible y accesible desde cualquier ubicación geográfica. Además, la computación en la nube ofrece beneficios adicionales, como la reducción de costos operativos, la alta disponibilidad, la seguridad reforzada y la capacidad de aprovisionamiento rápido de recursos según

sea necesario. Estas ventajas son cruciales para garantizar el correcto funcionamiento y la confiabilidad de un sistema de monitorización y gestión de SLA en entornos de redes de alta velocidad como las redes 5G/B5G (Amazon Web Services, 2024).

Se ha seleccionado Google Cloud por las ventajas que presenta:

- ✓ Infraestructura de alto rendimiento: se basa en la misma infraestructura de alto rendimiento y escalable que utiliza Google para sus propios productos y servicios. Esto garantiza un rendimiento sólido y confiable para las aplicaciones y servicios implementados en su plataforma.
- ✓ Inteligencia artificial y aprendizaje de máquina: destaca por sus capacidades avanzadas en inteligencia artificial (IA) y aprendizaje de máquina (ML). Ofrece servicios como *Cloud AI*, *TensorFlow*, *Vertex AI* y *Cloud Machine Learning Engine*, lo que facilita la integración de estas tecnologías en soluciones prácticas (Google Cloud, 2024).
- ✓ Análisis de datos proporciona herramientas potentes para el análisis de datos, como *BigQuery*, *Dataflow* y *Dataproc*. Estas herramientas permiten procesar y analizar grandes volúmenes de datos de manera eficiente, lo que puede ser fundamental para tu sistema de monitorización y análisis de rendimiento.
- ✓ Servicios de red: ofrece servicios de red escalables y de alta velocidad, como *Cloud Load Balancing*, *Cloud CDN* y *Cloud Interconnect*. Estos servicios pueden ser relevantes si tu sistema requiere una infraestructura de red robusta y distribuida.
- ✓ Seguridad y cumplimiento: cuenta con medidas de seguridad y cumplimiento de alto nivel, incluyendo cifrado de datos, autenticación y control de acceso, registros de auditoría y cumplimiento con estándares y regulaciones de la industria.
- ✓ Amplia gama de servicios: ofrece una amplia gama de servicios, como *Compute Engine* (máquinas virtuales), *Cloud Storage*, *Cloud SQL*, *Cloud Firestore*, entre otros, lo que te permite construir y escalar tu solución de manera integral (Google Cloud, 2024).
- ✓ Integración con otras herramientas de Google: se integra de manera fluida con otras herramientas y servicios de Google, como *G Suite*, *Android* y *Firebase*, lo que puede ser beneficioso si utilizas

estos productos en tu entorno de trabajo.

Al elegir *Google Cloud*, se tiene el beneficio de su infraestructura de alto rendimiento, sus capacidades en IA y ML, sus herramientas para análisis de datos, su robusta infraestructura de red, sus medidas de seguridad, cumplimiento, y su amplia gama de servicios. Estas características hacen de *Google Cloud* una plataforma sólida y adecuada para implementar tu propuesta de sistema de monitorización y gestión de SLA en entornos de redes 5G/B5G (Google Cloud, 2024).

3.7 Conclusiones parciales

En este capítulo, se abordó la importancia de garantizar el cumplimiento de los Acuerdos de Nivel de Servicio (SLA) en sistemas de comunicaciones modernos, especialmente en redes avanzadas como 5G/B5G. La naturaleza dinámica y compleja de estas redes exige soluciones innovadoras para la gestión y monitorización de SLA. Se presentó una propuesta integral que combina diversas tecnologías avanzadas, incluyendo Redes Definidas por Software (SDN), la plataforma de emulación de redes Mininet, un sistema de monitorización robusto, una API de enrutamiento eficiente y técnicas de Aprendizaje de Máquina (ML).

- ✓ La propuesta aprovecha las capacidades de programabilidad, control centralizado y abstracción de recursos que ofrecen las SDN, junto con la flexibilidad de Mininet para simular entornos de red complejos. La integración de un sistema de monitorización permite verificar continuamente el cumplimiento de los SLA, mientras que la API de enrutamiento facilita el redireccionamiento ágil del tráfico. La inclusión de técnicas de ML como LSTM añade una capa de inteligencia al sistema, que permite la anticipación de posibles violaciones de SLA y la optimización dinámica de la asignación de recursos.
- ✓ La arquitectura propuesta demuestra que la combinación de estas tecnologías puede proporcionar un sistema de gestión de SLA adaptativo y eficiente. Las SDN permiten la implementación de políticas de gestión flexibles y adaptables, mientras que Mininet proporciona un entorno de prueba realista y controlado. El sistema de monitorización y la API de enrutamiento trabajan en conjunto para optimizar el rendimiento de la red y asegurar el cumplimiento de los SLA. Además, las técnicas de ML permiten al sistema aprender de los patrones de tráfico y comportamiento de la red, mejorando proactivamente la calidad del servicio.

- ✓ La implementación de esta propuesta en la nube, en plataformas como *Google Cloud*, ofrece beneficios adicionales en términos de escalabilidad, flexibilidad y accesibilidad. La capacidad de responder rápidamente a las condiciones cambiantes de la red y la automatización de procesos garantizan un rendimiento óptimo y una calidad de servicio consistente para los usuarios finales.

La combinación de SDN, Mininet, técnicas de ML y la infraestructura en la nube crea un sistema robusto y adaptativo capaz de mantener altos estándares de rendimiento y calidad en entornos de red avanzados.

Capítulo 4. Validación de la propuesta

El presente capítulo aborda la implementación y validación de un sistema de gestión y monitorización de Acuerdos de Nivel de Servicio (SLA) en redes 5G, con tecnologías de Redes Definidas por Software (SDN). Este estudio se centra en la creación de un entorno de simulación realista que permite evaluar el desempeño del sistema propuesto en condiciones que reflejen las complejidades y demandas de las redes de telecomunicaciones avanzadas.

La investigación se desarrolla en tres etapas fundamentales: 1) la simulación y monitorización de una topología de red compleja, 2) el desarrollo de una API para la configuración dinámica de flujos de red, y 3) el análisis de datos para la predicción del comportamiento de los SLA. Se utilizan herramientas como Mininet para la simulación de la topología, OpenDaylight como controlador SDN, y una API personalizada desarrollada con Flask y Postman, se busca crear un ecosistema que permita una gestión eficiente y adaptativa de los recursos de red.

Este capítulo detalla la validación de la propuesta empleada, incluyendo la configuración del entorno de simulación, los parámetros de generación de tráfico para diferentes servicios 5G, y la implementación de un sistema de priorización de flujos basado en los requisitos específicos de cada aplicación. El análisis de los resultados obtenidos abarca cinco áreas clave: el comportamiento del SLA, las pérdidas de paquetes, la latencia, la eficiencia del sistema y la predicción de SLA. Estos aspectos proporcionan una evaluación integral del rendimiento y la efectividad del sistema propuesto en condiciones que emulan las complejidades de las redes 5G reales. El objetivo es ofrecer una comprensión profunda de cómo las tecnologías SDN pueden optimizar la gestión de SLA en redes 5G, mejorar la calidad de servicio y la eficiencia en la utilización de recursos de red. Los hallazgos de este análisis multifacético no solo validan la efectividad del enfoque propuesto, sino que también sientan las bases para futuras innovaciones en la gestión de redes avanzadas.

4.1 Escenario de validación

Para realizar la simulación de la propuesta del sistema de gestión y monitorización de SLA, se crea un entorno virtual. Es fundamental seleccionar el Entorno de Desarrollo Integrado (IDE) adecuado para las tareas de programación, la plataforma para el desarrollo del entorno virtual de la simulación, el software

para probar la API REST, el lenguaje de programación, el controlador SDN que gestionará el tráfico y el *framework* para desarrollar la API.

Para las tareas de programación, se utiliza el IDE de *Microsoft Visual Studio Code* (VSC) por las siguientes razones:

- ✓ Versatilidad y soporte multilenguaje: admite una amplia gama de lenguajes de programación, lo que lo hace adecuado para proyectos que requieren el uso de múltiples tecnologías. Esto permite a los desarrolladores trabajar en diferentes partes del proyecto dentro de un solo entorno (Visual Studio Code, 2024).
- ✓ Extensiones y *plugins*: cuenta con un ecosistema robusto de extensiones y *plugins* que permiten personalizar y mejorar la funcionalidad del IDE según las necesidades específicas del proyecto. Esto incluye soporte para herramientas de desarrollo web, integración con sistemas de control de versiones, depuración, y más.
- ✓ Integración con Git: la integración nativa con Git permite a los desarrolladores gestionar el control de versiones de manera efectiva, directamente desde el IDE, esto facilita la colaboración en equipo y el seguimiento de cambios en el código. Esta funcionalidad es crucial para mantener un flujo de trabajo organizado y eficiente.

Estas características clave hacen de *Microsoft Visual Studio Code* una herramienta poderosa y flexible, ideal para el desarrollo de proyectos complejos y diversos (Visual Studio Code, 2024).

Uno de los objetivos principales del sistema propuesto es su implementación en la nube. Para ello, se emplea *Google Cloud*, por sus beneficios analizados en el capítulo anterior, donde se crea una máquina virtual (VM) de tipo e2-medium (2 CPU virtuales, 1 núcleo y 4 GB de memoria) para realizar las simulaciones necesarias y validar el sistema.

Para el desarrollo e implementación de la API REST utilizamos Postman y Flask. Para interactuar con el controlador se utiliza Postman que es una herramienta de desarrollo de software que facilita la creación, prueba y documentación de API (Postman, 2024). Sus características y usos principales incluyen:

- ✓ Interfaz intuitiva: permite a los desarrolladores enviar solicitudes HTTP (GET, POST, PUT, DELETE,

etc.) a una API y ver las respuestas, todo a través de una interfaz gráfica de usuario. Facilita la creación y organización de solicitudes en colecciones, lo que permite gestionar diferentes entornos y configuraciones de manera eficiente.

- ✓ Automatización de pruebas: permite escribir y ejecutar scripts de prueba automatizados para validar las respuestas de la API, asegurando que funcionen según lo esperado. Ofrece capacidades de monitoreo y programación de pruebas, lo que ayuda a detectar problemas de rendimiento y disponibilidad (Postman, 2024).
- ✓ Documentación y colaboración: genera automáticamente documentación detallada de las API a partir de las colecciones de solicitudes creadas, facilitando la referencia y el uso por otros desarrolladores. Permite compartir colecciones y entornos con otros miembros del equipo, mejorar la colaboración y la comunicación entre desarrolladores y *testers*.

Por otro lado, Flask es un *microframework* de Python utilizado para desarrollar aplicaciones web. Es conocido por su simplicidad y flexibilidad (Flask, 2024). Entre sus ventajas se encuentran:

- ✓ Ligero y extensible: no impone muchas restricciones ni decisiones de diseño, permite a los desarrolladores elegir las herramientas y librerías que mejor se adapten a sus necesidades, lo que facilita el desarrollo rápido y la adaptación a diferentes tipos de proyectos.
- ✓ Facilidad de uso y aprendizaje: es sencillo de aprender y usar, siendo una excelente opción para desarrolladores que se inician en el desarrollo web. Su documentación es clara y completa, facilita la comprensión de sus funcionalidades y la implementación de proyectos.
- ✓ Soporte y comunidad: cuenta con una gran comunidad de desarrolladores que contribuyen con extensiones, tutoriales y soporte, lo que significa que los desarrolladores pueden beneficiarse de una amplia gama de recursos y soluciones preexistentes para resolver problemas comunes y mejorar sus aplicaciones (Flask, 2024).

Al utilizar Postman y Flask conjuntamente en la API de enrutamiento, se puede lograr una integración eficiente y fluida. Postman facilita la creación y prueba de solicitudes HTTP para verificar las rutas y respuestas del API, mientras que Flask proporciona un entorno flexible y ligero para el desarrollo del *backend* de enrutamiento. Esta combinación permite una rápida iteración y resolución de problemas,

garantizan que la API de enrutamiento funcione correctamente y cumpla con los requisitos esperados (Postman, 2024; Flask, 2024).

Se usa Python como lenguaje de programación, ya que es muy utilizado y popular en el desarrollo de software (Python, 2024). Entre sus principales ventajas se encuentran:

- ✓ Sintaxis sencilla y legible: la sintaxis de Python es clara y fácil de leer, lo que facilita el aprendizaje y la escritura de código limpio y comprensible. Esto mejora la productividad de los desarrolladores y la mantenibilidad del código.
- ✓ Amplia biblioteca estándar y ecosistema de paquetes: cuenta con una vasta biblioteca que proporciona módulos y funciones para prácticamente cualquier tarea. Además, tiene un ecosistema robusto de paquetes y *frameworks* (como *Django*, *Flask*, *NumPy*, *Pandas*) que permiten desarrollar aplicaciones web, científicas, de análisis de datos, inteligencia artificial y más de manera rápida y eficiente.
- ✓ Comunidad activa y soporte extenso: tiene una comunidad de desarrolladores muy activa que contribuye constantemente al crecimiento del lenguaje. Esto significa que hay una gran cantidad de recursos disponibles, como documentación, foros, tutoriales y proyectos de código abierto, que pueden ayudar a resolver problemas y aprender nuevas habilidades (Python, 2024).

Estas ventajas hacen de Python una opción ideal para una amplia variedad de aplicaciones, desde desarrollo web hasta análisis de datos y aprendizaje de máquina (Python, 2024).

Además, se utiliza para la implementación de la propuesta, la plataforma de simulación Mininet y el controlador SDN OpenDaylight, como se argumentó en el capítulo anterior.

4.2 Topología de red propuesta

La topología propuesta en la Figura 15 para el sistema de monitorización y gestión de SLA en redes 5G representa una arquitectura completa que abarca todos los componentes críticos de una red 5G avanzada. En el núcleo de esta arquitectura se sitúa el Controlador SDN, que actúa como el cerebro central de la red, proporciona una gestión unificada, programable y altamente adaptable. Este controlador se interconecta

directamente con la SBA, que incorpora funciones de red virtualizadas esenciales como UDM, PCF, AUSF, NRF, NSSF y NEF. Esta integración permite una flexibilidad en la gestión de servicios y recursos de red, y facilita la personalización dinámica de *slices* de red para cumplir con SLA específicos.

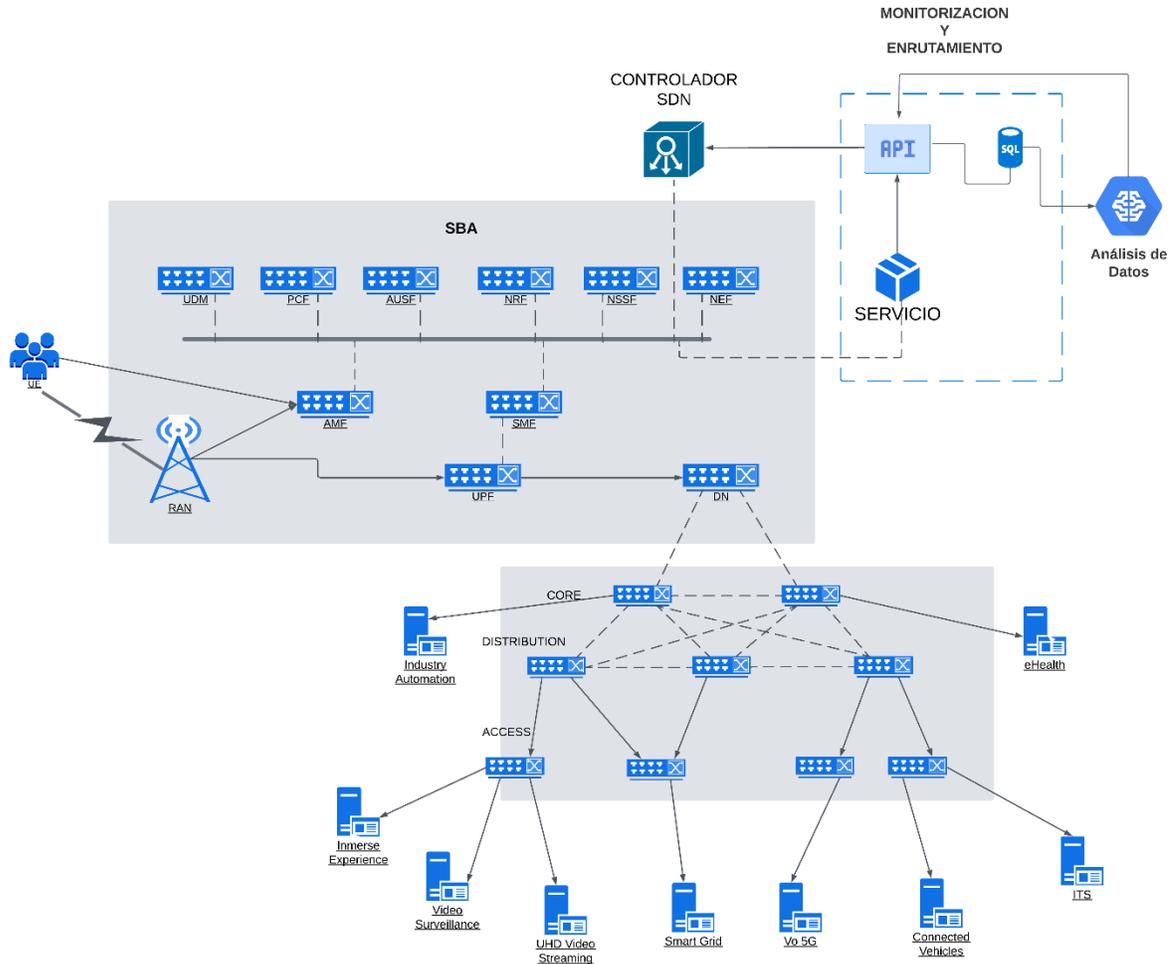


Figura 15. Topología de red propuesta.

La inclusión del acceso de radio (RAN) y su conexión fluida con funciones de *core* como AMF, SMF y UPF asegura una gestión eficiente y coordinada de la movilidad, las sesiones de usuario y el tráfico de datos, son aspectos cruciales para mantener la calidad de servicio en escenarios de alta movilidad y demanda variable. La estructura jerárquica de la red, que se extiende desde el *core* hasta las capas de distribución y acceso, permite una segmentación precisa del tráfico y una aplicación granular de políticas de QoS, esencial para gestionar eficazmente la diversidad de requisitos de los servicios 5G.

La representación de una amplia gama de servicios verticales, como *Video* vigilancia, *Streaming de Video*

UHD, Automatización de la Industria, Smart Grid, eHealth y Vo5G, demuestran la capacidad de la red para soportar y optimizar múltiples casos de uso con demandas de SLA altamente divergentes. La incorporación de una API robusta conectada al controlador SDN, junto con un servicio dedicado de monitorización y enrutamiento de SLA, facilita la recopilación en tiempo real de métricas de rendimiento, el análisis predictivo de posibles violaciones de SLA y la implementación automatizada de ajustes de red. Esta capacidad de monitorización y adaptación en tiempo real es fundamental para mantener los niveles de servicio acordados en un entorno de red dinámico y complejo. Además, la estructura *end-to-end* de la topología, que abarca desde el dispositivo del usuario hasta los servicios finales, permite una visibilidad completa y un control preciso sobre todos los factores que influyen en la experiencia del usuario y el cumplimiento de los SLA.

En conjunto, esta topología proporciona una base integral y robusta para implementar un sistema avanzado de monitorización y gestión de SLA en redes 5G, ofrece la flexibilidad, escalabilidad y control necesarios para garantizar la calidad de servicio en un ecosistema de red en constante evolución y con demandas cada vez más exigentes.

4.3 Simulaciones del sistema

El proyecto se desarrolló en tres etapas distintas, cada una enfocada en un aspecto crucial del sistema:

1. Simulación y monitorización: en esta primera etapa, se llevó a cabo la simulación de la topología de red utilizando Mininet. Esta herramienta permitió replicar fielmente la estructura de la red propuesta, incluyendo enlaces, dispositivos y conexiones. Simultáneamente, se implementó un sistema de monitorización para extraer métricas relevantes durante la simulación. Este paso fue fundamental para obtener datos realistas en un entorno controlado.
2. Desarrollo de la API y configuración de flujos: la segunda etapa se centró en la programación de una API diseñada para interactuar con el controlador de la red. Esta API permitió la configuración dinámica de los flujos de red, facilitó la comunicación entre los componentes del sistema y aseguró una clasificación adecuada de los servicios solicitados. Esta fase fue crucial para establecer la base de la gestión inteligente de la red.

3. Análisis de datos y predicción: en la etapa final, se procedió al análisis de los datos recopilados durante la fase de monitorización. Con el objetivo de predecir el comportamiento de los Acuerdos de Nivel de Servicio (SLA), se aplicaron técnicas de *Machine Learning* como redes LSTM para procesar esta información. Este análisis permitió obtener *insights* valiosos sobre el rendimiento de la red y anticipar posibles problemas o necesidades de optimización.

Esta aproximación en tres etapas permitió un desarrollo sistemático y completo del proyecto, desde la simulación inicial hasta el análisis predictivo final, pasando por la implementación de la lógica de control de la red.

4.3.1 Simulación y monitorización

La implementación de un entorno de simulación controlado y realista fue fundamental para evaluar el desempeño del sistema propuesto. Para lograr este objetivo, se utilizó la biblioteca Mininet, reconocida por su capacidad para generar topologías de red y enlaces con características específicas de QoS, elementos cruciales para los propósitos de esta investigación. El núcleo de la infraestructura de simulación se estableció en una máquina virtual (VM) alojada en la plataforma Google Cloud, donde se instalaron Mininet y el controlador ODL de SDN. Para integrar el controlador SDN ODL con la topología simulada, se empleó la función *RemoteController*, configurada con la dirección IP 127.0.0.1 y el puerto 6653, lo que permite una gestión centralizada y dinámica de los flujos de datos, replicando las capacidades de control de una red SDN real. Esta configuración en la nube no solo proporcionó flexibilidad en la gestión del entorno desde cualquier estación de trabajo, sino que también facilitó la implementación y el acceso a los recursos necesarios.

Es importante destacar que la topología implementada representa una abstracción significativa de una red 5G real, diseñada específicamente para desarrollar y evaluar un sistema de monitorización y gestión de SLA basado en SDN. Este sistema se enfoca en la gestión eficaz de la QoS y el enrutamiento dinámico para garantizar el cumplimiento de los SLA en una variedad de escenarios de red 5G. Esta simplificación modela una única estación base con 9 usuarios/servicios, en contraste con los miles que podría manejar una *gNodeB* (estación base en una red 5G real, capaz de soportar entre 10,000 y 100,000 conexiones simultáneas) (Hossain y Hasan, 2015). La topología establece una conexión directa desde el usuario (UE) a través de la RAN hasta el *Data Network* (DN), omitiendo componentes intermedios típicos como AMF, SMF

y UPF. Los switches interconectados abstraen las funciones del núcleo de red 5G, mientras que los anchos de banda reducidos (1-200 Mbps) simulan una fracción realista del total disponible en una célula 5G, que en la práctica puede ofrecer velocidades de 100-900 Mbps en áreas urbanas densas y hasta 1-3 Gbps en escenarios ideales (Gohil et al., 2013; Agiwal et al., 2016). Aunque se limita el número de usuarios a menos del 0.1% de la capacidad real de una *gNodeB*, la diversidad de servicios y los parámetros de QoS alineados con estándares ITU, ETSI y 3GPP mantienen la representatividad del modelo (Agiwal et al., 2016).

El código implementa esta topología de red abstracta, con 10 hosts (el UE y 9 servicios) que serán solicitados, identificados con la letra "h" (Tabla 1) y 10 switches (identificados con la letra "s"). Se configuran los enlaces con parámetros específicos de QoS como ancho de banda, latencia, pérdida de paquetes y *jitter*, como se muestran en las Tablas 2 y 3. Esta aproximación, que agrupa servicios similares en switches compartidos y aísla servicios con requisitos únicos, refleja una estrategia de optimización que balancea el realismo con la eficiencia de la simulación. Asimismo, se integra un sistema completo de monitorización que incluye pruebas de conectividad inicial, evaluaciones periódicas de rendimiento mediante *Ping* e *Iperf* personalizados para diferentes casos de uso y generación de tráfico de fondo. Este sistema realiza iteraciones de pruebas múltiples, registrando todos los resultados en un archivo de log, lo que permite una evaluación detallada y realista del rendimiento de la red bajo diversas condiciones y tipos de tráfico. Este enfoque crea un modelo manejable y computacionalmente factible que captura los aspectos esenciales de una red 5G, permitiendo obtener resultados significativos y extrapolables a escenarios más grandes, sin la complejidad de modelar una implementación completa con miles de conexiones simultáneas.

Tabla 1. Listado de Servicios/Aplicaciones por nodo.

Servicio	ID del Host
Usuario	h1
Experiencia Inmersiva	h2
Vigilancia de video	h3
eHealth	h4
Smart Grid	h5
Automatización Industrial	h6
Vehículos Conectados	h7
ITS Sistema de Transporte Inteligente	h8
Vo5G	h9
<i>Streaming</i> de Video HD	h10

Los dispositivos conectados a cada aplicación pueden variar en dependencia del servicio, por ejemplo, en Vigilancia de video, un ejemplo clave sería una cámara IP de alta resolución. Para eHealth, se podría considerar un reloj inteligente (smartwatch) con monitor de ritmo cardíaco continuo. En el ámbito de Smart Grid, un medidor inteligente instalado en hogares sería un dispositivo fundamental. Para Automatización Industrial, un robot industrial con sensores integrados representa un ejemplo claro. En el caso de Vehículos Conectados, la unidad de control electrónico (ECU) del vehículo es esencial. Para Sistemas de Transporte Inteligente, un semáforo inteligente con capacidad de ajuste en tiempo real es un ejemplo representativo. En Vo5G, un *smartphone* compatible con esta tecnología es el dispositivo por excelencia. Para *Streaming* de Video UHD, una *smart* TV con capacidad de reproducción en alta definición es un ejemplo común. En aplicaciones de Experiencia Inmersiva, unas gafas de realidad virtual serían un dispositivo típico. Finalmente, para aplicaciones de hogar inteligente, un termostato conectado a internet sería un ejemplo característico.

Tabla 2. Características de QoS de cada enlace de los switches al servicio o UE de la topología.

Enlace de conexión (fuente, destino)	BW(Mbps)	Latencia (ms)	Pérdida de paquetes (%)	Jitter (ms)
h1, s0	200	0.3	0.001	0.05
h2, s6	50	2	0.005	0.05
h3, s6	10	1	0.0005	0.5
h4, s1	16	0.1	0.000001	0.2
h5, s8	1	1	0.00001	0.1
h6, s2	1	0.1	0.000001	0.02
h7, s9	10	0.5	0.0001	0.01
h8, s9	0.5	0.5	0.001	0.5
h9, s10	10	1	0.002	0.5
h10, s6	10	0.5	0.0005	0.5

Tabla 3. Características de QoS de cada enlace de los switches de la topología.

Enlace de conexión (fuente, destino)	BW(Mbps)	Latencia (ms)	Pérdida de paquetes (%)	Jitter (ms)
s0, s1	100	0.1	0.00001	0.01
s0, s2	100	0.2	0.00001	0.01
s1, s3	100	0.1	0.00001	0.01
s1, s4	100	0.2	0.00001	0.01
s1, s5	100	0.3	0.00001	0.01
s2, s3	100	0.1	0.00001	0.01
s2, s4	100	0.1	0.00001	0.01
s2, s5	100	0.1	0.00001	0.01
s3, s4	100	0.2	0.00005	0.02
s4, s5	100	0.2	0.00005	0.02

s3, s6	100	0.3	0.00001	0.02
s4, s8	100	0.1	0.00001	0.005
s5, s9	100	0.15	0.0001	0.01
s5, s10	100	0.15	0.0001	0.02

Es crucial destacar que todos los valores asignados se basan en los estándares de telecomunicaciones establecidos por la ITU, ETSI y 3GPP, lo que garantiza la validez y relevancia de la simulación en el contexto de las redes de telecomunicaciones actuales.

Para la configuración de la simulación, los archivos de la topología se cargaron en la VM utilizando el protocolo SSH (por las siglas del término en inglés, *Secure Shell*) desde la terminal. Esta topología, se ilustra en la Figura 16, y representa fielmente la estructura de red necesaria para los experimentos planteados. La adhesión a los estándares mencionados no solo asegura la representatividad de los parámetros de QoS utilizados, sino que también facilita la extrapolación de los resultados a escenarios de despliegue reales, lo que permite una evaluación más precisa del sistema propuesto y su capacidad para cumplir con los acuerdos de nivel de servicio (SLA) en condiciones operativas realistas.

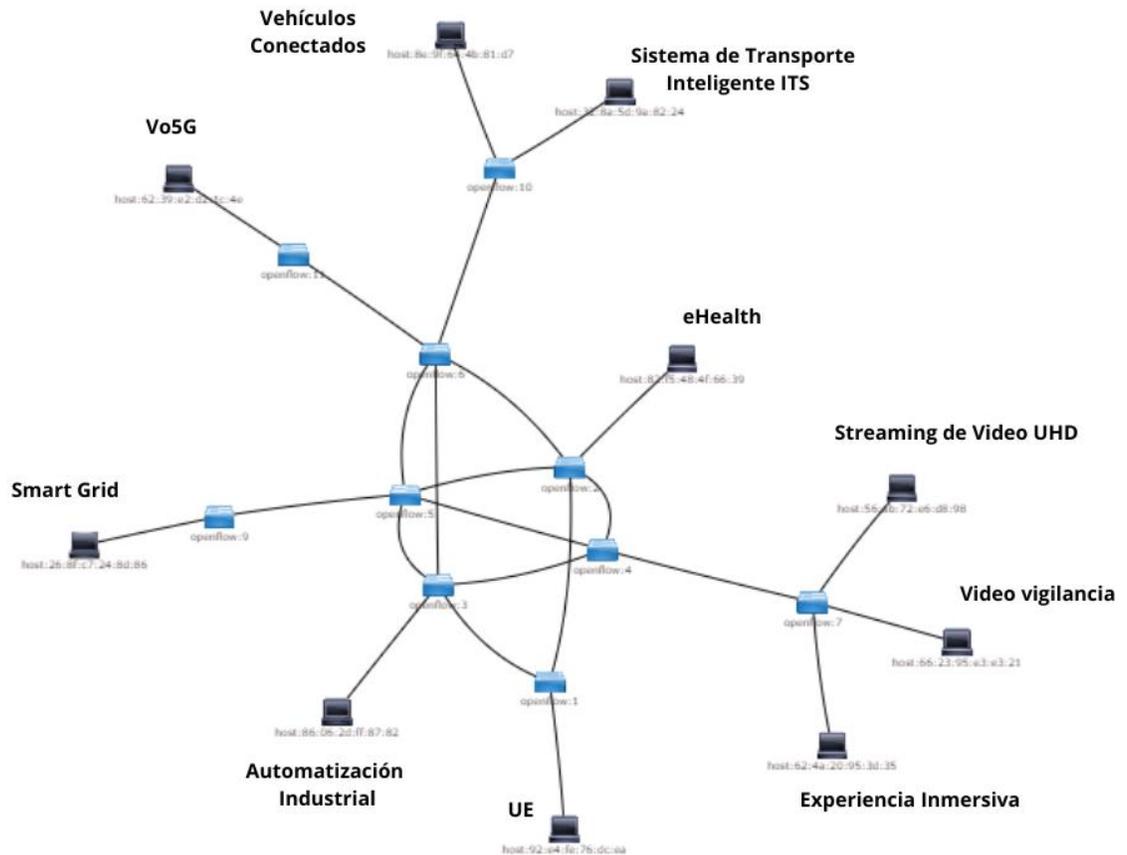


Figura 16. Topología de la red en Mininet.

El diseño y la distribución de la topología y los parámetros de los SLA que se muestran en la Tabla 4 se tomó del trabajo de Tesis de (Gonzalez-Franco et al., 2023). La Figura 16 ilustra una estrategia de agrupación de servicios en la topología de red, donde servicios similares, como Redes Inteligentes y Automatización Industrial, o ITS y Vehículos Conectados, comparten switches de destino por ser similares en parámetros de rendimiento. En contraste, servicios con requisitos únicos como Vo5G y eHealth se mantienen aislados. Esta aproximación de agrupar servicios similares y aislar los distintivos, busca optimizar la precisión de las simulaciones, reduciendo la complejidad de la topología y simplificando el proceso de simulación, sin comprometer la evaluación del rendimiento del sistema. Esta estrategia evita la necesidad de diseñar topologías excesivamente complejas para cada servicio individual, manteniendo el enfoque en los objetivos principales de la investigación.

En conjunto, estas decisiones de diseño y configuración proporcionan un entorno de simulación que combina realismo, eficiencia y alineación con los estándares de la industria. La integración de la plataforma *cloud*, el software de simulación, la topología personalizada y los parámetros basados en estándares crea

una base sólida y flexible para la investigación y experimentación en el campo de las redes definidas por software (SDN) y la calidad de servicio (QoS) en entornos de telecomunicaciones avanzados.

Tabla 4. Parámetros de SLA por aplicación. Tomada de (González-Franco et al., 2023).

ID del host	PacketLoss	Latencia (ms)	Jitter (ms)	BitRate (Mbits/s)
h2	5	7	20	50
h3	0.001	10	5	10
h4	0.00000001	1	10	16
h5	0.0001	5	1	1
h6	0.0000001	1	0.1	1
h7	0.001	3	0.44	10
h8	0.1	10	20	0.5
h9	1	20	30	1
h10	1	4	5.84	10

Este enfoque integral no solo permite una evaluación rigurosa del sistema propuesto, sino que también genera resultados significativos y aplicables al mundo real de las redes de telecomunicaciones avanzadas, proporcionando una valiosa contribución a la metodología de evaluación de sistemas de red de próxima generación. La implementación de un entorno de simulación controlado y realista fue fundamental para evaluar el desempeño del sistema propuesto en el contexto de las redes de telecomunicaciones avanzadas.

4.3.1.1 Parámetros de generación de tráfico

En la Tabla 5 se describen los parámetros utilizados para generar el tráfico para cada servicio el cual es diferente para cada caso, la prueba de *Iperf* se realizó durante 10 segundos en todas las simulaciones.

Tabla 5. Parámetros de la generación de tráfico para cada aplicación.

Aplicación	Ancho de Banda (Mbits/s)	Distribución del Ancho de	Tamaño de Paquete (bytes)	Intervalo Ping	Repeticiones Ping	Protocolo iPerf
------------	--------------------------	---------------------------	---------------------------	----------------	-------------------	-----------------

		Banda (Mbits/s)		(s)		
Experiencia Inmersiva	50 - 55	Normal ($\mu=55, \sigma=5$)	1200	0.01	5	UDP
Vigilancia de video	10-12	Normal ($\mu=12, \sigma=2$)	800	0.01	4	UDP
eHealth	16 - 20	Poisson ($\lambda=20, k=5$)	500	0.01	6	TCP
Smart Grid	1 - 1.5	Normal ($\mu=1.5, \sigma=0.5$)	64	0.01	3	TCP
Automatización Industrial	1-3	Normal ($\mu=3, \sigma=2$)	500	0.1	5	TCP
Vehículos Conectados	10-15	Normal ($\mu=15, \sigma=5$)	300	0.05	5	TCP
ITS Sistema de Transporte Inteligente	0.5 - 1	Normal ($\mu=1, \sigma=0.5$)	64	0.2	4	TCP
Vo5G	0.8-1	Poisson ($\lambda=1, k=2$)	160	0.01	8	UDP
<i>Streaming</i> de Video HD	10-20	Normal ($\mu=20, \sigma=5$)	1200	0.5	5	UDP

Las distribuciones de Poisson y Normal son fundamentales para modelar el tráfico en diversas aplicaciones de red, cada una con sus propias características y usos específicos. La distribución de Poisson se emplea para eventos discretos en intervalos fijos, siendo ideal para modelar la llegada de paquetes, eventos raros y tráfico de voz y datos discretos. Se define por un único parámetro λ que representa la tasa media de ocurrencia, y es adecuada para eventos independientes con tasa constante (J. Cao et al., 2003). Por otro lado, la distribución Normal o Gaussiana se utiliza para fenómenos continuos, modelando variaciones en el ancho de banda, tráfico agregado y aplicaciones con variabilidad simétrica. Se caracteriza por dos parámetros: μ (media) y σ (desviación estándar), y es simétrica alrededor de la media (Karagiannis et al., 2004).

En el contexto de redes, Poisson se aplica a servicios como Vo5G y eHealth, mientras que la Normal es útil para experiencias inmersivas, *Streaming* de Video UHD, Smart Grid y automatización industrial. La elección entre ambas depende de la naturaleza del tráfico: Poisson para eventos discretos y llegadas aleatorias, y Normal para variaciones continuas y tráfico agregado. Estas distribuciones son cruciales para simular patrones de tráfico realistas, facilitando el diseño y prueba de redes capaces de manejar una amplia gama de aplicaciones y servicios (J. Cao et al., 2003; Karagiannis et al., 2004).

La elección entre UDP y TCP en el diseño de aplicaciones de red es crucial y depende de las características específicas de cada aplicación. UDP se prefiere cuando la velocidad y el tiempo real son prioritarios, como en aplicaciones de *streaming* de video, juegos en línea o VoIP, donde se puede tolerar cierta pérdida de paquetes y se requiere un *overhead* mínimo. El *overhead* se refiere a la cantidad adicional de información o procesamiento necesario para implementar un protocolo o función, más allá de los datos útiles. UDP tiene un *overhead* menor comparado con TCP, lo que lo hace más eficiente para aplicaciones que requieren transmisión rápida y pueden manejar alguna pérdida de datos. En el contexto de aplicaciones específicas, UDP se utiliza para experiencias inmersivas, *streaming* de video HD y Vo5G (Fahad Taha Al-Dhief, 2018).

Por otro lado, TCP es la opción ideal cuando la confiabilidad y la integridad de los datos son esenciales, como en transferencias de archivos, transacciones financieras o aplicaciones de base de datos. TCP garantiza la entrega ordenada de paquetes y maneja automáticamente la congestión de la red, es preferido para eHealth, Smart Grid, automatización industrial y sistemas de transporte inteligente (Afanasyev et al., 2010).

La decisión entre UDP y TCP implica un balance entre velocidad y confiabilidad, donde UDP ofrece rapidez a costa de posibles pérdidas, y TCP asegura la entrega completa y ordenada a expensas de mayor latencia y *overhead*. Ultimadamente, la elección depende de las necesidades específicas de cada aplicación en términos de tiempo real, integridad de datos y tolerancia a pérdidas (Fahad Taha Al-Dhief, 2018).

El tamaño de paquete, intervalo de *Ping* y repeticiones de este, son parámetros importantes en la configuración y evaluación de redes. Estos valores se eligen según las características específicas de cada aplicación y los requisitos de la red. El tamaño de paquete varía según el tipo de aplicación y los requisitos de la red. Paquetes más grandes (como 1200 bytes) son eficientes para transferencias de datos voluminosos, mientras que paquetes más pequeños (64 bytes) son típicos para aplicaciones que requieren baja latencia y actualizaciones frecuentes (Botta et al., 2012).

El intervalo de *Ping* determina la frecuencia con la que se envían paquetes de prueba. Intervalos cortos (0.01 s) son útiles para aplicaciones en tiempo real que requieren monitoreo constante, mientras que intervalos más largos (0.5 s) son adecuados para aplicaciones menos sensibles al tiempo. El número de repeticiones de *Ping* afecta la precisión de las mediciones de latencia y pérdida de paquetes. Más repeticiones (como 8) proporcionan datos más confiables, mientras que menos repeticiones (3-5) pueden ser suficientes para verificaciones rápidas (Botta et al., 2012).

4.3.2 Desarrollo de la API y configuración de flujos

Esta API, implementada en Flask y Postman, está diseñada para gestionar y optimizar rutas de red en un entorno de red definida por software (SDN). La aplicación se conecta a una base de datos *SQLite* para almacenar y gestionar registros de rendimiento de red, incluyendo métricas como pérdida de paquetes, latencia, *jitter* y ancho de banda. El código está disponible bajo petición a quien lo requiera.

Está compuesta por tres *endpoints* principales que ofrecen una gestión completa de las rutas de red. El *endpoint* "*configure-route*" (método POST) permite la configuración manual de rutas para nodos específicos, aceptando datos JSON con información del nodo destino y tipo, y utilizando parámetros predefinidos para la configuración. El *endpoint* "*update-routes*" (método GET) proporciona una funcionalidad automatizada para actualizar múltiples rutas que no cumplen con el SLA, buscando registros con incumplimientos y configurando nuevas rutas según sea necesario. Por último, "*records*" (método GET) recupera todos los registros almacenados en la base de datos, ofreciendo una visión integral del estado actual e histórico de la red. En conjunto, estos *endpoints* permiten una gestión flexible y eficiente de la red, combinando opciones de configuración manual y automatizada con capacidades de monitoreo completo.

En la Figura 17 se ilustra el entorno visual en el Postman donde se muestra la respuesta a una solicitud de tipo GET a la dirección <http://127.0.0.1:5000/update-routes> donde se actualizan las rutas que no cumplen con el SLA para el caso de los servicios h4, h8 y h9.

Además de estos *endpoints*, la API integra funcionalidades avanzadas de gestión de red. Utiliza grafos para representar y manipular la topología de la red SDN. Calcula todas las rutas posibles entre nodos fuente y destino, evaluando métricas de red como ancho de banda, retardo, pérdida de paquetes y *jitter* para cada ruta y seleccionando la de menor latencia. Interactúa directamente con el controlador SDN OpenDayLight para configurar, eliminar y consultar reglas de flujo en los switches *OpenFlow*. Permite la implementación dinámica de rutas optimizadas en la infraestructura de red física.

Cuando se configuran las rutas en el controlador SDN, la API asigna una prioridad específica a cada flujo de tráfico en dependencia de la aplicación como se muestra en la Tabla 5. Esta prioridad se asigna directamente a la regla de flujo en el controlador, determinando el orden de ejecución de los flujos, donde los de mayor prioridad se procesan primero. El rango de prioridades utilizado se basa en las especificaciones de *OpenFlow*, que permiten valores de 0 a 65535, aunque en este caso se emplea un subconjunto de ese

rango para permitir futuras expansiones. Este enfoque tiene como objetivo priorizar las aplicaciones con requisitos más críticos de calidad de servicio (QoS). La escala de prioridades va desde 280 (más alta) para aplicaciones de eHealth, consideradas las más críticas, hasta 200 (más baja) para *Streaming* de Video UHD. Esta estrategia de priorización asegura que aplicaciones cruciales como eHealth, automatización industrial y vehículos conectados reciban los recursos de red necesarios, incluso en condiciones de congestión. Al mismo tiempo, se mantiene un nivel de servicio adecuado para aplicaciones menos críticas pero importantes para la experiencia del usuario. Este sistema de priorización basado en aplicaciones permite una gestión más eficiente de los recursos de red, adaptándose a los diversos requisitos de QoS de las diferentes aplicaciones en entornos 5G y IoT avanzados (3rd Generation Partnership Project, 2017).

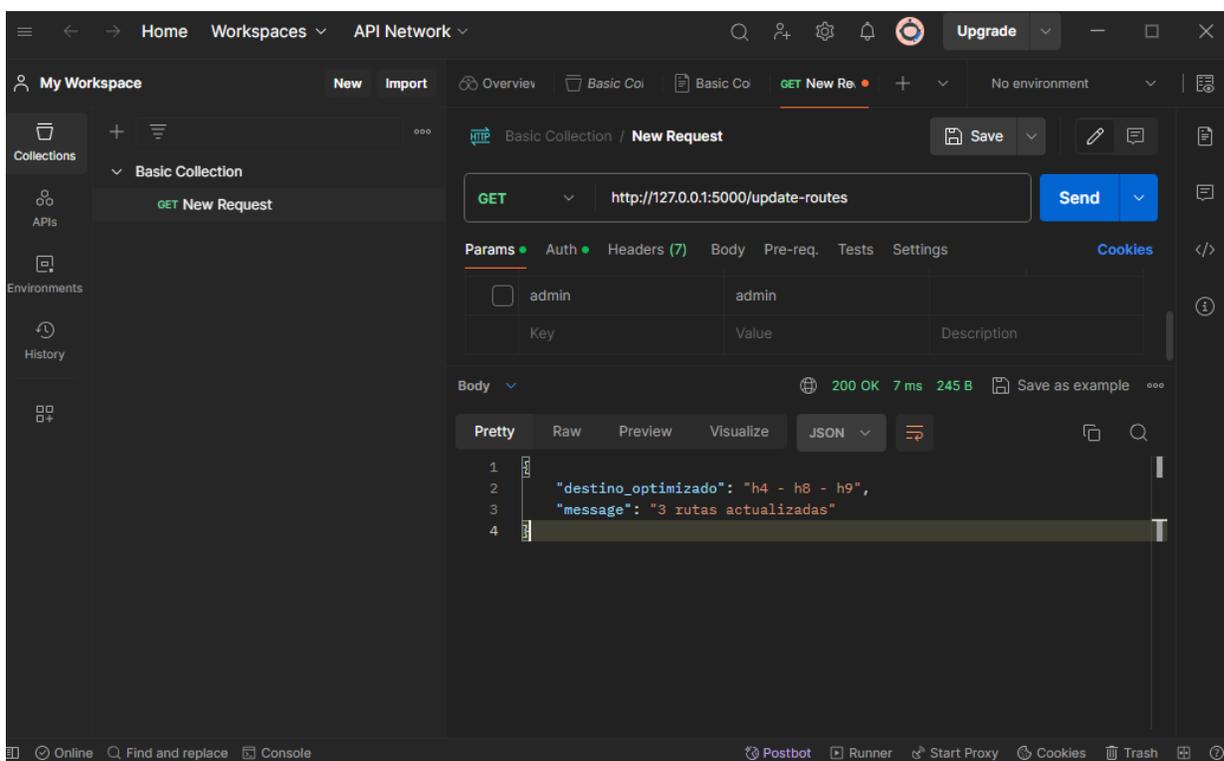


Figura 17. Imagen de Postman.

Tabla 5. Prioridad de los flujos por aplicación.

Aplicación	Nodo	Prioridad
eHealth	h4	280
Automatización Industrial	h6	270
Vehículos Conectados	h7	260

Smart Grid	h5	250
Vo5G	h9	240
Vigilancia de Video	h3	230
Sistema de Transporte Inteligente ITS	h8	220
Experiencia Inmersiva	h2	210
<i>Streaming</i> de Video HD	h10	200

1. eHealth (h4, prioridad 280): la más alta prioridad se asigna a aplicaciones de salud electrónica, reconociendo la criticidad de los servicios médicos y la necesidad de transmisión de datos en tiempo real para situaciones potencialmente vitales (3rd Generation Partnership Project, 2017).
2. Automatización de la Industria (h6, prioridad 270): la automatización industrial recibe la segunda prioridad más alta, reflejando la importancia de la precisión y la baja latencia en entornos industriales para mantener la eficiencia y la seguridad de los procesos automatizados (Li et al., 2017).
3. Vehículos Conectados (h7, prioridad 260): los vehículos conectados requieren comunicaciones rápidas y confiables para garantizar la seguridad vial y la eficiencia del tráfico (Storck y Duarte-Figueiredo, 2019).
4. Smart Grid (h5, prioridad 250): las redes eléctricas inteligentes necesitan comunicación en tiempo real para la gestión eficiente de la energía y la respuesta rápida a cambios en la demanda o emergencias (Wang et al., 2018).
5. Voz sobre 5G (h9, prioridad 240): la voz sobre 5G recibe una prioridad alta para mantener la calidad de las llamadas de voz en la nueva generación de redes móviles (3GPP Release 15, 2017).
6. Videovigilancia (h3, prioridad 230): la vigilancia por video requiere un ancho de banda constante y una latencia baja para la transmisión efectiva de imágenes en tiempo real (Nightingale et al., 2018).
7. ITS (h8, prioridad 220): los sistemas de transporte inteligente necesitan comunicación confiable para la gestión eficiente del tráfico y la seguridad (Storck y Duarte-Figueiredo, 2019).

8. Experiencia Inmersiva (h2, prioridad 210): las experiencias inmersivas, como la realidad virtual o aumentada, requieren un alto ancho de banda y baja latencia para una experiencia de usuario fluida (H. Cao et al., 2022).
9. *Streaming* de Video UHD (h10, prioridad 200): aunque importante para la experiencia del usuario, el *Streaming* de Video UHD recibe una prioridad relativamente menor en comparación con servicios más críticos (Nightingale et al., 2018).

La API implementa esta jerarquía de prioridades directamente en la configuración de rutas, utilizando estos valores para asignar recursos de red adecuados mediante reglas de flujo en los *switches* SDN. Este sistema garantiza que las aplicaciones críticas reciban los recursos necesarios, incluso en situaciones de congestión, mientras mantiene un funcionamiento eficiente de aplicaciones menos críticas. Esta característica es esencial para la gestión efectiva de redes 5G y servicios IoT avanzados, permitiendo una asignación inteligente y dinámica de recursos que se adapta a la diversidad de aplicaciones y sus requisitos específicos.

4.3.3 Análisis de datos y predicción de SLA

En este módulo, se implementó un modelo predictivo basado en redes neuronales recurrentes de tipo *Long Short-Term Memory* (LSTM) para analizar y predecir el comportamiento del Acuerdo de Nivel de Servicio (SLA) en diferentes nodos de la red propuesta. El modelo se aplicó a distintos tipos de simulaciones (rf-ct, rf-st, usd-ct, usd-st) obtenidas del sistema de monitorización y a los nodos (h2, h3, h5, h7, h8, h9, h10), excluyendo el nodo h4 debido a que por sus condiciones específicas de latencia 1 ms no cumple el SLA en ninguna medición.

El algoritmo se diseñó con varios parámetros ajustables: tf (tiempo futuro de predicción, variando de 1 a 4), tp (cantidad de datos históricos utilizados, abarcando valores de 1, 4, 7, 10, 13, 16 y 19), nodo específico y tipo de simulación. Para cada combinación de estos parámetros, se prepararon los datos correspondientes, creando secuencias de entrada X con las cuatro métricas de red (pérdida de paquetes, latencia, *jitter* y ancho de banda) y sus valores objetivo Y (SLA futuro).

El proceso de entrenamiento y evaluación se realizó mediante un enfoque de validación cruzada, utilizando el 80% de los datos para entrenamiento y el 20% restante para pruebas. Este proceso se repitió 20 veces

para cada configuración, permitiendo obtener estimaciones robustas de la exactitud o precisión (*accuracy*) y la pérdida (*loss*). Al utilizar diferentes conjuntos de datos aleatorios en cada iteración, se reduce el sesgo de selección, se mitiga el riesgo de sobreajuste y se proporciona una evaluación más estable del rendimiento del modelo. Además, este método aumenta la confianza estadística en los resultados y permite analizar la variabilidad del rendimiento del modelo a través de iteraciones múltiples.

Accuracy mide la proporción de predicciones correctas del modelo, ofrece una visión general de su rendimiento fácil de interpretar, aunque puede ser engañosa en conjuntos de datos desequilibrados (Sokolova y Lapalme, 2009). Por otro lado, la pérdida (*loss*) cuantifica la diferencia entre las predicciones del modelo y los valores reales, se utiliza comúnmente la entropía cruzada binaria en problemas de clasificación binaria. Un valor más bajo de *loss* indica predicciones más cercanas a la realidad y guía el proceso de aprendizaje durante el entrenamiento (Janocha y Czarnecki, 2017). Estas métricas son complementarias: mientras la *accuracy* proporciona una medida intuitiva, la *loss* ofrece una perspectiva más matizada del rendimiento del modelo. Aunque una pérdida baja suele correlacionarse con una exactitud alta, monitorear ambas métricas proporciona una evaluación más completa del modelo LSTM en la predicción del comportamiento de la red en escenarios y configuraciones de SLA diversos (Janocha y Czarnecki, 2017).

Se calcularon la media y la desviación estándar de estos indicadores para evaluar el rendimiento y la estabilidad del modelo bajo diferentes condiciones. La implementación se realizó utilizando *TensorFlow* y *Keras* para la construcción y entrenamiento del modelo LSTM, junto con *scikit-learn* para la preparación y división de los datos. El modelo LSTM se configuró con una capa de 50 unidades, seguida de una capa densa con activación sigmoide para la clasificación binaria del SLA.

Los resultados obtenidos proporcionan una visión comprensiva del rendimiento del modelo para cada combinación de parámetros, lo que permite identificar las configuraciones óptimas para cada tipo de simulación y nodo. Esta información es crucial para entender qué cantidad de datos históricos y qué horizonte de predicción son más efectivos en diferentes escenarios de red, y así informar decisiones de gestión y optimización de servicios de red en tiempo real.

Este enfoque no solo demuestra la aplicabilidad de las técnicas de aprendizaje profundo en la predicción de SLA, sino que también proporciona una metodología sistemática para evaluar y comparar diferentes configuraciones de modelos en el contexto de redes de comunicaciones. Los resultados de este estudio

pueden servir como base para el desarrollo de sistemas predictivos más avanzados y la implementación de estrategias proactivas de gestión de la calidad de servicio en redes complejas.

4.4 Resultados obtenidos

En los cuatro escenarios de simulación, se recopilieron 1350 muestras, lo que resulta en un total de 5400 mediciones en nuestra bitácora (1350 muestras por 2 tipos de simulación por 2 fases de monitorización). Esta recopilación de datos ilustrada en la Tabla 6, proporciona una base estadística robusta para el análisis, lo cual permite una comparación directa entre las trayectorias por defecto y las optimizadas, así como entre los métodos de monitorización individual y grupal.

Tabla 6. Resultados de la monitorización.

Tipo de monitorización	% de muestras con Pérdidas de paquetes	Muestras que cumplen SLA	% de muestras que cumplen SLA	Latencia promedio (ms)
usd-st	11.85	832	61.63	7.98
usd-ct	7.53	998	73.93	5.98
rf-st	4.74	1227	90.89	1.84
rf-ct	0	1349	99.93	1.86

4.4.1 Comportamiento del SLA

En el caso de las mediciones para evaluar el cumplimiento del SLA, se puede concluir que, como se ilustra en la Figura 18, el porcentaje menor de cumplimiento se obtuvo cuando se realizó la simulación con envío simultáneo de ráfagas de tráfico desde el usuario a todos los nodos destino (rf-st). En contraste, se alcanzó un mayor cumplimiento del SLA cuando se realizó la simulación secuencial, cuando se envió tráfico individual a cada nodo y al configurar todas las rutas con la trayectoria que cumple el SLA, y que presentan menor latencia punto a punto. En la monitorización por ráfagas hubo una mejora en el cumplimiento de SLA de 12.3% y en la monitorización individual de 9.04%. El incremento del porcentaje del SLA en la gráfica era un valor esperado, ya que en cada paso se mejoran las condiciones de la red.

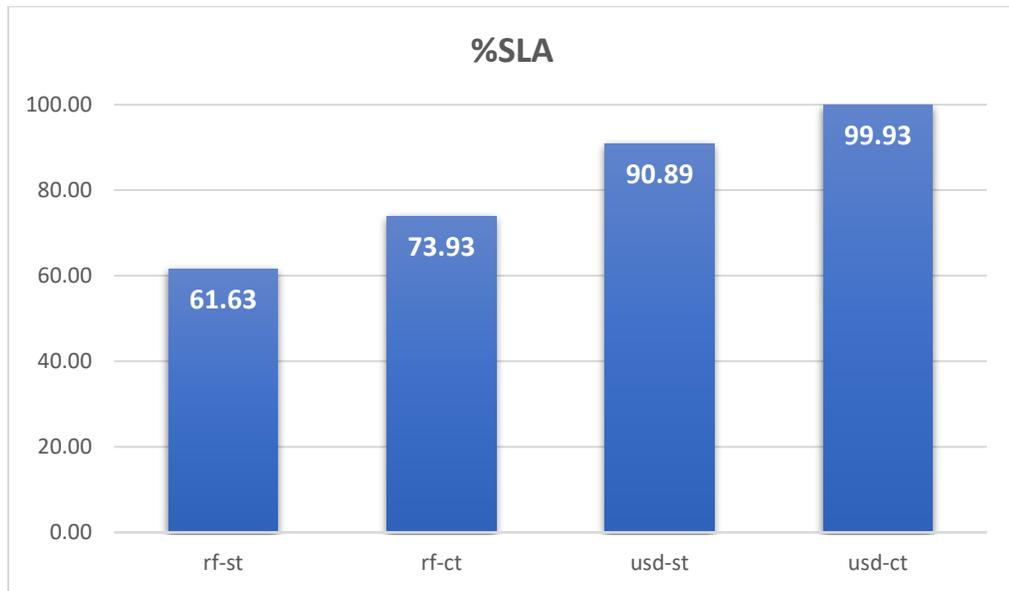


Figura 18. Porcentaje de cumplimiento de SLA.

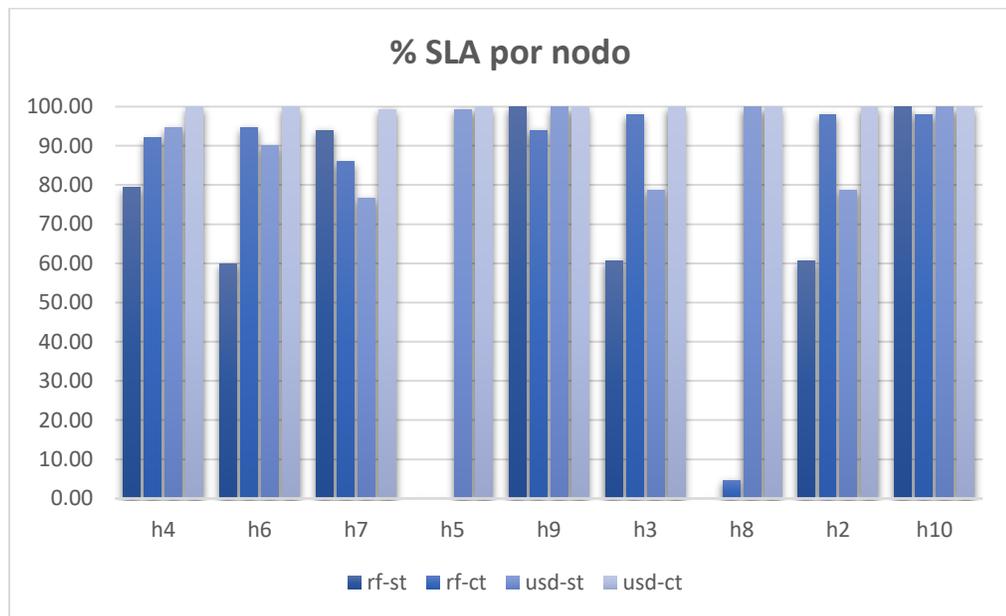


Figura 19. Porcentaje de cumplimiento de SLA por aplicación.

En cuanto al comportamiento de este parámetro por nodo o aplicación como se ilustra en la Figura 19 las aplicaciones con un mejor desempeño en todas las monitorizaciones fueron h10 (*Streaming* de Video UHD) con 99.5% y h9 (Vo5G) con 98.5% de cumplimiento de SLA. De forma opuesta las de más bajo desempeño fueron h8 (ITS) con 51.17% y h5 (Smart Grid) con 49.83% de cumplimiento.

4.4.2 Comportamiento de las pérdidas de paquetes

De los resultados obtenidos, además, se puede concluir que la pérdida de paquetes, como se ilustra en la Figura 20, fue mayor (11.85%) cuando se enviaron simultáneamente las ráfagas de tráfico a todos los destinos al utilizar las rutas por defecto establecidas por el controlador ODL. Esto se traduce en una congestión mayor. En cambio, se observó una mejoría a 0% de pérdidas de paquetes en el caso donde se envió tráfico individual a una aplicación y se configuraron las trayectorias que cumplen el SLA y tienen menor latencia. Realizando el análisis por tipo de monitorización, disminuyó cuando se seleccionaron las trayectorias en ráfagas e individual en 4.32% y 4.74% respectivamente.

En cuanto al comportamiento de las pérdidas de paquetes por nodos como se ilustra en la Figura 21, de manera general, el nodo h5 (Smart Grid) presenta valores más altos de enlaces con pérdidas de paquetes seguido de los nodos h2 (Experiencia Inmersiva) y h3 (Vigilancia de Video). En contraste, los nodos h4 (eHealth) y h10 (*Streaming* de Video UHD) son los que menos por ciento de pérdidas de paquetes presentan.

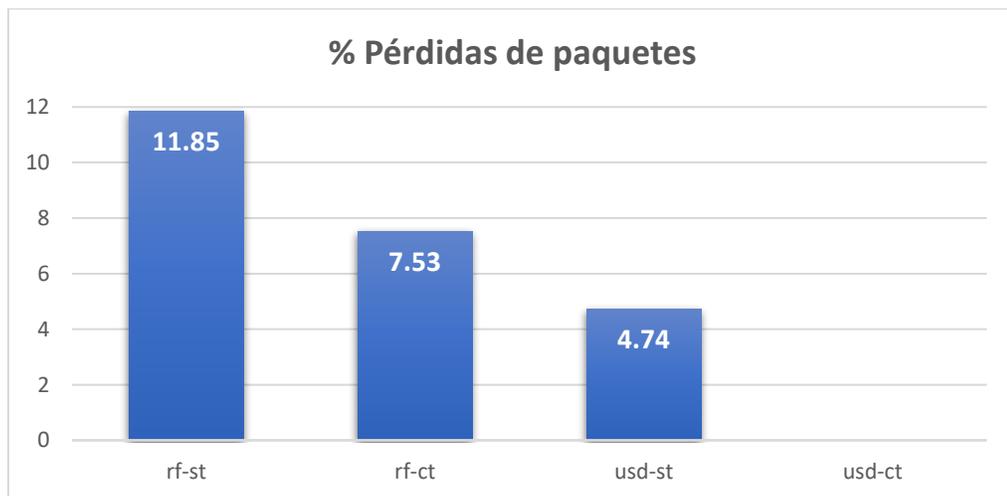


Figura 20. Porcentaje de pérdida de paquetes.

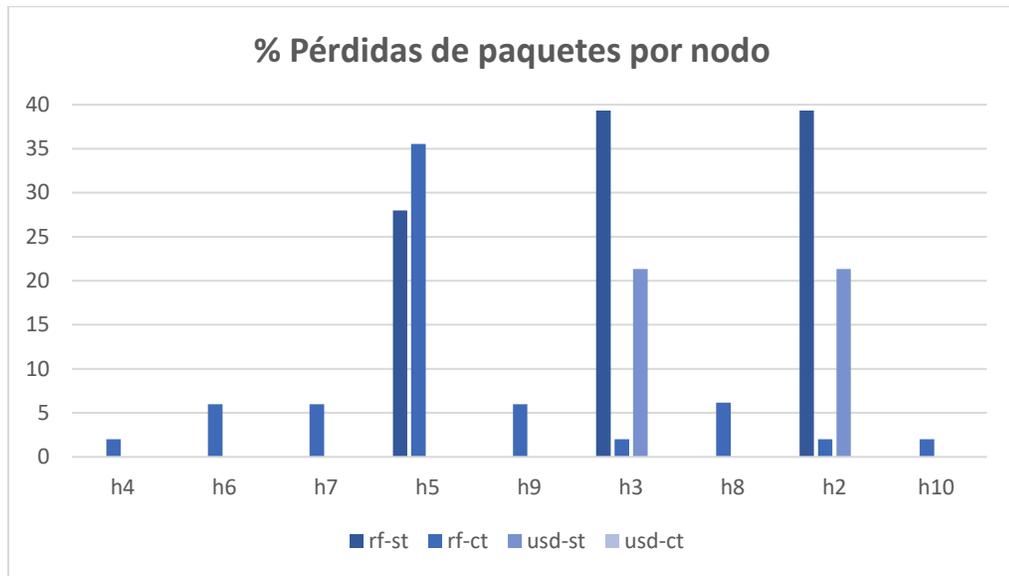


Figura 21. Porcentaje de pérdidas de paquetes por aplicación.

4.4.3 Comportamiento de la latencia

Al redirigir el tráfico a la ruta que cumple el SLA y que tiene menor latencia, se observa cómo disminuye la latencia promedio de la simulación al realizar esta selección en las mediciones con (ct) (ver Figura 22). La mayor latencia promedio de 7.98 ms, se registró en el caso de ráfagas simultáneas (rf-st) y se observa una disminución a 5.98 ms cuando se seleccionan las rutas.

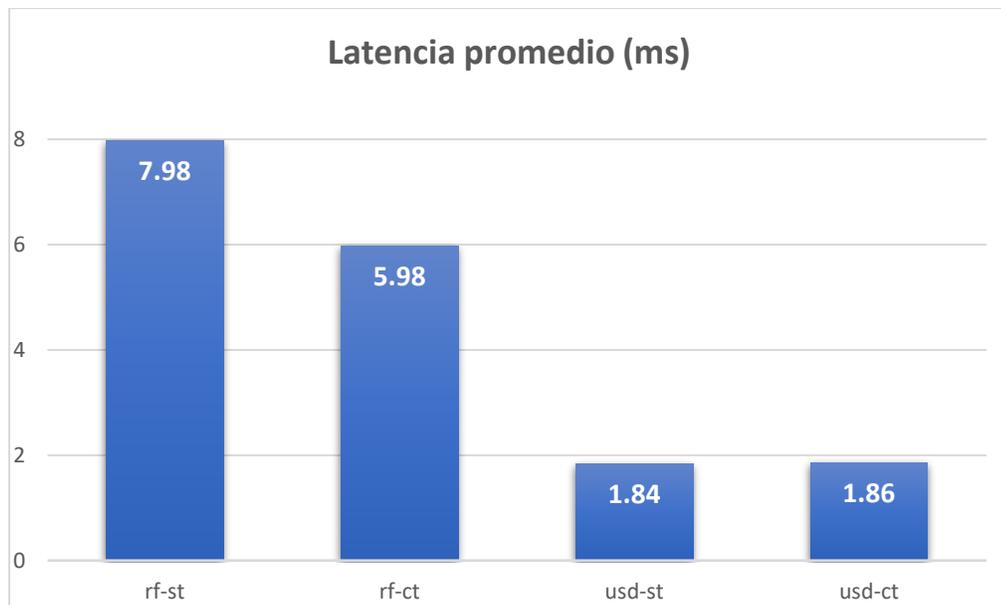


Figura 22. Latencia promedio por tipo de medición.

Además, se mantuvo casi constante en los casos de tráfico secuencial (usd-st y usd-ct), algo comprensible ya que en estos casos solo se envía tráfico a un solo nodo en la red. En general, el análisis de latencia revela una mejora significativa al pasar de escenarios de ráfaga de tráfico (rf) a tráfico secuencial (usd), y al implementar configuraciones de trayectorias optimizadas (ct).

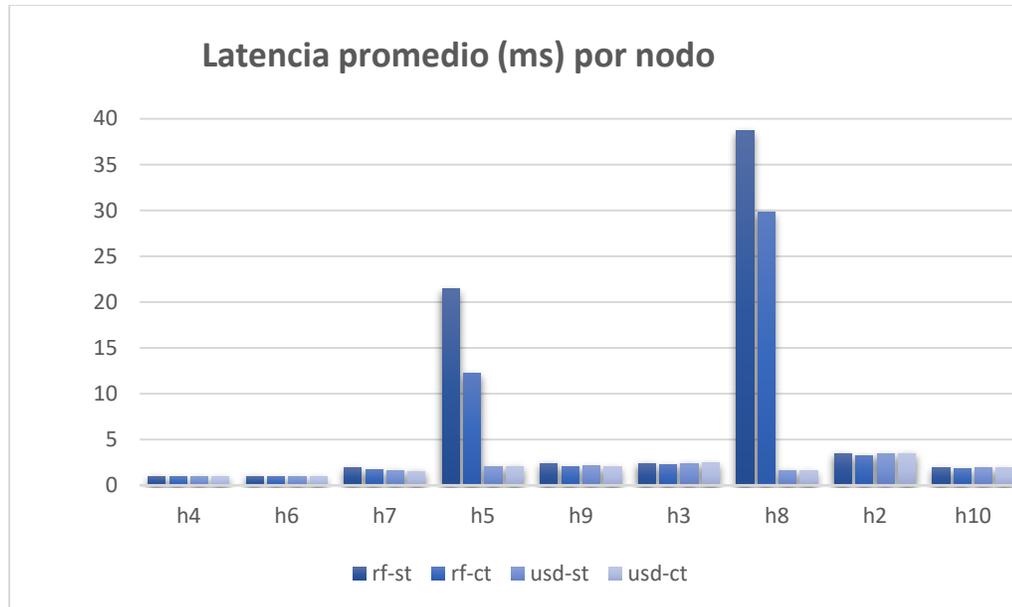


Figura 23. Latencia promedio por aplicación.

Como se ilustra en la Figura 23, los nodos que presentaron mayores resultados de latencia promedio fueron h8 (ITS) y h5 (Smart Grid) en las simulaciones en ráfagas, los cuales muestran una disminución de la latencia al seleccionar las rutas. Cabe destacar el comportamiento de la latencia en cada aplicación, las cuales están ordenadas según la asignación de prioridad en el controlador, en dependencia de los requisitos de cada servicio. Los resultados obtenidos van en correspondencia con la asignación de prioridades establecida para cada aplicación.

4.4.4 Resultados de la predicción de SLA

El modelo LSTM para la predicción del comportamiento del SLA en diferentes nodos de la red muestra un rendimiento variado, generalmente efectivo. Se observan patrones interesantes tanto en la precisión del modelo como en los parámetros utilizados para las predicciones.

En cuanto al rendimiento, como se observa en la Tabla 7, el modelo logra una precisión perfecta o casi

perfecta en varios escenarios, especialmente en los nodos h5, h8, h9 y h10, independientemente del tipo de simulación. Se destaca que la precisión del modelo puede ser 1 cuando todos los valores de SLA son 0 o cuando todos los valores son 1, como pasa en varios casos de estos nodos. No obstante, se identificaron algunos puntos que requieren atención. El nodo h2 presentó el menor valor de precisión, con 0.86 en la simulación *usd-st*. Esto sugiere que este nodo en particular podría requerir un ajuste más fino del modelo o un análisis más detallado de sus características específicas

Tabla 7. Resultado de la predicción.

Nodos/ Med Accuracy/ Med Loss	rf-st	rf-ct	usd-st	usd-ct
h2	(16,1)	(10,1)	(19,1)	(10,1)
<i>Med Accuracy</i>	0.95	0.99	0,86	1
<i>Med Loss</i>	0.14	0.04	0,33	0.0001
h3	(13,1)	(16,1)	(16,1)	(10,1)
<i>Med Accuracy</i>	0.94	0.99	0,87	1
<i>Med Loss</i>	0.20	0.06	0,34	0.0001
h5	(10,1)	(10,1)	(10,1)	(13,1)
<i>Med Accuracy</i>	1	1	1	1
<i>Med Loss</i>	0.00020	0.0002	0,0002	0.0002
h6	(16,1)	(16,1)	(4,1)	(19,1)
<i>Med Accuracy</i>	1	1	0,88	1
<i>Med Loss</i>	0,0003	0,0003	0,34	0,0003
h7	(10,1)	(4,1)	(10,1)	(7,1)
<i>Med Accuracy</i>	0.96	0.90	1	0.99
<i>Med Loss</i>	0.17	0.36	0,0002	0.04
h8	(13,1)	(13,1)	(7,1)	(10,1)
<i>Med Accuracy</i>	1	1	1	1
<i>Med Loss</i>	0.0003	0.0003	0,0002	0.0002
h9	(10,1)	(16,1)	(10,1)	(10,1)
<i>Med Accuracy</i>	1	0.97	1	1
<i>Med Loss</i>	0.0002	0.15	0,0002	0,0002
h10	(10,1)	(19,1)	(10,1)	(10,1)
<i>Med Accuracy</i>	1	0.99	1	1
<i>Med Loss</i>	0.0002	0.08	0,0002	0.0001

Asimismo, el nodo h7 mostró la mayor pérdida (*loss*) con un valor de 0.36 en la simulación rf-ct. Este resultado indica que, aunque el modelo puede estar prediciendo correctamente en muchos casos, sus predicciones para este nodo en esta simulación específica tienen un mayor grado de incertidumbre o error.

En cuanto a los parámetros del modelo, se observa una tendencia interesante en las combinaciones de (x , y) utilizadas para las predicciones. El valor de $x = 10$ fue el más frecuente, apareciendo 15 veces en total. Esto sugiere que, en promedio, analizar los 10 puntos de datos históricos más recientes proporciona un buen equilibrio entre la cantidad de información histórica y la capacidad predictiva del modelo. El valor mínimo de x encontrado fue 4, lo que indica que incluso con pocos datos históricos, el modelo puede hacer predicciones en algunos casos.

Es notable que el modelo logra predicciones precisas con relativamente pocos datos históricos (entre 10 y 19 puntos de datos en la mayoría de los casos) y un horizonte de predicción de 1 paso al futuro ($y = 1$ en todos los casos). Esto sugiere que el modelo puede capturar patrones relevantes en el comportamiento del SLA con una cantidad limitada de información histórica, y que es efectivo en predicciones a corto plazo. Al comparar estos resultados con los valores reales de cumplimiento del SLA proporcionados, se observa una correlación general entre el rendimiento del modelo y el nivel de cumplimiento del SLA. Los nodos con alto cumplimiento del SLA en la mayoría de las simulaciones (como en los nodos h5, h8, h9 y h10) también mostraron un rendimiento perfecto o casi perfecto en el modelo.

El modelo LSTM muestra ser una herramienta efectiva para predecir el comportamiento del SLA en la mayoría de los nodos y tipos de simulación. Sin embargo, los resultados también destacan áreas específicas de mejora, particularmente en los nodos h2 y h7 para ciertas simulaciones. La prevalencia de $x = 10$ como el valor más común para los datos históricos analizados proporciona una guía útil para futuras implementaciones del modelo. La capacidad del modelo para lograr alta precisión con relativamente pocos datos históricos, lo hace particularmente valioso para aplicaciones en tiempo real de gestión de redes, aunque se podrían explorar ajustes adicionales para mejorar el rendimiento en los escenarios más desafiantes, por lo que puede ser el modelo para predicción del comportamiento de los SLA.

4.5 Conclusiones parciales

Los resultados obtenidos en este estudio demuestran la eficacia del sistema propuesto para la gestión y

monitorización de SLA en redes 5G/B5G basadas en SDN. Las simulaciones realizadas revelaron mejoras significativas en el cumplimiento de los SLA y en la eficiencia general de la red cuando se implementaron estrategias de enrutamiento dinámico y priorización de tráfico.

- ✓ Se observó que la configuración de rutas optimizadas para cumplir con los SLA resultó en una reducción notable de la pérdida de paquetes, pasando de un 11.85 % en el escenario de enrutamiento por defecto a un 0% cuando se aplicaron las configuraciones optimizadas. Además, el porcentaje de cumplimiento de SLA aumentó de un 61.63% a un 99.93% en los escenarios más favorables y la latencia promedio disminuyó de 7.98 ms a 1.86 ms.
- ✓ La implementación de un sistema de priorización basado en las necesidades específicas de cada aplicación mostró ser crucial para garantizar la calidad de servicio en aplicaciones críticas como eHealth y automatización industrial, incluso en condiciones de alta congestión de la red. Este enfoque permitió una gestión más eficiente de los recursos de red, adaptándose a los diversos requisitos de QoS de las diferentes aplicaciones en entornos 5G y IoT avanzados.
- ✓ El modelo LSTM implementado para la predicción del comportamiento del SLA mostró ser altamente efectivo, se tiene una precisión casi perfecta en varios nodos, especialmente en los nodos h5, h8, h9 y h10. Esta capacidad predictiva, alcanzada con relativamente pocos datos históricos (generalmente 10 puntos), posiciona al modelo como una herramienta valiosa para la gestión de redes en tiempo real.

Estos hallazgos subrayan la importancia de implementar sistemas de gestión de red inteligentes y adaptativos en entornos 5G/B5G. La capacidad de ajustar dinámicamente las configuraciones de red según el monitoreo en tiempo real, y la predicción de comportamientos de SLA es una solución prometedora para maximizar la eficiencia de la red, y garantizar la calidad de servicio en aplicaciones con requisitos diversos y exigentes.

Capítulo 5. Conclusiones y recomendaciones

La investigación presentada en esta tesis ha desarrollado y evaluado un sistema novedoso de gestión y monitorización de Acuerdos de Nivel de Servicio (SLA) en redes 5G/B5G, que destaca por su capacidad para mejorar la eficiencia y adaptabilidad en entornos de red complejos. Este sistema es innovador porque integra de manera única varias tecnologías avanzadas, como Redes Definidas por Software (SDN), emulación con Mininet, un sistema robusto de monitorización, una API eficiente para el enrutamiento, y técnicas de Aprendizaje de Máquina (ML). Estas tecnologías no solo optimizan la gestión de recursos y la calidad del servicio, sino que también permiten una adaptabilidad superior en redes de próxima generación. Los resultados obtenidos validan la efectividad del sistema, mostrando avances significativos en la capacidad de respuesta y la flexibilidad en la gestión de redes altamente dinámicas y exigentes.

El sistema propuesto demostró una mejora sustancial en el cumplimiento de los SLA, con un incremento del 62.63% al 99.93% en los escenarios opuestos, representando una mejora total del 37.3%. La monitorización en ráfagas mostró un aumento del 12.3% en el cumplimiento del SLA, mientras que la monitorización individual mejoró en un 9.04%. Se logró una reducción significativa en la pérdida de paquetes, del 11.85% al 0% en condiciones óptimas, y la latencia promedio disminuyó de 7.98 ms a 1.86 ms. En todos los casos, se observó una mejora en el comportamiento de los parámetros al seleccionar las rutas de menor latencia que cumplían con el SLA para cada aplicación. Estas mejoras se alcanzaron mediante la implementación de estrategias de enrutamiento dinámico, priorización de tráfico y ajustes de los recursos de red. El sistema demostró su robustez en el manejo del tráfico y su capacidad para mejorar el cumplimiento del SLA en los diferentes escenarios.

Los parámetros de configuración de la simulación impactan significativamente el rendimiento de las aplicaciones y a menudo superan el efecto de la priorización asignada. No obstante, h4 (eHealth) y h6 (Automatización Industrial), con las prioridades más altas, muestran buenos valores de latencia (0.93 ms y 0.94 ms), lo que indica un efecto positivo de la priorización. El alto rendimiento de h10 (*Streaming* de Video UHD) y h9 (Vo5G), a pesar de la baja prioridad de h10, se atribuye al uso de UDP, tamaños de paquete grandes (1200 para h10) y, en el caso de Vo5G, intervalos de Ping cortos (0.01 s). Estas aplicaciones logran los mayores porcentajes de cumplimiento de SLA (99.5% y 98.5%). En contraste, h8 (ITS) y h5 (Smart Grid) presentan un bajo desempeño, con los porcentajes más bajos de cumplimiento de SLA (51.17% y 49.83%) debido a altos valores de latencia (17.9 ms y 9.5 ms) y pérdida de paquetes en h5 (15.89%). Las aplicaciones h2 (Experiencia Inmersiva) y h3 (Vigilancia de video) alcanzan un 84.33% de cumplimiento de SLA, con el

uso de UDP y altos requisitos de ancho de banda (50 Mbits/s y 10 Mbits/s), pero muestran altas pérdidas de paquetes (15.66%), afectadas principalmente cuando el controlador configura las rutas por defecto. En general, las aplicaciones con UDP, alto ancho de banda y paquetes grandes se benefician de la optimización de la red, mientras que la priorización resulta crucial para el rendimiento de aplicaciones críticas con TCP, como demuestran h4 y h6.

El papel central de las Redes Definidas por Software (SDN) en esta investigación ha sido fundamental, y proporciona una plataforma flexible y programable para la gestión de la red. La capacidad de SDN para separar el plano de control del plano de datos permitió una administración más dinámica y adaptativa, esencial para cumplir con los requisitos de los SLA en tiempo real. Esta flexibilidad se tradujo en una mejor utilización de los recursos de red y en la capacidad de responder rápidamente a las fluctuaciones en la demanda de servicios, aspectos críticos en el contexto de las redes 5G/B5G, donde la variabilidad y la exigencia de las aplicaciones son constantes.

La integración de SDN con Mininet mostró ser una combinación interesante, ya que facilita la simulación de entornos de red complejos y proporciona un laboratorio de pruebas robusto para evaluar la eficacia del sistema bajo diversas condiciones de tráfico y topologías de red. Esta sinergia permitió la validación exhaustiva del sistema propuesto en escenarios que reflejan la complejidad de las redes 5G/B5G sin los costos y riesgos asociados a las pruebas en infraestructuras de producción. El controlador OpenDaylight (ODL) jugó un papel crucial en este proceso, pues muestra ser una herramienta eficaz para la gestión flexible del tráfico de red, que se adapta a las demandas cambiantes y optimizando el rendimiento global del sistema.

Se exploró la inclusión de técnicas de Aprendizaje Automático (ML), específicamente mediante el uso de modelos LSTM, como un componente potencial para la predicción del comportamiento del SLA. Los resultados preliminares de esta exploración mostraron una precisión prometedora, cercana al 100% en varios nodos simulados (h5, h8, h9, h10), utilizando solo 10 puntos de datos históricos. Aunque esta integración no se completó en la implementación final del sistema, los hallazgos sugieren que tal capacidad predictiva es importante para anticipar posibles incumplimientos de SLA y tomar acciones proactivas, antes de que ocurran problemas reales en la red. La incorporación de ML tiene el potencial de mejorar significativamente la gestión de recursos y el enrutamiento de los flujos de tráfico, así como elevar la calidad de servicio y la satisfacción del usuario, aspectos cruciales en la era de las comunicaciones 5G/B5G.

La implementación y validación del sistema se realizaron en un entorno simulado, lo que, si bien proporciona una base sólida para la evaluación inicial, no captura todas las complejidades y variabilidades de un entorno de red real. Esta limitación, aunque común en investigaciones de esta naturaleza, destaca la importancia de las pruebas en entornos operativos reales como un paso crucial en la evolución futura del sistema. No obstante, la robustez y la consistencia de los resultados obtenidos en las simulaciones proporcionan una fuerte indicación de la viabilidad y el potencial del sistema propuesto en escenarios del mundo real.

Las implicaciones prácticas de estos resultados para la industria de las telecomunicaciones son sustanciales. El sistema propuesto ofrece una solución viable para la gestión dinámica de recursos en redes 5G/B5G, y permitiría a los operadores de red optimizar la utilización de recursos y mejorar la calidad de servicio. Esto podría traducirse en una mejor experiencia del usuario final, y en la posibilidad de ofrecer servicios más avanzados y personalizados, lo que abriría nuevas oportunidades de negocio y aceleraría la adopción de tecnologías emergentes que dependen de redes de alto rendimiento y baja latencia.

En conclusión, esta investigación mostró la viabilidad y eficacia de un sistema avanzado de gestión y monitorización de SLA en redes 5G/B5G, que integra tecnologías de vanguardia como SDN, emulación de redes, monitorización avanzada, API eficientes y técnicas de ML. Los resultados obtenidos no solo representan un avance significativo en el estado del arte de la gestión de redes de próxima generación, sino que también establecen una base sólida para futuras investigaciones y desarrollos en este campo crítico. A pesar de las limitaciones identificadas, especialmente en escenarios de latencia ultra baja, las conclusiones dan una hoja de ruta clara para desarrollar de soluciones más eficientes y adaptativas en la gestión de SLA en redes 5G y más allá, lo que marca un paso significativo hacia la realización del potencial completo de las tecnologías de comunicación avanzadas.

5.1 Contribuciones al conocimiento

Las contribuciones de esta investigación son múltiples y significativas en el campo de la gestión de redes 5G/B5G.

- ✓ Desarrollo de un sistema integrado de gestión y monitorización de SLA: se desarrolló un sistema que combina de manera innovadora SDN, emulación con Mininet, monitorización avanzada, API de

enrutamiento y técnicas de ML. Esta integración representa un enfoque holístico y pionero para la gestión de SLA en redes de próxima generación, superando las limitaciones de soluciones previas que abordaban estos aspectos de manera aislada.

- ✓ Mejora sustancial del cumplimiento de SLA: mediante enrutamiento dinámico se ha logrado un progreso del 37.3 %, lo que refleja una mejoría en la eficiencia de la gestión de recursos en redes 5G.
- ✓ Aplicación innovadora de modelos LSTM: la aplicación de modelos LSTM para la predicción de SLA con alta precisión, se utilizan solo 10 puntos de datos históricos, se marca un avance significativo en la capacidad de anticipación y respuesta proactiva a posibles incumplimientos de SLA.
- ✓ Metodología de priorización de tráfico: se ha desarrollado una metodología basada en requisitos de aplicaciones, lo que posibilita una gestión más eficiente de los recursos en entornos de red heterogéneos y de alta demanda.
- ✓ Herramienta para evaluación de rendimiento en redes 5G: se ha desarrollado una herramienta basada en Mininet y OpenDaylight que facilita la evaluación del rendimiento en redes 5G simuladas. Esta herramienta proporciona un entorno de pruebas robusto y flexible, que permite a los investigadores evaluar diversas condiciones de tráfico y topologías de red sin los costos y riesgos asociados a las pruebas en infraestructuras de producción.
- ✓ Estrategia de integración de SDN y ML: esta estrategia para la gestión adaptativa de redes establece una base sólida para futuros sistemas de redes autónomas y auto-optimizables.

En general, estas contribuciones abarcan tanto aspectos teóricos como prácticos, y abordan problemas críticos en la gestión de redes 5G/B5G. La combinación de mejoras cuantificables (como la optimización del cumplimiento de SLA) con desarrollos innovadores (como la aplicación de modelos LSTM y la herramienta de evaluación) hace que su trabajo sea particularmente valioso y relevante en el campo. Estas contribuciones no solo avanzan el estado del arte, sino que también proporcionan herramientas y metodologías concretas que pueden ser utilizadas y expandidas en futuras investigaciones y aplicaciones prácticas. Además, el enfoque presentado sienta las bases para nuevas líneas de investigación enfocadas

en la automatización completa de redes a través de la inteligencia artificial y su integración con tecnologías emergentes en redes de comunicaciones.

5.2 Limitaciones de la investigación

A pesar de los logros significativos, la investigación ha identificado varias limitaciones importantes.

La implementación y validación del sistema se realizaron en un entorno simulado, lo que limita capturar completamente las complejidades y variabilidades de un entorno de red real. Esta limitación, subraya la necesidad de pruebas con datos operativos reales, paso crucial en la evolución futura del sistema.

Otra limitante identificada, es la dependencia en datos históricos para la predicción, que, aunque mostró buenos resultados, podría beneficiarse de conjuntos de datos extensos para mejorar la precisión e incrementar la robustez de las predicciones.

Se identificaron desafíos específicos en el rendimiento de aplicaciones con requisitos de bajo ancho de banda, particularmente h5 (Smart Grid) y h8 (ITS). Estas aplicaciones mostraron los porcentajes más bajos de cumplimiento de SLA (49.83% y 51.17% respectivamente), lo que indica la necesidad de desarrollar estrategias más especializadas para optimizar su rendimiento en redes 5G/B5G.

Estas limitaciones destacan áreas críticas de continuar la investigación y el desarrollo de esquemas que aborden estos desafíos de perfeccionar el desempeño de redes inalámbricas móviles reales.

5.3 Trabajos futuros

Los trabajos futuros en esta línea de investigación deben enfocarse en varios aspectos clave para mejorar y expandir las capacidades del sistema propuesto.

En primer lugar, es crucial la integración completa del modelo LSTM con la API del sistema. Aunque se ha demostrado la viabilidad del uso de modelos LSTM para la predicción de SLA con alta precisión utilizando solo 10 puntos de datos históricos, la implementación práctica y la programación de estos modelos para

cada nodo de la red es un paso fundamental que aún está pendiente. Este esfuerzo permitirá anticipar posibles incumplimientos de SLA y tomar medidas correctivas de manera proactiva antes de que ocurran problemas reales en la red, mejorando significativamente la gestión de recursos y la calidad del servicio.

Además, se deben realizar pruebas extensivas en entornos operativos reales. La implementación y validación del sistema en un entorno simulado, aunque proporciona una base sólida para la evaluación inicial, no captura todas las complejidades y variabilidades de un entorno de red real. Por tanto, es esencial realizar pruebas en condiciones del mundo real para validar la eficacia del sistema y ajustar sus componentes según los resultados obtenidos.

Otro aspecto importante para la mejora de la precisión y la robustez de las predicciones del modelo LSTM, es necesario utilizar base de datos para el entrenamiento del modelo, es decir, el uso de conjuntos de datos más extensos.

Se debe prestar especial atención a la optimización del rendimiento de aplicaciones con requisitos de bajo ancho de banda, particularmente h5 (Smart Grid) y h8 (ITS), que mostraron los porcentajes más bajos de cumplimiento de SLA (49.83% y 51.17% respectivamente). Desarrollar estrategias más especializadas para estas aplicaciones críticas es fundamental para garantizar un rendimiento óptimo en toda la gama de servicios 5G/B5G como seleccionar rutas que no solo cumplan con el SLA, sino que también prioricen un mayor ancho de banda o una menor pérdida de paquetes, según el parámetro más crítico para cada aplicación.

Los trabajos futuros de forma general deberán centrarse en la integración completa del modelo LSTM con la API, la realización de pruebas en entornos operativos reales, el incremento de los datos utilizados para el entrenamiento del modelo, la optimización de servicios con requisitos de latencia baja y el perfeccionamiento del rendimiento para aplicaciones de bajo ancho de banda como Smart Grid e ITS.

Literatura citada

- 3rd Generation Partnership Project. (2017). *Technical specification*. <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>
- 3rd Generation Partnership Project. (2018). *Technical report*. Study on security aspects of the 5G Service Based Architecture. <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3481>
- 3rd Generation Partnership Project. (2019a). *5G System Overview*. <https://www.3gpp.org/technologies/5g-system-overview>
- 3rd Generation Partnership Project. (2019b). *Technical report*. Study on User Plane Function (UPF) enhancement for control and 5G Service Based Architecture (SBA). <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3623>
- 5G Americas Whitepaper. (2017). *5G Service and Use Cases*. <https://www.5gamericas.org/white-papers/>
- Afanasyev, A., Tilley, N., Reiher, P., & Kleinrock, L. (2010). Host-to-host congestion control for TCP. *IEEE Communications Surveys and Tutorials*. <https://doi.org/10.1109/SURV.2010.042710.00114>
- Agiwal, M., Roy, A., & Saxena, N. (2016). Next generation 5G wireless networks: A comprehensive survey. *IEEE Communications Surveys and Tutorials*. <https://doi.org/10.1109/COMST.2016.2532458>
- Ahad, A., Tahir, M., & Yau, K. L. A. (2019). 5G-based smart healthcare network: Architecture, taxonomy, challenges and future research directions. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2019.2930628>
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics (Switzerland)*. <https://doi.org/10.3390/electronics8030292>
- Amazon Web Services. (2024, mayo 10). *Welcome to AWS Documentation*. <https://docs.aws.amazon.com/>
- Azzure (2024, junio 15). *Azure documentation*. <https://learn.microsoft.com/en-us/azure/?product=popular>
- Barakabitze, A. A., Ahmad, A., Mijumbi, R., & Hines, A. (2020). 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. *Computer Networks*. <https://doi.org/10.1016/J.COMNET.2019.106984>
- Botta, A., Dainotti, A., & Pescapé, A. (2012). A tool for the generation of realistic network workload for emerging networking scenarios. *Computer Networks*. <https://doi.org/10.1016/J.COMNET.2012.02.019>

- Bouras, C., Kollia, A., & Papazois, A. (2017). SDN & NFV in 5G: Advancements and challenges. *Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks, ICIN 2017*. <https://doi.org/10.1109/ICIN.2017.7899398>
- Cao, H., Du, J., Zhao, H., Luo, D. X., Kumar, N., Yang, L., & Yu, F. R. (2022). Toward Tailored Resource Allocation of Slices in 6G Networks with Softwarization and Virtualization. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2021.3111644>
- Cao, J., Cleveland, W. S., Lin, D., & Sun, D. X. (2003). *Internet Traffic Tends Toward Poisson and Independent as the Load Increases*. https://doi.org/10.1007/978-0-387-21579-2_6
- Distrital Francisco Jose de Caldas, U., Eduardo Cáceres Guevara, J., & Alexis Casilimas Fajardo, C. (2021). *Arquitectura y funcionamiento de redes definidas por software SDN*. <https://repository.udistrital.edu.co/items/95898da8-9a46-4a2b-8ef6-0fc7077c5561>
- Docomo, N. (2020). *White Paper 5G Evolution and 6G*. <https://doi.org/10.1145/3427796.3432714>
- Fahad Taha Al-Dhief, N. S. M. A. (2018). *Performance Comparison between TCP and UDP Protocols in Different Simulation Scenarios*. https://www.researchgate.net/publication/329698255_Performance_Comparison_between_TCP_and_UDP_Protocols_in_Different_Simulation_Scenarios
- Ferdian Thung. (2016). *API recommendation system for software development*. <https://ieeexplore.ieee.org/document/7582836>
- Flask. (2024, junio 5). *Welcome to Flask*. <https://flask.palletsprojects.com/en/3.0.x/>
- Floodlight. (2024, marzo 25). *Floodlight Controller - Confluence*. <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/overview>
- Gohil, A., Modi, H., & Patel, S. K. (2013). 5G technology of mobile communication: A survey. *2013 International Conference on Intelligent Systems and Signal Processing, ISSP 2013*. <https://doi.org/10.1109/ISSP.2013.6526920>
- Gonzalez-Franco, J. D., Preciado-Velasco, J. E., & Rivera-Rodriguez, R. (2023). *Reconocimiento de patrones en métricas de tráfico para el cumplimiento de los SLA (Service Level Agreement) en entorno de redes B5G (Beyond 5G) [CICESE]*. <https://doi.org/doi:10.3390/fi15100335>
- Google Cloud. (2024, mayo 22). *Documentación de Google Cloud*. <https://cloud.google.com/docs?hl=es-419>
- Graphical Network Simulator 3. (2024, febrero 13). *GNS3 Documentation*. <https://docs.gns3.com/docs/>
- Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*. <https://doi.org/10.1109/TNNLS.2016.2582924>
- Habibi, M. A., Han, B., Nasimi, M., & Schotten, H. D. (2018). *The Structure of Service Level Agreement of Slice-based 5G Network*. <https://doi.org/10.48550/arXiv.1806.10426>

- Hossain, E., & Hasan, M. (2015). 5G cellular: Key enabling technologies and research challenges. *IEEE Instrumentation and Measurement Magazine*. <https://doi.org/10.1109/MIM.2015.7108393>
- International Telecommunication Union. (2015). *ITU-R Recommendations M.2083-0*. <http://www.itu.int/ITU-R/go/patents/en>
- Iperf. (2024, mayo 29). *iPerf - The TCP, UDP and SCTP network bandwidth measurement tool*. <https://iperf.fr/>
- Jalalian, A., Yousefi, S., & Kunz, T. (2023). Network slicing in virtualized 5G Core with VNF sharing. *Journal of Network and Computer Applications*. <https://doi.org/10.1016/J.JNCA.2023.103631>
- Janocha, K., & Czarnecki, W. M. (2017). On Loss Functions for Deep Neural Networks in Classification. *Schedae Informaticae*. <https://doi.org/10.4467/20838476SI.16.004.6185>
- Juniper Networks. (2023). *NFV*. <https://www.juniper.net/us/en/research-topics/what-is-network-functions-virtualization-nfv.html>
- Kapassa, E., Touloupou, M., Stavrianos, P., Xylouris, G., & Kyriazis, D. (2019). Managing and optimizing quality of service in 5G environments across the complete SLA lifecycle. *Advances in Science, Technology and Engineering Systems*. <https://doi.org/10.25046/AJ040132>
- Karagiannis, T., Molle, M. L., & Faloutsos, M. (2004). Long-range dependence: Ten years of internet traffic modeling. *IEEE Internet Computing*. <https://doi.org/10.1109/MIC.2004.46>
- Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). *Visualizing and Understanding Recurrent Networks*. <http://arxiv.org/abs/1506.02078>
- Khan, M. T., & Adholiya, A. (2023). Machine Learning-Based Application for Predicting 5G/B5G Service. *Proceedings of the 13th International Conference on Cloud Computing, Data Science and Engineering, Confluence 2023*. <https://doi.org/10.1109/CONFLUENCE56041.2023.10048885>
- Kosiński, J., Nawrocki, P., Radziszowski, D., Zieliński, K., Zieliński, S., Przybylski, G., & Wnęk, P. (2008). SLA monitoring and management framework for telecommunication services. *Proceedings - 4th International Conference on Networking and Services, ICNS 2008*. <https://doi.org/10.1109/ICNS.2008.31>
- Kurose, J. F., & Ross, K. W. (2017). *Computer networking: a top-down approach: Vol. Seventh Edition*. <http://lccn.loc.gov/2016004976>
- León, D. A., Martínez, J. G., Ardila D J Mosquera, I. A., & Distrital Francisco José De Caldas, U. (2022). Artificial intelligence for traffic control in data networks: A Review. *Entre Ciencia e Ingeniería*. <https://doi.org/10.31908/19098367.2655>
- Li, X., Li, D., Wan, J., Vasilakos, A. V., Lai, C. F., & Wang, S. (2017). A review of industrial wireless networks in the context of Industry 4.0. *Wireless Networks*. <https://doi.org/10.1007/S11276-015-1133-7/METRICS>
- López Telenchana, L. S. L., Serrano Torres, G. J., Quintana López, X. A., & Reina Haro, D. M. (2024). Machine Learning en la industria 4.0: una revisión sistematica. *Salud, Ciencia y Tecnología, ISSN 2796-9711, No.*

4, 2024 (Ejemplar dedicado a: Salud, Ciencia y Tecnología). <https://dialnet.unirioja.es/servlet/articulo?codigo=9563564&info=resumen&idioma=ENG>

Lozano-Rizk, J. E., Gonzalez-Trejo, J. E., Rivera-Rodriguez, R., Tchernykh, A., Villarreal-Reyes, S., & Galaviz-Mosqueda, A. (2022). Application-Aware Flow Forwarding Service for SDN-Based Data Centers. *Electronics* 2022. <https://doi.org/10.3390/ELECTRONICS11233882>

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015). Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*. <https://doi.org/10.1109/TITS.2014.2345663>

Mininet. (2024, enero 25). *Mininet: An Instant Virtual Network on Your Laptop*. <https://mininet.org/>

Mininet GitHub. (2021, enero 22). *Documentation*. <https://github.com/mininet/mininet/wiki/Documentation>

Nam Nguyen, X., Saucez, D., Barakat, C., Turletti, T., & Nguyen, X.-N. (2016). *Rules Placement Problem in OpenFlow Networks: a Survey*. <https://doi.org/10.1109/COMST.2015.2506984i>

Network Simulator 3. (2024, marzo 22). *Documentation*. <https://www.nsnam.org/documentation/>

Nightingale, J., Salva-Garcia, P., Calero, J. M. A., & Wang, Q. (2018). 5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks. *IEEE Transactions on Broadcasting*. <https://doi.org/10.1109/TBC.2018.2816786>

OMNet. (2024, febrero 12). *OMNeT++ Documentation*. <https://omnetpp.org/documentation/>

Open Network Operating System. (2024, febrero 12). *ONOS SDN Controller*. <https://opennetworking.org/onos/>

OpenDaylight. (2024, febrero 2). *Welcome to OpenDaylight Documentation*. <https://docs.opendaylight.org/en/stable-potassium/>

Ordóñez-Lucena, J., Ameigeiras, P., Lopez, Di., Ramos-Munoz, J. J., Lorca, J., & Folgueira, J. (2017). Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. *IEEE Communications Magazine*, 5. <https://doi.org/10.1109/MCOM.2017.1600935>

Paessler. (2024, junio 13). *Ping - Definition and details*. <https://www.paessler.com/es/it-explained/ping>

Pavani, A., & Kathirvel, A. (2023). Machine Learning and Deep Learning Algorithms for Network Data Analytics Function in 5G Cellular Networks. *6th International Conference on Inventive Computation Technologies, ICICT 2023 - Proceedings*. <https://doi.org/10.1109/ICICT57646.2023.10134247>

Pocovi, G., Shariatmadari, H., Berardinelli, G., Pedersen, K., Steiner, J., & Li, Z. (2018). Achieving Ultra-Reliable Low-Latency Communications: Challenges and Envisioned System Enhancements. *IEEE Network*. <https://doi.org/10.1109/MNET.2018.1700257>

Postman. (2024, mayo 22). *Postman API Platform*. <https://www.postman.com/>

POX. (2024, marzo 22). *POX Manual Current documentation*. <https://noxrepo.github.io/pox-doc/html/>

- Python. (2024, abril 25). *Welcome to Python.org*. <https://www.python.org/>
- Rafael Zambrano. (2019). *Diferencias entre Machine Learning y Deep Learning*. Rafael Zambrano. <https://openwebinars.net/blog/diferencias-entre-machine-learning-y-deep-learning/>
- Ramos-García, R., Preciado-Velasco, J. E., & Lozano-Risk, J. E. (2021). *Predicción del comportamiento de SLA en redes 5G utilizando inteligencia artificial* [CICESE]. <http://cicese.repositorioinstitucional.mx/jspui/handle/1007/3656>
- Rekkas, V. P., Sotiroudis, S., Sarigiannidis, P., Wan, S., Karagiannidis, G. K., & Goudos, S. K. (2021). Machine learning in beyond 5g/6g networks state of the art and future trends. En *Electronics (Switzerland)*. MDPI. <https://doi.org/10.3390/electronics10222786>
- Ryait, D. K., & Sharma, M. (2023). Performance Evaluation of SDN Controllers. *Lecture Notes in Networks and Systems*. https://doi.org/10.1007/978-981-99-5166-6_68
- Ryu. (2024, marzo 25). *Getting Started Ryu 4.34 documentation*. <https://ryu.readthedocs.io/en/latest/parameters.html>
- Saad, S. Ben, Ksentini, A., & Brik, B. (2021). A Trust architecture for the SLA management in 5G networks. *IEEE International Conference on Communications*. <https://doi.org/10.1109/ICC42927.2021.9500990>
- Sayed, S. A., Abdel-Hamid, Y., & Hefny, H. A. (2022). *Artificial Intelligence-Based Traffic Flow Prediction: A Comprehensive Review*. <https://doi.org/10.21203/rs.3.rs-1885747/v1>
- Shaho Ismael Hassen, A. A. (2023). *Heterogeneous SDN Controllers for Dynamic Environments with Different Network Requirements*. https://www.researchgate.net/publication/373948498_Heterogeneous_SDN_Controllers_for_Dynamic_Environments_With_Different_Network_Requirements
- Siddiqui, M. U. A., Abumarshoud, H., Bariah, L., Muhaidat, S., Imran, M. A., & Mohjazi, L. (2023). URLLC in Beyond 5G and 6G Networks: An Interference Management Perspective. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2023.3282363>
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*. <https://doi.org/10.1016/J.IPM.2009.03.002>
- Storck, C. R., & Duarte-Figueiredo, F. (2019). A 5G V2X ecosystem providing internet of vehicles. *Sensors (Switzerland)*. <https://doi.org/10.3390/s19030550>
- Talavera, E., Díaz-Álvarez, A., Jiménez, F., & Naranjo, J. E. (2018). Impact on congestion and fuel consumption of a cooperative adaptive cruise control system with lane-level position estimation. *Energies*. <https://doi.org/10.3390/EN11010194>
- Touloupou, M., Kapassa, E., Symvoulidis, C., Stavrianos, P., & Kyriazis, D. (2019). An Integrated SLA Management Framework in a 5G Environment. *22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. <https://doi.org/10.1109/ICIN.2019.8685916>

- Troia, S., Alvizu, R., Zhou, Y., Maier, G., & Pattavina, A. (2018). Deep Learning-Based Traffic Prediction for Network Optimization. *International Conference on Transparent Optical Networks*. <https://doi.org/10.1109/ICTON.2018.8473978>
- Usman, M., Gebremariam, A. A., Raza, U., & Granelli, F. (2015). A Software-Defined Device-to-Device Communication Architecture for Public Safety Applications in 5G Networks. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2015.2479855>
- Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Applying deep learning approaches for network traffic prediction. *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*. <https://doi.org/10.1109/ICACCI.2017.8126198>
- Visual Studio Code. (2024, junio 1). *Documentation*. <https://code.visualstudio.com/docs>
- Wang, K., Yu, J., Yu, Y., Qian, Y., Zeng, D., Guo, S., Xiang, Y., & Wu, J. (2018). A survey on energy internet: Architecture, approach, and emerging technologies. *IEEE Systems Journal*. <https://doi.org/10.1109/jsyst.2016.2639820>
- Wu, Y. J., Chen, F., Lu, C. T., & Yang, S. (2016). Urban Traffic Flow Prediction Using a Spatio-Temporal Random Effects Model. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*. <https://doi.org/10.1080/15472450.2015.1072050>
- Yousaf, F. Z., Bredel, M., Schaller, S., & Schneider, F. (2017). NFV and SDN-Key technology enablers for 5G networks. *IEEE Journal on Selected Areas in Communications*. <https://doi.org/10.1109/JSAC.2017.2760418>
- Yuan, H., & Li, G. (2021). A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation. *Data Science and Engineering*, 1. <https://doi.org/10.1007/S41019-020-00151-Z/TABLES/3>
- Zhang, S. (2019). An Overview of Network Slicing for 5G. *IEEE Wireless Communications*. <https://doi.org/10.1109/MWC.2019.1800234>

Anexos

Anexo A

Ejemplo del log de cómo se registran los datos en el sistema de monitorización y como se guardan en la bitácora.

```

Archivo  Editar  Ver

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 43ms
rtt min/avg/max/mdev = 6.070/6.963/10.370/1.703 ms
-----
Client connecting to 10.0.0.2, UDP port 5001
Sending 1200 byte datagrams, IPG target: 153.66 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 3] local 10.0.0.1 port 40067 connected with 10.0.0.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  74.5 MBytes 62.5 Mbits/sec
[ 3] Sent 65076 datagrams

2024-07-16 17:19:52 - h3 - video_surveillance - Results:
PING 10.0.0.3 (10.0.0.3) 800(828) bytes of data.
808 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=5.99 ms
808 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=4.04 ms
808 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=4.24 ms
808 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=4.12 ms

--- 10.0.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 30ms
rtt min/avg/max/mdev = 4.036/4.598/5.994/0.810 ms
-----
Client connecting to 10.0.0.3, UDP port 5001
Sending 800 byte datagrams, IPG target: 610.35 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 3] local 10.0.0.1 port 37893 connected with 10.0.0.3 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  12.5 MBytes 10.5 Mbits/sec
A[ 3] Sent 16384 datagrams

Ln 40, Col 2 | 2,166,143 caracteres. | 100% | Unix (LF)

```

Figura 24. Log de los resultados del Sistema de Monitorización.

A	B	C	D	E	F	G	H	I
Fecha	Hora	Nodo	Tipo	PaketLoss	Latencia	Jitter	BitRate	SLA
2024-07-14	00:00:05	h5	usd-st	0	2.0535	0.097	1.03	SI
2024-07-14	00:01:05	h6	usd-st	0	1.0345	0.1245	1.05	NO
2024-07-14	00:02:06	h7	usd-st	0	1.68	0.097	9.73	NO
2024-07-14	00:03:07	h8	usd-st	0	1.6225	0.1725	0.524	SI
2024-07-14	00:04:07	h9	usd-st	0	2.082	0.236	10.5	SI
2024-07-14	00:05:07	h10	usd-st	0	1.7915	0.121	22.9	SI
2024-07-15	00:08:07	h7	usd-ct	0	1.7965	0.251	24.5	SI
2024-07-14	00:08:10	h4	usd-st	0	0.9275	0.1435	15.5	NO
2024-07-14	00:09:10	h5	usd-st	0	1.907	0.086	1.03	SI
2024-07-15	00:10:09	h9	usd-ct	0	2.0925	0.247	10.5	SI
2024-07-14	00:10:10	h6	usd-st	0	1.038	0.164	1.02	NO
2024-07-15	00:11:11	h10	usd-ct	0	2.126	0.266	17.7	SI
2024-07-14	00:11:12	h7	usd-st	0	1.7415	0.179	9.69	NO
2024-07-13	00:11:51	h9	usd-st	0	2.086	0.2195	11.5	SI
2024-07-14	00:12:12	h8	usd-st	0	1.965	0.1865	0.524	SI
2024-07-13	00:12:52	h10	usd-st	0	1.5995	0.2525	25.3	SI
2024-07-14	00:13:13	h9	usd-st	0	2.1415	0.2415	10.5	SI
2024-07-14	00:14:13	h10	usd-st	0	1.59	0.252	16.6	SI
2024-07-15	00:16:14	h6	usd-ct	0	0.9675	0.0955	1.04	SI
2024-07-15	00:17:15	h7	usd-ct	0	1.8735	0.396	23.2	SI
2024-07-14	00:17:16	h4	usd-st	0	1.0125	0.0525	15.5	NO
2024-07-14	00:18:16	h5	usd-st	0	2.014	0.1335	1.05	SI

Figura 25. Ejemplo de la Bitácora de las mediciones.

Anexo B

En este Anexo se muestran los datos para las gráficas mostradas en el Capítulo 4.

Tabla 8. Porcentaje de SLA por nodo.

Nodo	rf-st	rf-ct	usd-st	usd-ct
h2	60.67	98	78.67	100
h3	60.67	98	78.67	100
h4	79.33	92	94.67	100
h5	0	0	99.33	100
h6	60	94.67	90	100
h7	94	86	76.67	99.33
h8	0	4.67	100	100
h9	100	94	100	100
h10	100	98	100	100

Tabla 9. Latencia promedio por nodo.

Nodo	rf-st	rf-ct	usd-st	usd-ct
h2	3.47	3.28	3.4	3.44
h3	2.4	2.22	2.4	2.43
h4	0.954	0.93	0.9319	0.92
h5	21.43	12.2	2.03	2.05
h6	0.953	0.936	0.936	0.92
h7	1.93	1.67	1.62	1.54
h8	38.66	29.78	1.65	1.58
h9	2.39	2	2.1	2.01
h10	1.94	1.79	1.88	1.88

Tabla 10. Porcentaje de Pérdidas de paquetes por nodo.

Nodo	rf-st	rf-ct	usd-st	usd-ct
h2	39.33	2	21.33	0
h3	39.33	2	21.33	0
h4	0	2	0	0
h5	28	35.55	0	0
h6	0	6	0	0
h7	0	6	0	0
h8	0	6.16	0	0
h9	0	6	0	0
h10	0	2	0	0

Anexo C

Ejemplo de cómo se configura una regla del flujo en el controlador ODL

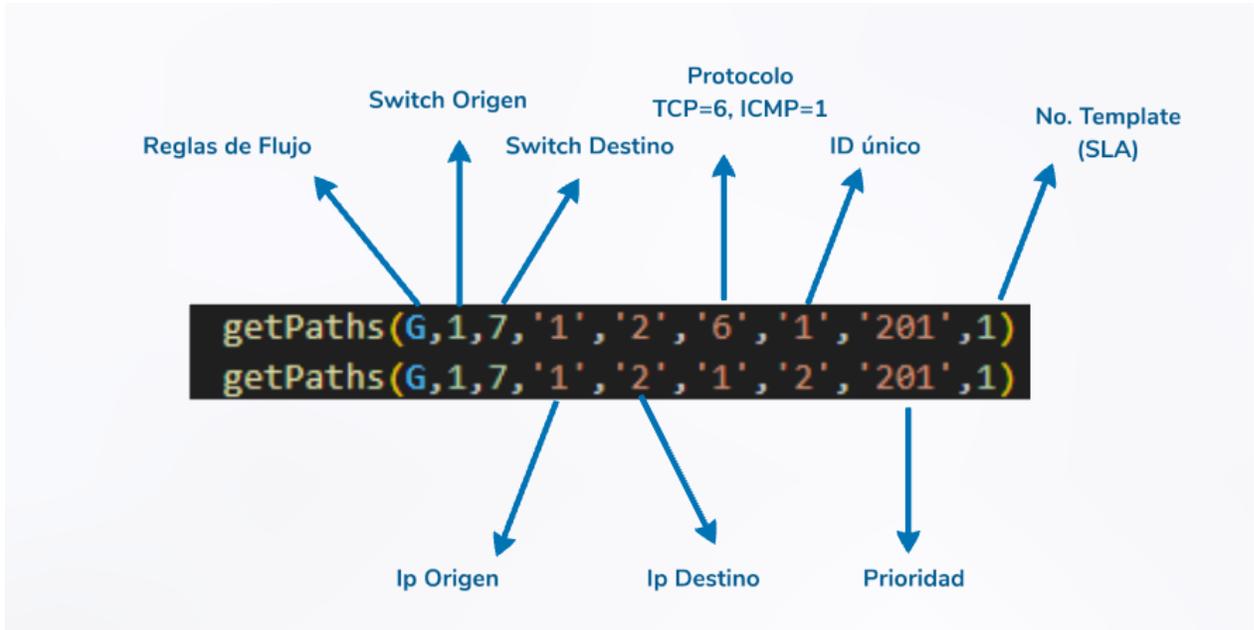


Figura 26. Configuración regla del flujo en el controlador ODL.

Como queda configurada la regla de flujo en el switch en este caso el switch s0

```
*** Starting CLI:
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s0
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=270,icmp,nw_src=10.0.0.1,nw_dst=10.0.0.6 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=260,icmp,nw_src=10.0.0.1,nw_dst=10.0.0.7 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=260,tcp,nw_src=10.0.0.1,nw_dst=10.0.0.7 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=230,icmp,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=210,icmp,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=280,tcp,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=230,udp,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=250,tcp,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=270,tcp,nw_src=10.0.0.1,nw_dst=10.0.0.6 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=250,icmp,nw_src=10.0.0.1,nw_dst=10.0.0.5 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=210,udp,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=220,icmp,nw_src=10.0.0.1,nw_dst=10.0.0.8 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=220,tcp,nw_src=10.0.0.1,nw_dst=10.0.0.8 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=240,udp,nw_src=10.0.0.1,nw_dst=10.0.0.9 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=240,icmp,nw_src=10.0.0.1,nw_dst=10.0.0.9 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=200,udp,nw_src=10.0.0.1,nw_dst=10.0.0.10 actions=output:"s0-eth2"
cookie=0x67, duration=8.708s, table=0, n_packets=0, n_bytes=0, idle_timeout=1800, hard_timeout=1800, priority=200,icmp,nw_src=10.0.0.1,nw_dst=10.0.0.10 actions=output:"s0-eth2"
cookie=0x2b00000000000001, duration=9.010s, table=0, n_packets=6, n_bytes=510, priority=100,dl_type=0x88cc actions=CONTROLLER:65535
cookie=0x2b00000000000004, duration=0.969s, table=0, n_packets=4, n_bytes=280, priority=2,in_port="s0-eth3" actions=output:"s0-eth1"
cookie=0x2b00000000000005, duration=0.969s, table=0, n_packets=1, n_bytes=70, priority=2,in_port="s0-eth1" actions=output:"s0-eth3",CONTROLLER:65535
cookie=0x2b00000000000000, duration=9.010s, table=0, n_packets=3, n_bytes=210, priority=0 actions=drop
mininet>
```

Figura 27. Reglas de Flujo en el switch s0.

Anexo D

Código utilizado para la predicción del comportamiento de los SLA utilizando el modelo LSTM.

```
C: > Users > Pixel > Downloads > model_time_series.py
1  import pandas as pd
2  import numpy as np
3  from sklearn.preprocessing import StandardScaler
4  from sklearn.model_selection import train_test_split
5  from tensorflow.keras.models import Sequential
6  from tensorflow.keras.layers import LSTM, Dense
7
8
9
10 def create_sequences(df, sequence_length=5, time_after=0):
11     X, y = [], []
12     for i in range(len(df) - sequence_length - time_after):
13         X.append(df.iloc[i:i + sequence_length, :-1].values)
14         y.append(df.iloc[i + sequence_length + time_after, -1])
15     return np.array(X), np.array(y)
16
17
18 def evalua_modo(temporal, sequence_length, time_after):
19     X, y = create_sequences(temporal, sequence_length, time_after)
20     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
21     model = Sequential()
22     model.add(LSTM(50, input_shape=(sequence_length, X.shape[2])))
23     model.add(Dense(1, activation='sigmoid'))
24     # Compilar el modelo
25     model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
26     # Entrenar el modelo
27     model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_test))
28     # Evaluar el modelo
29     loss, accuracy = model.evaluate(X_test, y_test)
30     return loss, accuracy
```

Figura 28. Código predicción con LSTM.