



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE TELECOMUNICACIONES**

**Análisis del balanceo de carga en una red 4G LTE e implementación de
un algoritmo de handover horizontal basado en aprendizaje por refuerzo
para mejorar la calidad de servicio (QoS)**

**Trabajo de titulación para optar al título de:
INGENIERO EN TELECOMUNICACIONES**

Autor:

Quiroz Pérez Gristel Alejandra

Tutor:

PhD. Pedro Fernando Escudero Villa

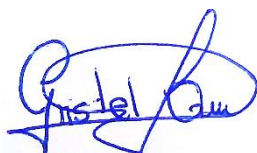
Riobamba, Ecuador. 2025

DECLARATORIA DE AUTORÍA

Yo, **Gristel Alejandra Quiroz Pérez**, con cédula de ciudadanía 0604763375, autora del trabajo de investigación titulado: **Análisis del balanceo de carga en una red 4G LTE e implementación de un algoritmo de handover horizontal basado en aprendizaje por refuerzo para mejorar la calidad de servicio (QoS)**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autora de la obra referida será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 14 de junio de 2025.



Gristel Alejandra Quiroz Pérez
C.I: 0604763375

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

En la Ciudad de Riobamba, a los 23 días del mes de julio de 2025, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por el estudiante **GRISTEL ALEJANDRA QUIROZ PÉREZ** con CC: **0604763375**, de la carrera **TELECOMUNICACIONES** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado “**ANÁLISIS DEL BALANCEO DE CARGA EN UNA RED 4G LTE E IMPLEMENTACIÓN DE UN ALGORITMO DE HANDOVER HORIZONTAL BASADO EN APRENDIZAJE POR REFUERZO PARA MEJORAR LA CALIDAD DE SERVICIO (QOS)**”, por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



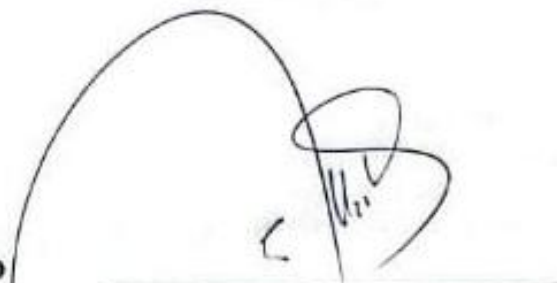
PhD. Pedro Fernando Escudero Villa
TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Análisis del balanceo de carga en una red 4G LTE e implementación de un algoritmo de handover horizontal basado en aprendizaje por refuerzo para mejorar la calidad de servicio (QoS)**, presentado por **Gristel Alejandra Quiroz Pérez**, con cédula de identidad número **0604763375**, bajo la tutoría de **PhD. Pedro Fernando Escudero Villa**; certificamos que recomendamos la **APROBACIÓN** de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor, no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba a la fecha de su presentación.

Marlon Danilo Basantes Valverde, PhD.
PRESIDENTE DEL TRIBUNAL DE GRADO

A large, stylized handwritten signature in black ink, written over a horizontal line.

Eduardo Daniel Haro Mendoza, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO

A handwritten signature in black ink, written over a horizontal line.

Luis Patricio Tello Oquendo, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO

A handwritten signature in black ink, written over a horizontal line.

CERTIFICADO ANTIPLAGIO



Dirección
Académica
VICERRECTORADO ACADÉMICO



CERTIFICACIÓN

Que, **QUIROZ PÉREZ GRISTEL ALEJANDRA** con CC: **0604763375**, estudiante de la Carrera **INGENIERÍA EN TELECOMUNICACIONES**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**ANÁLISIS DEL BALANCEO DE CARGA EN UNA RED 4G LTE E IMPLEMENTACIÓN DE UN ALGORITMO DE HANDOVER HORIZONTAL BASADO EN APRENDIZAJE POR REFUERZO PARA MEJORAR LA CALIDAD DE SERVICIO (QOS)**", cumple con el 1% de acuerdo al reporte del sistema Anti plagio **COMPILATIO**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 27 de octubre de 2025

PhD. Pedro Fernando Escudero Villa
TUTOR

DEDICATORIA

Dedico el presente trabajo a mi familia, quienes han sido parte fundamental de este camino. En especial a mi padre Sergio, gracias por creer en mí y darme la oportunidad de continuar con mi sueño, por ser la inspiración para crecer y ser una mejor persona, por ser un ejemplo para no rendirse, por cuidarme, entenderme y procurar que tenga lo necesario para salir adelante.

A mi madre Esperanza y mi hermana Mayra por estar presentes en cada etapa, por acompañarme y ayudarme en los momentos más complicados. Su presencia y apoyo han significado mucho para mí. A mi tía Angelina que ha sido como mi segunda madre, gracias, por lo gestos sencillos pero que han sido muy valiosos.

Y, sobre todo, me lo dedico a mí misma.

Porque, a pesar de las dudas, el cansancio y las veces que pensé que no iba a poder, seguí adelante. Me agradezco por haber sido fuerte y haber confiado en mí. Por demostrarle a mi yo del pasado que no hay obstáculo que no pueda ser vencido.

AGRADECIMIENTO

A lo largo de este proceso he aprendido que los logros no se alcanzan en soledad. Siempre se encuentra alguien en el camino que nos ayuda, nos acompaña o simplemente está presente con sus palabras de apoyo.

En primer lugar, quiero agradecer a Dios, por darme la fuerza de continuar, por recordarme que todo es posible si confío.

Agradezco los docentes por su compromiso y dedicación a lo largo de mi formación académica, y en especial a mi tutor de tesis, PhD. Pedro Fernando Escudero, por guiarme con paciencia, por sus observaciones claras y oportunas, y por confiar en mi capacidad para culminar el trabajo.

Un agradecimiento especial para mis amigos, quienes nos hemos encontrado desde la primera etapa de este proceso y quienes se fueron sumando en el camino. Gracias porque su compañía, apoyo y amistad hicieron de este recorrido una experiencia inolvidable.

Por último, pero no menos importante. Gracias, Riki, por tu compañía, por tu paciencia y por ser mi fuente de luz en mis días más oscuros, por brindarme tu apoyo de forma incondicional. Gracias, por tanto.

ÍNDICE GENERAL

| | |
|--|--|
| DECLARATORIA DE AUTORÍA | |
| DICTAMEN FAVORABLE DEL PROFESOR TUTOR | |
| CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL | |
| CERTIFICADO ANTIPLAGIO | |
| DEDICATORIA | |
| AGRADECIMIENTO | |
| ÍNDICE DE TABLAS | |
| ÍNDICE DE FIGURAS | |
| RESUMEN | |
| ABSTRACT | |

| | |
|---|-----------|
| CAPÍTULO I | 15 |
| 1.1 Introducción..... | 15 |
| 1.2 Planteamiento del problema | 16 |
| 1.3 Justificación..... | 17 |
| 1.4 Objetivos..... | 18 |
| 1.4.1 Objetivo General..... | 18 |
| 1.4.2 Objetivos Específicos | 18 |
| CAPÍTULO II..... | 19 |
| MARCO TEÓRICO..... | 19 |
| 2.1 Estado del arte | 19 |
| 2.2 Marco teórico..... | 27 |
| 2.2.1 Definición de LTE | 27 |
| 2.2.2 Arquitectura de una red LTE | 28 |
| 2.2.3 Bandas de frecuencia en redes LTE | 29 |
| 2.2.4 Handover en redes LTE | 30 |
| 2.2.5 Tipos de Handover | 31 |
| 2.2.6 Parámetros de configuración del handover | 32 |
| 2.2.7 Parámetros indicadores para la optimización de redes de comunicación.... | 33 |

| | | |
|-------------------------------------|--|-----------|
| 2.2.8 | Gestión de redes celulares mediante algoritmos de aprendizaje automático | 36 |
| 2.2.9 | Clasificación de las técnicas de aprendizaje automático | 37 |
| CAPÍTULO III | | 40 |
| METODOLOGIA..... | | 40 |
| 3.1 | Tipo y diseño de la investigación | 40 |
| 3.2 | Técnicas de recolección de datos..... | 40 |
| 3.3 | Población de estudio y muestra de tamaño..... | 40 |
| 3.3.1 | Población | 40 |
| 3.3.2 | Muestra | 40 |
| 3.4 | Hipótesis | 41 |
| 3.5 | Operacionalización de las variables | 41 |
| 3.6 | Métodos de análisis y procesamiento de datos..... | 41 |
| 3.7 | Diagrama de procesos..... | 42 |
| CAPÍTULO IV..... | | 44 |
| RESULTADOS Y DISCUSIÓN | | 44 |
| 4.1 | Modelos de propagación..... | 44 |
| 4.1.1 | Modelo Rural Macro (RMa)..... | 44 |
| 4.1.2 | Modelo Urban Macro (UMa) | 44 |
| 4.1.3 | Modelo Urban micro (UMi) | 44 |
| 4.2 | Arquitectura del sistema de red 4G LTE simulado | 44 |
| 4.2.1 | Interfaces del sistema..... | 45 |
| 4.3 | Entorno de simulación | 45 |
| 4.3.1 | Software..... | 45 |
| 4.3.2 | Interfaz Gráfica..... | 45 |
| 4.4 | Generación y configuración del escenario 4G LTE | 46 |
| 4.4.1 | Asignación de recursos y topología de la red..... | 46 |
| 4.4.2 | Generación de usuarios..... | 48 |
| 4.4.3 | Posicionamiento y asignación de servicios a los usuarios..... | 48 |
| 4.4.4 | Configuración del modelo del canal | 49 |

| | | |
|--|---|-----------|
| 4.4.5 | Movilidad de usuarios | 49 |
| 4.4.6 | Inicialización de la tabla Q | 50 |
| 4.5 | Modelo matemático del sistema de red 4G LTE | 50 |
| 4.5.1 | Pérdida de trayecto | 50 |
| 4.5.2 | Potencia recibida..... | 51 |
| 4.5.3 | Cálculo de SINR..... | 52 |
| 4.5.4 | Cálculo del throughput | 53 |
| 4.6 | Gestión del proceso de handover..... | 54 |
| 4.6.1 | Criterio de cambio de celda basado en la calidad de la señal..... | 54 |
| 4.6.2 | Evaluación de los umbrales de carga de la celda destino | 54 |
| 4.6.3 | Adaptación de parámetros de handover según el tipo de entorno | 55 |
| 4.7 | Algoritmo de aprendizaje por refuerzo..... | 55 |
| 4.7.1 | Descripción general del proceso del algoritmo Q-Learning..... | 55 |
| 4.7.2 | Hiperparámetros de aprendizaje | 61 |
| 4.8 | Resultados..... | 62 |
| 4.8.1 | Análisis de los resultados por escenario | 62 |
| 4.8.2 | Análisis gráfico de cada escenario | 64 |
| 4.8.3 | Análisis comparativo entre escenarios | 71 |
| 4.8.4 | Análisis de la distribución del throughput..... | 71 |
| 4.8.5 | Prueba de hipótesis | 72 |
| 4.9 | Discusión | 77 |
| CAPÍTULO V | | 78 |
| CONCLUSIONES Y RECOMENDACIONES | | 78 |
| 5.1 | Conclusiones..... | 78 |
| 5.2 | Recomendaciones | 79 |
| BIBLIOGRAFÍA | | 80 |
| ANEXOS | | 84 |
| Anexo 1. Códigos del sistema | | 84 |
| Anexo 2. Parámetros generales del entorno de simulación de la red 4G LTE | | 89 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Resumen de los estudios relevantes sobre el proceso de handover y la implementación de técnicas y algoritmo en redes celulares..... | 24 |
| Tabla 2. Bandas de frecuencia para LTE [38]. | 30 |
| Tabla 3. Comparación de las métricas en redes LTE [21]..... | 35 |
| Tabla 4. Número de PRBs disponibles según el ancho de banda [46]. | 36 |
| Tabla 5. Operacionalización de las variables. | 41 |
| Tabla 6. Parámetros asignados según el número de celdas de la red. | 47 |
| Tabla 7. Parámetros utilizados para el cálculo de la potencia recibida. | 52 |
| Tabla 8. Parámetros de handover y Q-Learning. | 55 |
| Tabla 9. Configuración de hiperparámetros en el algoritmo Q-Learning. | 62 |
| Tabla 10. Resumen de las métricas de rendimiento para el escenario de 5 celdas..... | 63 |
| Tabla 11. Resumen de las métricas de rendimiento para el escenario de 10 celdas..... | 63 |
| Tabla 12. Resumen de las métricas de rendimiento para el escenario de 15 celdas..... | 64 |
| Tabla 13. Comparación de métricas promedio por escenario..... | 71 |
| Tabla 14. Resultados de la prueba de normalidad. | 73 |
| Tabla 15. Análisis T de Student para muestras pareadas..... | 73 |
| Tabla 16. Resultados del ANOVA aplicado a los datos del throughput. | 74 |
| Tabla 17. Comparación de la mejora del throughput entre escenarios mediante la prueba de Tukey HSD..... | 74 |
| Tabla 18. Resultados de la prueba de normalidad para el análisis del índice de Jain. | 75 |
| Tabla 19. Análisis T de Student para muestras pareadas y prueba no paramétrica de Wilcoxon. | 76 |
| Tabla 20. Parámetros generales de simulación..... | 89 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Arquitectura de red LTE. | 28 |
| Figura 2. Comparación del proceso de Hard Handover y Soft Handover [39]. | 32 |
| Figura 3. Bloque de recursos físicos. | 36 |
| Figura 4. Proceso de actualización de la política en el algoritmo Q-Learning [49]. | 38 |
| Figura 5. Fases de trabajo. | 42 |
| Figura 6. Interfaz gráfica del simulador LTE desarrollado en App Designer. | 46 |
| Figura 7. Distribución de celdas para los distintos escenarios simulados: (a) red con 5 celdas (RMa), (b) red con 10 celdas (UMa), (c) red con 15 celdas (UMi). | 48 |
| Figura 8. Flujo de ejecución del algoritmo Q-Learning aplicado al proceso de handover. | 56 |
| Figura 9. Visualización del sistema de simulación: (a) Topología de red con 5 celdas (RMa). (b) Generación y distribución de usuarios. (c) Ejecución del handover y trayectoria de los usuarios transferidos. (d) Comparación del throughput antes y después de aplicar el algoritmo. | 66 |
| Figura 10. Visualización del sistema de simulación: (a) Topología de red con 10 celdas (UMa). (b) Generación y distribución de usuarios. (c) Ejecución del handover y trayectoria de los usuarios transferidos. (d) Comparación del throughput antes y después de aplicar el algoritmo. | 68 |
| Figura 11. Visualización del sistema de simulación: (a) Topología de red con 15 celdas (UMi). (b) Generación y distribución de usuarios. (c) Ejecución del handover y trayectoria de los usuarios transferidos. (d) Comparación del throughput antes y después de aplicar el algoritmo. | 70 |
| Figura 12. Distribución del throughput antes y después de aplicar el algoritmo de handover en escenarios con 5, 10 y 15 celdas | 72 |
| Figura 13. Mejora del throughput por escenario. | 75 |
| Figura 14. Comparación del antes y después del índice de Jain. | 76 |

RESUMEN

En el presente trabajo, se diseñó un modelo de red 4G LTE y se implementó un mecanismo de handover horizontal basado en el algoritmo de aprendizaje por refuerzo Q-Learning. Su objetivo es mejorar la Calidad de Servicio (QoS) y el balanceo de carga entre celdas. Uno de los principales problemas que enfrentan las redes móviles es la sobrecarga de usuarios, lo que provoca una degradación del rendimiento de la red y afecta de forma directa a la experiencia del usuario. En este trabajo, se propuso una solución que integra un algoritmo inteligente que aprende de su entorno y realiza el proceso de handover hacia celdas vecinas con mejores condiciones. Por medio del simulador Matlab se desarrolló el modelo de red para tres distintos escenarios, cada uno con 5, 10 y 15 celdas, una frecuencia de operación de 2100 MHz y la asignación de servicios a cada usuario de voz, videos o datos. La implementación del algoritmo se realizó a partir de la discretización de los valores del SINR y la carga para definir las acciones posibles a realizarse. Esta selección se rige por medio de la política ϵ -greedy, que alterna entre la explotación de la mejor acción conocida y la exploración de nuevas opciones. La recompensa se calcula a partir de la combinación de tres parámetros: *throughput*, balanceo global y la equidad entre las celdas, medido por el índice de Jain. En total, se ejecutaron 180 simulaciones, 60 por cada escenario, registrando sus parámetros antes y después de aplicar el algoritmo. Como resultado se obtuvo una mejora del throughput del 23.92% para el escenario de 5 celdas, 31.03% para el escenario de 10 celdas y el 40.86% para el escenario de 15 celdas. Además, el algoritmo logró mantener una distribución de carga uniforme entre las celdas de los tres escenarios, con un índice de Jain igual o superior a 0.90 y una convergencia estable del proceso de aprendizaje. Los resultados obtenidos demostraron que el algoritmo Q-Learning incrementa el throughput y logra una distribución de carga más equitativa entre las celdas, inclusive en escenarios con alta densidad.

Palabras claves: LTE, handover, aprendizaje por refuerzo, Q-Learning, balanceo de carga, calidad de servicio.

ABSTRACT

In this work, a 4G LTE network model was designed, and a horizontal handover mechanism based on the Q-Learning reinforcement learning algorithm was implemented. It aims to improve Quality of Service (quality of service) and load balancing between cells. One of the main problems in mobile networks is cell overload, which degrades network performance and directly affects the user experience. In this work, a solution was proposed that integrates an intelligent algorithm that learns from its environment and performs handover to neighboring cells with better conditions. Using the MATLAB simulator, the network model was developed for three scenarios: 5, 10, and 15 cells, an operating frequency of 2100 MHz, and the assignment of voice, video, or data services to each user. The algorithm was implemented by discretizing the SINR and load values to define the possible actions. This is governed by the ϵ -greedy policy, which alternates between exploiting the best-known action and exploring new options. The reward is calculated from three parameters: throughput, overall balance, and fairness between cells, measured by the Jain index. In total, 180 simulations were run — 60 for each scenario — recording their parameters before and after applying the algorithm. The result was a 23.92% improvement in the 5-cell scenario, 31.03% in the 10-cell scenario, and 40.86% in the 15-cell scenario. Furthermore, the algorithm maintained even load distribution across cells in all three scenarios, with a Jain index of 0.90 or higher and stable convergence of the learning process. The results demonstrated that the Q-Learning algorithm increases performance and achieves a more equitable load distribution across cells, even in high-density scenarios.

Keywords: LTE, handover, reinforcement learning, Q-Learning, load balancing, quality of service.



Reviewed by:

Mgs. Sonia Granizo Lara.

ENGLISH PROFESSOR.

c.c. 0602088890

CAPÍTULO I

1.1 Introducción

El crecimiento de las comunicaciones móviles ha generado la necesidad de tecnologías capaces de soportar servicios y aplicaciones que demandan altas velocidades de transmisión, baja latencia y una buena Calidad de Servicio (QoS). Actualmente, la telefonía móvil se logra gracias a la conexión entre centrales móviles y públicas [1], facilitando no solo la comunicación de voz, sino también el acceso a la transmisión de datos y aplicaciones basadas en la nube [2]. En el Ecuador, la red de telefonía celular 4G LTE (*Long Term Evolution*) es la más empleada por parte de las operadoras de servicio del país [3]. Según la información obtenida del Ministerio de Telecomunicaciones y de la Sociedad de la Información [4], la cobertura poblacional con tecnología 4G registró un incremento entre el año 2022 y 2023, pasando del 77.84% al 78.49%. Debido al incremento de dispositivos que existen en el mercado, aumenta la demanda de conexiones más rápidas.

Uno de los conceptos importantes aplicados en los sistemas de redes móviles es la delimitación del territorio en áreas más pequeñas denominadas como celdas [5]. Cada celda permite que múltiples usuarios se conecten simultáneamente a la estación base más cercana. Cuando la intensidad de la señal de un dispositivo disminuye por debajo del mínimo y no es suficiente para mantener la conexión con una estación base, necesita conectarse a otra de forma automática [5], [6]. A este proceso se lo denomina Handover (HO) y su mecanismo de transferencia se compone de cuatro etapas: inicio de la transferencia, toma de decisiones, selección de red e implementación del proceso de transferencia [7]. Además, se clasifica en dos tipos, Handover Vertical (VHO), que se caracteriza porque las celdas son de tecnologías diferentes y Handover Horizontal (HHO), en el que los traspasos entre celdas se producen bajo la misma tecnología [8].

Para optimizar el proceso de handover es importante considerar la problemática que enfrentan las redes móviles con respecto a de sobrecarga de usuarios en las celdas. Este problema contribuye a la congestión de la red y a la conexión inestable que presentan los dispositivos. Tomando como referencia el principio del proceso de handover, el cual plantea que el cambio de celda que realice un dispositivo debe ser imperceptible por el usuario [9], se hace uso de sistemas de balanceo de carga, con el objetivo de distribuir el tráfico de red de forma equitativa entre las celdas, logrando así gestionar de forma eficiente volúmenes altos de tráfico [10]. Los avances en el campo del aprendizaje automático y el uso de algoritmos predictivos permiten desarrollar sistemas más eficientes que logren aprender de datos y adaptar su comportamiento a condiciones cambiantes de la red por medio de un proceso de entrenamiento [9], [11]. Bajo este contexto, en el presente trabajo de investigación se pretende analizar el balanceo de carga en una red 4G LTE e implementar un algoritmo de handover horizontal basado en aprendizaje por refuerzo, con el fin de gestionar la transferencia de usuarios entre celdas y asegurar una mejora en la Calidad de Servicio (QoS).

Para evaluar la efectividad del algoritmo propuesto, se inició con una revisión del estado de arte en distintas fuentes académicas, identificando las técnicas y algoritmos más relevantes utilizados para el balanceo de carga y la mejora del throughput dentro del contexto de las redes móviles. De esta manera, se estableció las bases teóricas para el diseño del sistema propuesto. Posteriormente, se diseñará 3 escenarios de red 4G LTE con diferentes números de celdas y se configurarán parámetros iniciales como el ancho de banda, la potencia de transmisión, la ganancia de la antena y el modelo de propagación, según el tipo de entorno. A cada celda se le asignará usuarios con diferentes tipos de servicio, lo que permitirá simular un sistema bajo diferentes cargas de tráfico. El entorno de simulación será desarrollado en el software Matlab, utilizando la herramienta App Designer para visualizar las gráficas correspondientes a la distribución de usuarios y a las trayectorias realizadas desde la estación base origen a la estación base destino después de realizar el proceso de handover aplicando el algoritmo Q-Learning. Para evaluar el desempeño del algoritmo se llevará a cabo diferentes simulaciones en cada escenario y se obtendrán parámetros del throughput, el número de handovers realizados y la distribución de cargas entre las celdas.

1.2 Planteamiento del problema

La tecnología de Cuarta Generación (4G) ha logrado satisfacer las necesidades de conectividad de los usuarios ofreciendo servicios de mayor calidad [12]. Las ventajas que presenta esta tecnología ha impulsado a un aumento en la penetración de líneas activas, se estima que hasta noviembre del 2024 se registraron 18.39 millones de líneas activas en el país [13], de las cuales 11.86 millones de líneas pertenecen a la tecnología LTE [14], según los datos publicados por la Agencia de Regulación y Control de las Telecomunicaciones. Considerando que Ecuador tiene 17.89 millones de habitantes el indicador de densidad de líneas activas alcanza el 102.82% [13] por lo que se establece que existen más líneas activas que población.

Con la expansión de la cobertura 4G, la penetración de líneas activas ha ido aumentado de forma significativa. Sin embargo, el aumento de dispositivos conectados ha afectado la capacidad de la infraestructura de las redes móviles, en ciertas ocasiones el volumen de datos y la cantidad de usuarios conectados superan la capacidad de la celda, lo que provoca que algunas de ellas se sobrecarguen de usuarios. Esta sobrecarga produce desconexiones inesperadas en los dispositivos [9], lo cual resulta perjudicial especialmente en áreas con alta densidad de usuarios o entornos urbanos. En estos escenarios, servicios como las transacciones en tiempo real o ciertas aplicaciones no logran establecer una conexión estable que pueda satisfacer necesidades del usuario, generando retrasos que afectan a la Calidad de Servicio (QoS). A pesar de que existen estrategias para gestionar eficazmente las redes móviles, algunas de ellas presentan limitaciones al abordar el desequilibrio de carga y la mejora de la experiencia de los usuarios de forma simultánea [15]. Generalmente, se pueden adaptar, diseñar u optimizar algoritmos que se encarguen de realizar el proceso de handover de forma automática.

1.3 Justificación

En las redes móviles actuales el handover es indispensable para la transferencia de usuarios entre las celdas [16]. En algunos casos, donde la cantidad de usuarios supera a capacidad de las celdas se puede generar una pérdida de la calidad de la señal o afectar a la estabilidad de los recursos de la red. Al hacer uso de los sistemas de balanceo de carga se reduce de manera significativa la congestión de la red, se acelera los tiempos de respuesta y se minimizan las interrupciones en la conexión de los dispositivos, especialmente en áreas de alta densidad de usuarios, entornos urbanos e inclusive áreas rurales. Los algoritmos basados en aprendizaje por refuerzo son fundamentales para abordar desafíos en el balanceo de carga [17], ya que en base a ciertos criterios redistribuyen el tráfico de usuarios entre las celdas más cercanas permitiendo establecer conexiones más estables. Por este motivo, se propone la implementación de un algoritmo de handover horizontal basado en Q-learning que permita realizar un balanceo de carga eficiente entre las celdas de una red 4G LTE, mediante la metodología propuesta, la disponibilidad de herramientas para el diseño y fundamentación teórica basada en las fuentes académicas revisadas se garantiza que los objetivos planteados sean alcanzados. En este trabajo, se busca establecer un mecanismo que analice las condiciones de la red y se evalúe el comportamiento del sistema mediante simulaciones donde varíe el número de usuarios y celdas con el objetivo de que el algoritmo aprenda, se adapte a las condiciones cambiantes de su entorno y logre tomar decisiones con respecto al equilibrio de carga y la mejora del throughput [9].

Este proyecto de investigación beneficia principalmente a los proveedores de servicios de telecomunicaciones móviles quienes, mediante el uso de algoritmos basados en aprendizaje automático, puedan minimizar la congestión de dispositivos en una celda y gestionar sus recursos aprovechando la infraestructura existente, logrando así una mejor experiencia en los usuarios. Adicionalmente, puede resultar beneficioso para las instituciones que requieran una conectividad constante y estable para sus operaciones diarias. Siendo un aporte para futuras investigaciones orientadas a la gestión de los recursos de las redes móviles LTE y otras tecnologías.

1.4 Objetivos

1.4.1 Objetivo General

Analizar el balanceo de carga en una red 4G LTE e implementar un algoritmo de handover horizontal basado en aprendizaje por refuerzo para mejorar la calidad de servicio (QoS).

1.4.2 Objetivos Específicos

- Revisar el estado de arte mediante una búsqueda exhaustiva en bases de datos científicas y publicaciones relevantes para la identificación de las técnicas más utilizadas y las tendencias actuales en el campo del balanceo de carga en redes LTE.
- Diseñar un modelo de simulación para el proceso de handover horizontal de una red 4G LTE en Matlab, considerando diferentes condiciones de carga y variaciones en el número de usuarios y celdas.
- Implementar un algoritmo de handover horizontal basado en aprendizaje por refuerzo, orientado al balanceo de carga mediante el traspaso de usuarios entre celdas.
- Evaluar el impacto del algoritmo de handover horizontal basado en aprendizaje por refuerzo en la calidad de servicio (QoS) en diferentes configuraciones de red 4G LTE.

CAPÍTULO II

MARCO TEÓRICO

2.1 Estado del arte

Diversos estudios han explorado diferentes enfoques con resultados positivos en la optimización de la distribución de tráfico y el uso de algoritmos predictivos para el proceso de handover. Un caso en particular es el desarrollado en 2020 por Hatipoglu et al., en el cual se propone un algoritmo que logre equilibrar las cargas entre las estaciones base vecinas adaptando automáticamente el Margen de Handover (HOM) y el Time To Trigger (TTT) en función a la velocidad de usuario y la potencia de la señal recibida. La estructura del algoritmo se divide en dos etapas, asignar valores de HOM y TTT según la velocidad del usuario y evaluar el nivel de carga de las estaciones base destino antes de ejecutar el handover. Los resultados mostraron que el algoritmo logra reducir el índice de Handover Fallido (HOF), mejorar el balance de carga entre las estaciones base, logrando que la diferencia máxima de carga entre las celdas pasara del 76% al 39% y reducir la desviación de carga de 0.19 a 0.07, mostrando una red más balanceada [18].

Un nuevo enfoque basado en el Balanceo de Carga por Movilidad (MLB) es el propuesto por Ramírez et al. por medio de un modelo matemático, presenta el comportamiento de las celdas LTE con diferentes condiciones de tráfico e implementa tres estrategias. Para la primera, los usuarios no realizan ningún proceso de handover y se toman las métricas resultantes, la segunda estrategia realiza el proceso de handover entre celdas vecinas solo cuando existe sobrecarga y para la tercera estrategia el handover se realiza a celdas no adyacentes dentro del mismo clúster. A partir de ello, se realizan simulaciones con diferentes tasas de llegada, demostrando una mejora en el throughput global de la red. Para el escenario de mayor congestión se obtuvo el valor de la tasa de pérdida de paquetes sin aplicar ninguna estrategia de balanceo del 45%, con el balanceo de primer orden se obtiene el 37% y con el balanceo de segundo orden el 35%. El throughput promedio alcanza un valor de 5.48 Mbps en el balanceo de segundo orden, siendo mayor en comparación con las otras dos estrategias. Se incrementa, además, el número de handovers hasta el 33% provocando el aumento del tráfico de señalización de la red [19].

Marí et al. presenta un algoritmo de balanceo de carga haciendo uso de la técnica de lógica difusa, adaptando de forma dinámica los valores de los márgenes de traspaso entre las celdas. El segundo algoritmo propuesto se basa en reglas heurísticas que están diseñadas para equilibrar la distribución del tráfico entre las celdas cercanas. Finalmente, propone el uso de un algoritmo de Optimización de la Robustez de la Movilidad (Mobility Robustness Optimization, MRO) basado en redes neuronales y Q-Learning para ajustar automáticamente los parámetros de handover, mejorar la robustez de la red y aumentar la tasa de traspasos exitosos. Como resultado se obtuvo que el desequilibrio en la cantidad de usuarios por celda se reduce de 15.2% a 6.3%. A su vez, el desequilibrio del throughput entre celdas se redujo hasta 0.28 Mbps. En cuanto a la mejora en la equidad de experiencia entre los diferentes servicios dentro de una celda se obtuvo 0.32 y a nivel global de la red 0.36, logrando una

mejor experiencia percibida por los usuarios sin necesidad de modificar la infraestructura de la red [15].

En el año 2023, Espinosa et al. realizan un estudio en la ciudad de Quito acerca del desempeño del proceso de handover en una red 4G LTE en tres escenarios urbanos. Para el análisis, los autores recolectaron datos de una de las operadoras móviles locales, obteniendo muestras por cada escenario, considerando el desplazamiento tanto peatonal como vehicular. Con todos los datos recolectados, clasificaron cada handover realizado en intra-celda e inter-celda y evaluaron distintas métricas de desempeño como Indicador de Intensidad de Señal Recibida (RSSI), Calidad de Señal Recibida de Referencia (RSRQ) y márgenes de potencia. Los resultados mostraron que el 50% de handovers no se realizaban hacia la estación base más cercana, siendo la causa principal de fallos y de la baja calidad de señal percibida por los usuarios. El RSSI para handover inter-celda disminuye entre -99 y -107 dBm, para intra-celda entre -81 y -89 dBm [20].

En la investigación realizada por Martínez et al. se aplica la técnica de lógica difusa basada en un controlador de tipo Mamdani compuesto por 4 etapas: fuzzificación, base de reglas, motor de inferencia y defuzzificador. Los autores proponen que, al aplicar la técnica de lógica difusa, se logren establecer reglas que determinen cuándo y hacia qué celda debe realizar un usuario el handover. Para explorar el rendimiento de la red en escenarios urbanos, se realizaron simulaciones en el software Matlab, donde se varía en número de usuarios, el tráfico y el tamaño del buffer. Los resultados obtenidos mostraron una mejora del 4% del throughput global, una reducción de la tasa de pérdida de paquetes entre el 12% y el 14%, en comparación con escenarios donde no se aplica ninguna técnica de handover y la reducción de handovers innecesarios. A pesar de que la investigación obtuvo resultados positivos, recomienda explorar y validar otros algoritmos [21].

Haciendo referencia a la recomendación del trabajo anterior, Párraga et al. realizan un análisis de las propiedades, características y de los parámetros RF que se involucran para realizar el proceso de handover dentro de dos escenarios distintos a través de un modelo predictivo. El primer escenario consiste en analizar el handover de una zona urbana, mientras que, el segundo escenario busca analizar los errores que se producen durante el proceso de handover en una zona rural. En el primer escenario determinaron que, para las zonas de cobertura normal el Indicador de Intensidad de Señal Recibida (RSSI) toma valores superiores a -87 dBm, mientras que, para zonas donde hay dos o más estaciones base vecinas el RSSI es menor a -98 dBm, generando una alta probabilidad de handover. En el escenario 2, detectaron que las zonas propensas a fallos en el handover tienen un RSSI menor a -107 dBm y un valor de la Calidad de Señal Recibida de Referencia (RSRQ) menor a -17 dBm [22].

Lupera et al. presentan una revisión sistemática del estado de arte sobre la aplicación de técnicas de Machine Learning (ML) en el proceso de handover para redes LTE y 5G usando la metodología de Estudio de Mapeo Sistemático (SMS). Se analizan varios trabajos

enfocados en la mejora de la calidad de servicio mediante algoritmos inteligentes y se destaca que muchos de los trabajos revisados mejoran aspectos aislados de la red. Por este motivo se menciona la importancia de estudiar y evaluar los modelos y algoritmos desarrollados, considerando que su elección debe lograr una optimización lo más general posible [23].

Algunos estudios emplean el uso de algoritmos de aprendizaje por refuerzo, como es el caso de Parreño et al., que proponen un agente Q-Learning que cada vez que detecte desequilibrios de carga entre estaciones base aprenda a modificar el margen de handover. sus resultados mostraron que, para entornos con tráfico generado de voz y datos, un algoritmo de aprendizaje por refuerzo les permite minimizar la congestión y balancear la carga entre celdas vecinas. Cuanto mayor es el paso de variación en el margen de handover, mayor es la recompensa obtenida durante el entrenamiento. En futuras investigaciones recomiendan reducir el efecto ping-pong, debido a que esto afecta a la experiencia del usuario produciendo handovers innecesarios [24].

La propuesta de Yajnanarayana et al. abre paso a escenarios más complejos y el uso de redes 5G. En esta investigación se maneja el proceso de handover en redes densas como un Problema de Bandido Contextual Multi-Brazo (CMAB), donde, empleando un algoritmo de Q-Learning y por medio de datos que los terminales envían a su estación base, se decide la celda óptima que el usuario debería realizar el handover. El objetivo es aumentar el RSRP relacionado con el throughput después de haber realizado el handover. Se comparó el desempeño del algoritmo en dos entornos de simulación y uno real y obtuvieron una mejora del RSRP del link-bean entre 0.3 y 0.5 dB [25].

Mollet et al. diseñan un modelo de algoritmo basado en Double Deep Reinforcement Learning (DDRL) para el proceso de handover en redes de ondas milimétricas (mm-Wave) 5G. El modelo de algoritmo propuesto usa dos redes neuronales gemelas para entrenar un agente inteligente y que sea capaz de aprender de políticas óptimas considerando la calidad del enlace. Para evaluar el modelo, simulan un entorno urbano incorporando trayectorias móviles con múltiples obstáculos y obtienen una reducción de handovers entre el 20% y el 69% menos que el método Rate Based Handover (RBH) y entre el 7% y el 49% menos que el método SMART. En función del throughput, obtuvieron un aumento global entre un 19% y un 40% en comparación del método RBH y un 24% y un 37% en comparación con el método SMART [26].

Un trabajo similar es el presentado por Delin et al., donde para optimizar las decisiones de handover en redes heterogéneas se establece un esquema de gestión de traspasos. Para ello se desarrolla un algoritmo Multi-Agent Proximal Policy Optimization (MAPPO) que entrena de forma centralizada y descentralizada, aprovechando la información global, decide a qué estación base debe conectarse cada usuario. Los resultados mostraron que este método logra un mejor rendimiento en comparación con otros métodos más tradicionales. Durante las iteraciones de entrenamiento, el throughput se mantiene estable, el número de handovers se reduce de forma significativa y el algoritmo tiene una mejor convergencia en entornos más

densos [27]. Ambas investigaciones [26], [27] mencionan que al aplicar los algoritmos propuestos incrementa la complejidad computacional.

Otro algoritmo para optimizar redes ultradensas es el Deep Q-Learning (DQN), presentado por Tianqi et al. En esta investigación se combina el DQN con un gemelo digital de la red para obtener una convergencia más rápida. El gemelo digital basado en una red de memoria a corto-largo plazo (LSTM) consta de tres etapas, como lo es la recopilación de datos, la predicción de la red, la estimación de fallos en el traspaso y en la tasa de efectos ping-pong. Los autores señalan que las simulaciones realizadas muestran una mejora del 2.7% en la tasa de handovers efectivos, más que el algoritmo DQN tradicional y hasta un 10.9% más que otras estrategias. En entornos con fading alto se alcanza una mejora hasta del 16% [28].

Nayak et al., por su parte, explora el uso del simulador de ns-3 para ejecutar múltiples simulaciones y a partir de ello buscar relaciones causales que vinculen las fallas de handover y la elección de parámetros de control. Durante las simulaciones, los parámetros de Hysteresis (Hys), Time To Trigger (TTT) y α 3offset fueron variados sistemáticamente en múltiples combinaciones para observar los efectos en la cantidad de handovers, fallos de enlace (RLFs) y en el throughput global de red. Para encontrar la combinación óptima que muestre un mejor potencial entre los tres parámetros, se usa algoritmos tipo multi-armed bandit y se emplea la regresión de procesos gaussianos como un modelo estadístico de predicción del throughput en función de los parámetros de Handover. Por medio de este modelo estadístico se mostró que los RLFs tienen un impacto negativo en el throughput mayor que los handovers. Además, se calculó el throughput máximo predicho de $1.3073e7$ Kbps con hysteresis aproximado de 3.82 dB y un TTT de 100 ms [29].

Por su parte, Liu et al. desarrollan un mecanismo de handover adaptativo para redes celulares ultra densas con celdas pequeñas. A diferencia de los trabajos anteriores que en su mayoría usaban parámetros establecidos en los estándares de telecomunicaciones, en esta investigación se entrena un agente Q-Learning en función de los parámetros RSRP, SINR y la distancia de transmisión. Este método demostró que el efecto ping pong se reduce de forma significativa, los handovers innecesarios se reducen hasta en un 90% y se mantiene una tasa de handovers fallidos baja. Adicionalmente, se resuelve que un algoritmo Q-Learning puede aprender las políticas óptimas adaptándose a las características del entorno sin la necesidad de establecer los parámetros de Margen de Handover (HOM) y TTT [30].

En una investigación reciente, Caisaguano et al. plantea estudiar a través de simulaciones el desempeño de un algoritmo predictivo mediante redes neuronales para realizar el proceso de handover vertical para dos distintas infraestructuras de comunicación, como son: las redes 4G y satelitales. Para iniciar y entrenar el modelo, se definen umbrales a los parámetros de Jitter, Delay y pérdida de paquetes. Si las métricas exceden los umbrales, se considera necesario realizar el handover. Una vez ejecutadas las simulaciones correspondientes el sistema muestra que es capaz de anticipar la degradación y la señal y posteriormente, ejecutar el handover hacia la red satelital antes de que ocurra una pérdida crítica de calidad. Durante

las simulaciones, la red LTE mostró una latencia de 200 a 300 ms, mientras que la red satelital mostró de 400 a 500 ms. En condiciones estables, la Relación Señal-Ruido (SNR) está entre 32.1 y 39.5 dB y la Tasa de Error de Bits (BER) es menor a 10^{-9} [9].

Los estudios presentados han demostrado la efectividad para mejorar el balanceo de carga, el rendimiento y la optimización de los recursos de la red utilizando diferentes enfoques, técnicas y algoritmos. Estos avances no solo aseguran una distribución de tráfico más equitativo entre las celdas, sino también una mejora en la experiencia de los usuarios. Además, abren camino a futuras investigaciones, sobre todo en el estudio de la combinación de varios algoritmos y lograr que la mejora de la red no solo sea en parámetros aislados.

Después de realizar el análisis del estado de arte, se resalta la importancia de plantear propuestas innovadoras en la gestión de redes móviles para lograr enfrentar los desafíos actuales de conectividad en las redes 4G LTE. Para el desarrollo del modelo de simulación de una red 4G LTE y la implementación de un algoritmo de handover horizontal basado en Q-Learning se tomará como punto de partida la investigación realizada por Marí et al. ya que en su modelo propuesto, demuestra que una recompensa óptima puede equilibrar los recursos de la red ocupados por los usuarios y mejorar la calidad de experiencia (QoE) percibida por medio de la transferencia de handover [15]. Esta idea se ajusta al objetivo de mejorar la calidad de servicio mediante balanceo de carga sin afectar directamente a los servicios asignados a los usuarios. La investigación realizada por Parreño et al. demuestra que un algoritmo de hard handover basado en Q-Learning puede aumentar el throughput bajo diferentes densidades de usuarios. Sin embargo, dicho algoritmo no distingue los tipos de servicio generado por el simulador [24]. Esta investigación será la base de estudio para la implementación del algoritmo de handover horizontal basado en Q-Learning, considerando la variación de celdas, la densidad de usuarios y la asignación de servicios a cada usuario con el objetivo de simular diferente carga de tráfico y un entorno más real. Finalmente, la investigación de Espinosa et al. me permitirá incorporar parámetros en mi investigación para ajustar tanto el algoritmo Q-Learning como los escenarios de simulación, evitando así que sean escenarios idealizados [20].

Tabla 1. Resumen de los estudios relevantes sobre el proceso de handover y la implementación de técnicas y algoritmo en redes celulares.

| Autor/año | Propósito | Metodología | Resultados |
|---|---|--|--|
| J. Ramírez (2020) | Validar un modelo de balanceo de carga en una red LTE utilizando un enfoque basado en Mobility Load Balancing (MLB). | Por medio de un modelo matemático se presenta el comportamiento de las celdas LTE y se aplican tres estrategias: sin balanceo de carga, balanceo de carga de primer orden y balanceo de carga de segundo orden | Para el escenario de mayor congestión, la tasa de pérdida de paquetes sin ninguna estrategia de balanceo es del 45% , con el balanceo de primer orden el 37% y para el balanceo de segundo orden el 35% . |
| S. Hatipoglu, B. Mehmet, M. Akif Yaz, y L. Ata Durak (2020) | Diseñar una estrategia de balanceo de carga para redes 5G adaptando automáticamente los parámetros de handover en función de la velocidad de usuario y las condiciones de la señal. | El algoritmo se estructura en dos etapas: ajuste adaptativo de los parámetros HOM, TTT y explorar el estado de la capacidad de las BS objetivo | Reducción del índice de handover fallido. La diferencia máxima de carga entre las celdas se redujo de 76% a 39% . Reducción de la desviación de carga de 0.19 a 0.017 . |
| V. Yajnanarayana, H. Rydén, y H. László (2020) | Gestionar el proceso de handover en redes 5G densas por medio de la aplicación de un algoritmo de Q-Learning para decidir el handover óptimo. | Modelar el proceso de handover como un problema (CMAB), aplicar el algoritmo Q-Learning para el entrenamiento con exploración aleatoria. | Comparación del algoritmo en tres escenarios, obteniendo una mejora del RSRP del link-bean entre 0.3 y 0.5 dB . |
| M. Mollel <i>et al.</i> (2020) | Diseñar un mecanismo de decisión de handover en redes mmWare mediante Double Deep Reinforcement Learning (DDRL) para minimizar los fallos de enlace. | Modelado de un entorno usando un esquema DDRL para redes 5G ultra densas. Uso de dos redes neuronales para entrenar un agente inteligente. Entorno de simulación urbano. | Aumento global del throughput entre un 19% y 40% en comparación del método RBH y un 24% y 37% en comparación con el método SMART. |
| G. Delin, T. Lan, Z. Xinggan, y L. | Optimizar las decisiones de handover y asignación de potencia de una red heterogénea | Desarrollo de un algoritmo MAPPO que entrena de forma centralizada y | Mejor rendimiento en comparación con otros métodos tradicionales. |

| | | | |
|---|--|--|---|
| Ying-Chang (2020) | mediante un enfoque Multi-Agent Proximal Policy Optimization (MAPPO) | descentralizada, aprovechando la información de la red. | Throughput más estable y una buena convergencia del algoritmo. |
| Q. Liu, C. Foong Kwong, W. Sun, L. Li, y H. Zhao (2020) | Diseñar un mecanismo adaptativo de handover para redes ultra densas usando Q-Learning basado en las mediciones generadas por los usuarios. | Se implementa un agente Q-Learning para decidir el handover sin depender de los parámetros HOM y TTT. | Se reduce el efecto ping pong, los handovers innecesarios bajan hasta un 90% y se mantiene una tasa de handovers fallidos baja. |
| M. Mari (2021) | Diseñar algoritmos que minimicen los fallos de enlace y la mejorar de la calidad de experiencia del usuario (QoE). | Se ajusta los parámetros de handover mediante un algoritmo basado en la técnica de lógica difusa, reglas heurísticas y un algoritmo MRO basado en redes neuronales y Q-Learning. | Reducción del desequilibrio de carga por celda de 15.2% a 6.3% . Reducción del desequilibrio en el throughput hasta 0.28 Mbps y a nivel global la equidad de experiencia mejora 0.36 , siendo una métrica adimensional. |
| C. Parreño, P. Lupera, R. Llusi, y V. Párraga (2021) | Implementar un algoritmo de handover basado en Q-Learning para balancear la carga entre celdas en escenarios sencillos. | Aplicación de un agente basado en Q-Learning dentro de un simulador de red LTE-A con carga variable, se ajusta el margen de handover y se controla la transferencia de usuarios. | Cuanto mayor es el paso de variación en el margen de handover, mayor es la recompensa obtenida durante el entrenamiento. La implementación del algoritmo minimiza la congestión. |
| J. Tianqi, Y. Huiyuan, y X. Zhang (2023) | Diseñar un modelo de optimización de handover basado en Deep Q-Learning (DQN) y un gemelo para redes ultra densas. | El DQN utiliza recompensas de acuerdo con la tasa de handover fallidos, mientras que el gemelo digital predice esas métricas usando datos históricos. | Mejora del 2.7% en la tasa de handovers efectivos, más que el algoritmo DQN tradicional en entornos con fading alto se alcanza una mejora del 16%. |

| | | | | |
|--|----------|---|---|--|
| S. (2023) | Nayak | Registrar y optimizar los parámetros de control de handover en redes LTE para mejorar el rendimiento de la red al reducir el número de fallos de enlace usando el simulador ns-3, y métodos estadísticos. | Se ejecutan múltiples para buscar relaciones caudales que vinculen las fallas de handover y los parámetros de Hys, TTT y a3Offset. Uso de un modelo estadístico para predecir el throughput máximo. | Se demostró que que los RLFs tienen un impacto negativo en el throughput mayor que los handovers. El throughput máximo predicho de 1.3073e7 Kbps con hysteresis aproximado de 3.82 dB y un TTT de 100 ms . |
| R. Espinosa, P. Lupera, V. Farre, R. Maldonado, y R. Llugsi (2023) | | Analizar de forma estadística la tasa de fallos de handover y efectos ping pong en escenarios reales de la ciudad de Quito. | Monitorización de 33 variables de una red móvil en escenarios urbanos. Procesar los datos y clasificarlos según el tipo de handover realizado, inter-celda o intra-celda. | El 50% de handovers no se realizan a la celda más cercana. El RSSI para handover inter-celda disminuye entre un -99 a -107 dBm . Para intra-celda de -81 a -89 dBm . |
| V. Párraga, C. Rocha, H. Navarrete, y P. Lupera (2023) | | Identificar mediante árboles de decisión zonas donde ocurren el handover y zonas más propensas a que se produzcan handovers fallidos en la ciudad de Quito. | Medición de campo en una zona urbana y en una zona rural. Uso de árboles de decisión en RStudio para identificar las zonas más propensas a fallos de handover y zonas con cobertura normal. | Escenario 1: valor de RSSI superior a -87 dBm . Cuando hay dos o más estaciones base vecinas el RSSI es menor a -98 dBm . Escenario 2: zonas propensas a handover presentan un RSSI menor a -107 dBm y un RSRQ menor a -17 dBm . |
| V. Párraga, P. Lupera, F. Grijalva, y H. Carvajal (2023) | | Revisión literaria sobre el uso de técnicas de machine learning aplicados al proceso de handover para identificar tendencia y posibles áreas de estudio. | Uso de la metodología SMS para la revisión de varios artículos sobre la aplicación de machine learning en la automatización de los procesos de handover. | Se destaca la importancia de datos de alta calidad como base para el entrenamiento de modelos ML. Se describe la variedad de algoritmos de machine learning. |
| K. (2024) | Martínez | Evaluar la efectividad de un sistema de | Diseño de un controlador difuso tipo | Mejora del throughput global hasta un 4% . |

| | | | |
|----------------------|--|--|---|
| | balanceo de carga en un escenario de red LTE aplicando la técnica de lógica difusa. | Mamdani, estructurado en 4 etapas: fuzzificación, base de reglas, motor de inferencia y defuzzificación. Simulaciones realizadas en Matlab. | Reducción de la tasa de pérdida de paquetes entre el 12% y el 14% . Reducción de handovers innecesarios. Mejora de la distribución de carga. |
| B. Caisaguano (2025) | Diseñar un algoritmo basado en redes neuronales para realizar un handover vertical entre redes 4G y redes satelitales. | Se desarrolla un algoritmo predictivo de handover horizontal definiendo umbrales para los parámetros Jitter, Delay y pérdida de paquetes. Si las métricas exceden los umbrales se considera necesario realizar el handover | La red LTE mostró una latencia de 200 a 300 ms . La red satelital mostró de 400 a 500 ms . En condiciones estables la Relación Señal-Ruido (SNR) está entre 32.1 y 39.5 dB y la Tasa de Error de Bits (BER) menor a 10⁻⁹ |

2.2 Marco teórico

2.2.1 Definición de LTE

Long Term Evolution (LTE), desarrollado por 3GPP, aparece en el año 2007 como un nuevo estándar de las comunicaciones móviles de Cuarta Generación (4G). Uno de sus principios fundamentales es garantizar la continuidad de la red mediante sistemas más fiables con el fin de satisfacer la necesidad de conexiones más rápidas por parte de los usuarios. Está diseñada para dar soporte a los teléfonos móviles y dispositivos portátiles a la red de internet, ofreciendo servicios como la transmisión de eventos en vivo, subir y descargar información en cuestión de segundos, presentar una alta calidad en las videoconferencias realizadas en tiempo real, entre otros [31].

Presenta algunas características importantes que lo diferencian de las tecnologías anteriores, por ejemplo, LTE ofrece velocidades de descarga de hasta 100 Mbps y de subida de hasta los 30 Mbps. Para las aplicaciones en tiempo real, resulta beneficioso que tenga una latencia baja, que generalmente es menor a 5 ms [32], reduce los costos de infraestructura ya que utiliza la conmutación de paquetes y presenta una alta compatibilidad entre los modos de *Duplexacion Frecuency Division Duplexing* (FDD) y *Time Division Duplexing* (TDD), lo que facilita su implementación a nivel global.

2.2.2 Arquitectura de una red LTE

Un sistema de red LTE incluye componentes diseñados para lograr una arquitectura plana con la menor cantidad de nodos e interfaces posibles [19]. Su arquitectura incluye componentes fundamentales como el *User Equipment* (UE), el cual permite acceder a la red a través del sistema de acceso *Evolved Universal Terrestrial Radio Access Network* (E-UTRAN). Este sistema está compuesto por estaciones base denominadas eNodeB quienes son las encargadas de la asignación de recursos y de las decisiones de transferencia. Se comunican mediante dos identidades, el MME y el S-GW [19]. Para la identificación del usuario la red hace uso de la tarjeta *Universal Integrated Circuit Card* (UICC), mientras que, los servicios de voz y datos los provee la red de acceso *Radio Access Network* (RAN). Además, una red LTE se caracteriza por tener una red troncal conocida como *Envolved Packet Core* (EPC), es la encargado de controlar y administrar el sistema, se divide en dos planos principales: plano de control y plano de usuario [33].

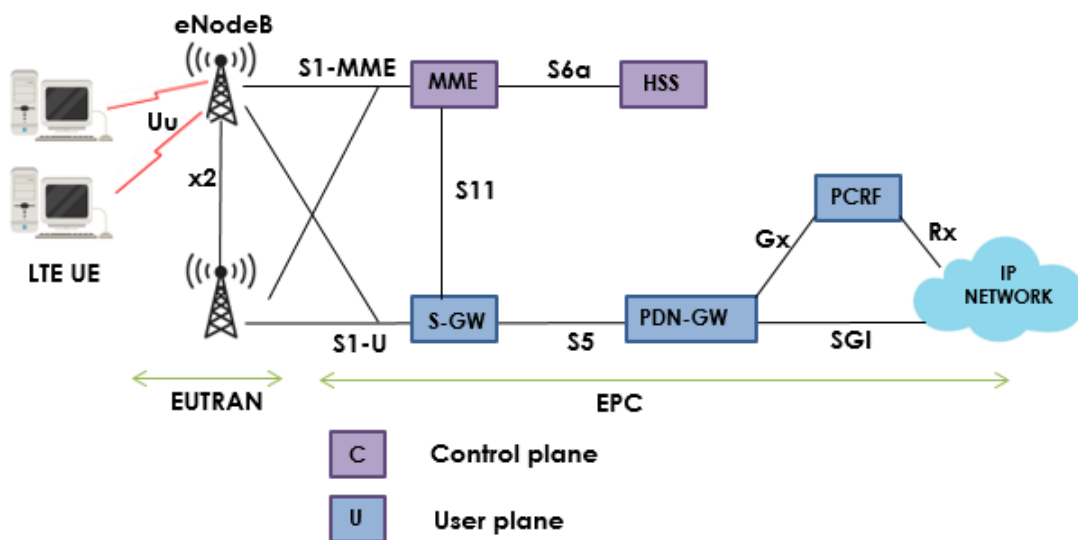


Figura 1. Arquitectura de red LTE.

Plano de control

- **MME (Mobility Management Entity):** Es un componente de la EPC y se encarga de las funciones principales relacionadas con la gestión y el acceso de movilidad de usuarios. A diferencia de otros elementos de la red, MME no transporta paquetes IP con datos de usuario, por lo que permite que sea escalable independientemente de la capacidad de señalización y el tráfico de datos. Entre sus funcionalidades se encuentra la autenticación de los usuarios, la gestión de parámetros asociados a la calidad de servicio (QoS), selección de los P-GW y S-GW, seguimiento de usuarios que se encuentran dentro del área del servicio de la red, entre otros. El proceso de

elección de la entidad MME depende de la posición geográfica del terminal y de los criterios de balanceo de cargas [34].

- **HSS (*Home Subscriber Service*):** Es una base central de datos del sistema 3GPP que contiene información de los usuarios de la red, almacena información como los datos de autenticación y autorización [35], las configuraciones de calidad de servicio y maneja información que le permite gestionar la movilidad de los usuarios. HSS utiliza la interfaz S6 para lograr comunicar estos datos al MME. Toda la información que contiene el HSS puede ser únicamente modificada mediante procesos administrativos. Su arquitectura integra dos funciones esenciales que han sido estandarizadas en 3GPP R5, *Home Location Register* (HLR) y *Authentication Center* (AUC) cada función se encarga de almacenar y actualizar la información de los usuarios y generar la seguridad necesaria para la protección de la comunicación dentro de la red respectivamente [34].

Plano de usuario

- **S-WG (*Serving Gateway*):** Es la puerta de enlace que gestiona un punto de conexión a través de la interfaz S1 desde la EPC hacia la E-UTRAN, proporcionando una entrada inicial para que los usuarios se movilen entre los ENodeBs con el fin de facilitar la continuidad del servicio cuando se produzca el proceso de handover [36]. Debido a que S-WG se adapta a la movilidad, el usuario debe lograr conectarse a cualquier P-WG, ya que este permanece constante en la red. A su vez, S-WG puede conectarse a un MME, por lo que en cada instante en el usuario llegue a una nueva celda, logra conectarse a un ENodeB. De este modo se optimiza la movilidad y la eficiencia en la transmisión de datos en la red.
- **PDN-GW o P-GW (*Packet Data Network Gateway*):** La puerta de enlace de la red de paquetes de datos similar al S-WG es un punto de conexión con las redes IP externas, a través de la interfaz SGI [35]. Entre algunas de sus funciones se encuentra la asignación de direcciones IP a los usuarios y la distribución de tráfico. Puede, además, actuar como un servidor DHCP o comunicarse con un servidor externo. Su funcionamiento se basa en que, cuando un usuario cambia de S-GW, el P-GW reenvía el tráfico al nuevo S-GW conectado. Es importante mencionar que un P-GW puede estar conectado a múltiples S-GW según las necesidades de la red.
- **PCRF (*Policy and Charging Rules Function*):** Es el nodo responsable de la gestión de Calidad de Servicio (QoS), tiene la capacidad de gestionar la política de redes y priorizar el tráfico de red de manera eficiente. Cuando un usuario se conecta a la red, en función DE la solicitud que realice el P-WG y el S-GW, la PCRF ejecuta reglas para la toma de decisiones en función [37].

2.2.3 Bandas de frecuencia en redes LTE

El sistema LTE opera en una variedad de bandas de frecuencia y están definidas por organismos de estandarización como el 3GPP. Estas bandas de frecuencia están distribuidas mediante enlaces ascendentes o descendentes y sus valores varían según la regulación local.

En la **¡Error! No se encuentra el origen de la referencia.**, se describen las bandas de frecuencia LTE según su clasificación, incluyendo información acerca del rango de frecuencia y los canales de ancho de banda que pueden ser asignados. En Ecuador, las redes 4G LTE operan en modo FDD y utilizan las bandas asignadas por la Agencia de Regulación y control de las Telecomunicaciones (ARCOTEL). A las operadoras Claro y CNT se les asigna la banda 4, mientras que la operadora Movistar hace uso de la banda 2. Además, la banda 28 también es usada para ampliar la cobertura en zonas rurales por CNT.

Tabla 2. Bandas de frecuencia para LTE [38].

| Banda de operación | Rango de enlace ascendente (MHz) | Rango de enlace descendente (MHz) | Canales de ancho de banda (MHz) |
|---------------------------|---|--|--|
| 1 | 1920 – 1980 | 2110 – 2170 | 5, 10, 15, 20 |
| 2 | 1850 – 1910 | 1930 – 1990 | 1, 4, 3, 5, 10, 15, 20 |
| 3 | 1710 – 1785 | 1805 – 1880 | 1, 4, 3, 5, 10, 15, 20 |
| 4 | 1710 – 1755 | 2110 – 2155 | 1, 4, 3, 5, 10, 15, 20 |
| 5 | 824 – 849 | 869 – 894 | 1, 4, 3, 5, 10 |
| 7 | 2500 – 2570 | 2620 – 2690 | 5, 10, 15, 20 |
| 8 | 880 – 915 | 925 – 960 | 1, 4, 3, 5, 10 |
| 12 | 699 – 716 | 729 – 746 | 1, 4, 3, 5, 10 |
| 13 | 777 – 787 | 746 – 756 | 1, 4, 3, 5, 10 |
| 14 | 788 – 798 | 758 – 768 | 1, 4, 3, 5, 10 |
| 17 | 704 – 716 | 734 – 746 | 1, 4, 3, 5, 10 |

En el presente trabajo, se ha seleccionado usar la frecuencia de 2100 MHz para la simulación de la red 4G LTE como una aproximación a la banda 4. Este valor ha sido utilizado en otros estudios, lo que refuerza su validez para obtener resultados que estén dentro de la metodología de trabajos similares.

2.2.4 Handover en redes LTE

El proceso de Handover (HO) es un mecanismo de las redes móviles que permite mantener la conexión entre la red y el terminal móvil cuando el usuario se mueve de una estación base a otra [22]. Este proceso se realiza cuando un usuario atraviesa el límite de la cobertura de la celda actual y se ve obligado a conectarse a la celda siguiente. Para ello, el dispositivo analiza los parámetros como la potencia de transmisión de la estación base, la potencia de la señal recibida de referencia y la relación señal-interferencia con el fin de realizar la selección que ofrezca las mejores condiciones [33]. Este proceso puede ocurrir en dos escenarios distintos, como son: el idle mode que ocurre cuando un dispositivo permanece en la red, pero no envía datos de forma activa [22], mientras que en el connected mode los usuarios transmiten los datos de forma continua y sin interrupciones.

La red es la encargada de gestionar el proceso de handover, al mismo tiempo, el dispositivo del usuario realiza las mediciones necesarias para determinar la posición de la celda óptima para realizar su traspaso, una vez que se ejecuta este proceso, la sincronización entre el

dispositivo y la celda debe ser de forma inmediata. En ocasiones, cuando existen un gran número de usuarios en una sola celda y más dispositivos intentan conectarse, se producen sobrecargas de usuarios, lo que genera desconexiones inesperadas de la red e inclusive fallos en la recepción y transmisión de llamadas. Para sobrellevar este problema, se considera necesario realizar un balanceo de carga mediante el proceso de handover a las celdas vecinas que tengan menor congestión de usuarios.

2.2.5 Tipos de Handover

Existen diferentes tipos de handover que se diferencian según su tecnología y los criterios que emplea para el traspaso de una celda a otra. A continuación, se presentan los principales tipos, que, según su base teórica, serán seleccionados para realizar el proyecto de investigación.

Hard handover

En el handover duro o también conocido como *break before make*, se produce una interrupción temporal en la comunicación, ya que el usuario debe desconectarse de la estación base actual antes de conectarse a la nueva estación base. Para que se produzca esta transferencia, la intensidad de la señal de la celda actual debe ser inferior a la intensidad de la señal de la celda vecina [33], [39]. Este tipo de handover es empleado en redes LTE, debido a su arquitectura, minimiza la latencia del proceso. Sin embargo, también presenta desventajas como la pérdida momentánea de la comunicación, especialmente cuando el usuario se mueve de una celda a otra de forma constante.

Soft handover

El handover suave o *make before break* permite que el usuario mantenga la conexión de la celda actual mientras se conecta a la nueva celda con el fin de evitar interrupciones en la comunicación [33]. A pesar de que ofrece una mayor fluidez en el traspaso de una celda a otra, mejorando la experiencia del usuario, se necesita de más equipos adicionales en la red. Este tipo de handover es usado en generaciones anteriores a la tecnología 4G. Donde su principal función es optimizar los servicios, especialmente en áreas donde la intensidad de la señal es débil [39].

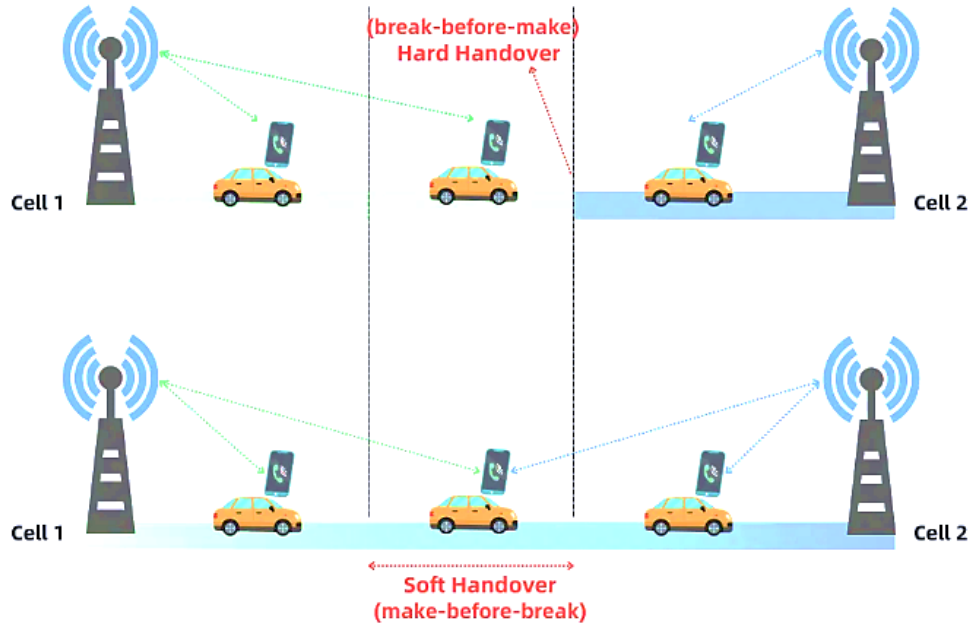


Figura 2. Comparación del proceso de Hard Handover y Soft Handover [39].

Handover horizontal

Se refiere al proceso mediante el cual un usuario móvil trasfiere su conexión de una celda a otra dentro de la misma red y usando la misma tecnología. Este mecanismo es esencial, ya que garantiza la continuidad de servicio durante la movilidad del usuario, permitiendo mantener la conexión activa y sin interrupciones [33].

Handover vertical

Este tipo de handover se realiza cuando el usuario cambia de una tecnología a otra diferente. A pesar de que este proceso resulta ser complejo debido a que requiere mayor cantidad de procesos de señalización. Es fundamental en entornos heterogéneos donde la cobertura o la calidad de servicio puede variar entre tecnologías [40].

2.2.6 Parámetros de configuración del handover

Margen de traspaso (HOM)

Uno de los parámetros claves del proceso de handover es el Margen de Traspaso (HOM), su función principal es definir el umbral a partir del cual un usuario debe iniciar el cambio de celda [15]. Para ello, se consideran las siguientes variables:

- $P_{rx}(i)$ es la potencia de transmisión recibida desde la celda actual.
- $P_{rx}(j)$ es la potencia de la señal recibida de la celda vecina.
- $HOM_{(i,j)}$ es el valor de margen de traspaso entre la celda actual y la celda vecina.

Si la diferencia entre ambas potencias supera el margen de traspaso se cumple la siguiente condición:

$$P_{rx}(j) - P_{rx}(i) \geq HOM(i,j) \quad (5.1)$$

Debido a que ciertas ocasiones puede ocurrir un handover repetitivo entre dos celdas, los márgenes de traspaso se establecen de manera que cumplan la relación:

$$HOM(i, j) + HOM(j, i) = H \quad (5. 2)$$

Donde

- H representa el valor de histéresis del sistema y actúa como un mecanismo de control para estabilizar el proceso de handover [3].

El ajuste dinámico del margen de traspaso es una herramienta efectiva para realizar el balanceo de carga de la red, ya que al aumentar el valor de $HOM_{(i,j)}$ la cobertura de la celda (i) reduce la cantidad de traspasos hacia la celda (j). Por el contrario, al disminuir su valor los traspasos se realizan con mayor facilidad lo que ayuda a distribuir la carga de tráfico y optimizar los recursos de la red.

2.2.7 Parámetros indicadores para la optimización de redes de comunicación

En las redes LTE, los parámetros indicadores de rendimiento son fundamentales para garantizar una buena experiencia en los usuarios. A medida que el aumento de dispositivos continúa la demanda de tráfico de datos también aumenta de forma progresiva. Por este motivo es importante implementar estrategias que nos permitan evaluar el desempeño de la red mediante parámetros que muestren la distribución y gestión de recursos y su impacto en la Calidad de Servicio (QoS). Uno de los principales desafíos que se presentan en las redes LTE es que, si existe una mala gestión de los recursos, la red puede presentar congestión o degradación en la transmisión de datos. Para ello, optimizar estos procesos depende del monitoreo de los recursos mediante parámetros que nos ayudan a identificar problemas de congestión, analizar y aplicar estrategias según el caso que sea necesario.

Tasa de transferencia efectiva (Throughput)

La tasa de transferencia efectiva es una métrica que indica la cantidad de datos que se transmiten con éxito a través de un enlace de comunicación en un tiempo determinado, reflejando el rendimiento real del sistema, ya que se encuentra influenciado por otros factores de la red como la pérdida de paquetes, la latencia, entre otros [41]. Este parámetro está estrechamente relacionado con el balanceo de carga, debido a que al distribuir el tráfico en diferentes celdas se logra mejorar el rendimiento general de la red, permitiendo así evaluar la calidad de servicio (QoS) y observa cómo su valor puede afectar a la experiencia del usuario.

Impacto de la carga en el Throughput

La cantidad de tráfico de un sistema influye directamente en su rendimiento. Cuando la carga de tráfico es baja, los recursos disponibles para procesar los datos no se encuentran saturados, esto permite obtener un alto throughput, lo que asegura que los datos sean transmitidos de

forma eficiente. A medida que la carga de la aumenta, la demanda de recursos crece, lo que conlleva una saturación progresiva. A este problema se lo conoce como sobrecarga de usuarios y sucede cuando varios usuarios intentan acceder a los mismos recursos de la red. Como consecuencia, el throughput disminuye gradualmente, provocando una mayor latencia y retrasos en el envío de datos. Todos estos factores afectan a la red y a la calidad de servicio experimentada por los usuarios, sobre todo en el uso de aplicaciones que requieran conexiones en tiempo real.

Reference Signal Received Power (RSRP)

Es una de las métricas fundamentales de las redes celulares, mide la potencia expresada en dB de las señales recibidas por un dispositivo [42]. Sus valores predefinidos para redes LTE varían entre -140 dB y -44 dB, mientras que para la tecnología 5G sus valores están entre los -156 dB y -31 dB [21]. Dichos valores reflejan la evolución de cada tecnología para crear sistemas más sensibles en la detección de señales.

Dentro del proceso de handover, esta métrica ejecuta un papel fundamental, ya que interfiere de forma directa en la selección de las celdas para el traspaso de usuarios, lo que resulta de gran importancia para la gestión efectiva de movilidad, en este caso dentro de una red LTE. Su medición se realiza cuando la red envía información al usuario a través de la RRC (*Radio Resource Control*), en donde se indica el número de Bloques de Recursos Físicos (PRBs) [43]. Mientras más alto sea el valor de la RSRP, los usuarios experimentan conexiones más estables, mientras que, si su valor es bajo, se presentan problemas de latencia, afectando a la calidad de servicio percibida por los usuarios.

Referenced Signal Strength Indicator (RSSI)

Representa la potencia expresada en dBm total que es detectada por el Equipo de Usuario (UE) [21]. A diferencia del RSRP, el valor de esta métrica incluye el valor del ruido térmico y las interferencias que son generadas por otras fuentes externas. Generalmente, el RSSI se encuentra en un rango de -120 dBm a 0 dBm para redes LTE y entre -120 dBm a -13 dBm para redes 5G. Debido a que su medición abarca la potencia total recibida, no depende directamente de una celda específica y se calcula a partir de la Multiplexación por División de Frecuencias Ortogonales (OFDM). Para las redes celulares, el valor óptimo de RSSI que garantice el funcionamiento estable del sistema debe ser superior a -100 dBm [42].

Reference Signal Received Quality (RSRQ)

Esta métrica evalúa la calidad de la señal recibida en las redes celulares [44], es considerada de mayor importancia cuando se requiere un análisis intra-frecuencia, es decir, cuando una celda destino y una celda vecina operan a la misma frecuencia. Su cálculo relaciona la PSRP y el RSSI con el fin de obtener información adicional sobre los escenarios donde la información del RSRP no es suficiente para realizar la toma de decisiones para el traspaso de celdas en el proceso de handover. Sus valores para redes LTE varían entre -20 dB y -3dB [21]. Con el fin de evitar errores en las decisiones del traspaso de celdas, el RSRQ se evalúa

en un periodo de tiempo mayor que el RSRP, siendo esto una característica clave de esta métrica [43].

Signal to Interference plus Noise Ratio (SINR)

La relación señal a interferencia más ruido es un indicador que permite evaluar la calidad de la señal recibida, considerando tanto el ruido como la interferencia que pueden provocar las celdas vecinas [42]. En redes LTE, comúnmente sus valores se encuentran entre 0 dB y 20 dB, a diferencia de la tecnología 5G, en donde sus valores pueden variar entre los -23 dB hasta los 40 dB [21]. Cuando se tiene un valor alto de SINR, se establece que la señal útil es mayor a las señales de ruido o interferencia. Por lo que esta métrica se considera importante para el análisis de redes celulares, ya que proporciona una visión realista sobre las condiciones del entorno, considerando factores claves que podrían afectar la optimización y eficiencia de la red. Es importante mencionar que, al considerarse una métrica importante, también resulta ser compleja, ya que puede experimentar distintos niveles de interferencias [43].

Tabla 3. Comparación de las métricas en redes LTE [21].

| Condiciones RF | RSRQ (dB) | RSRP (dBm) | SINR (dB) |
|-----------------------|------------------|-------------------|------------------|
| Excelente | ≥ -10 | ≥ -80 | ≥ 20 |
| Bueno | -10 a -15 | -80 a -90 | 13 a 20 |
| Centro de la celda | -15 a -20 | -90 a -100 | 0 a 13 |
| Borde de la celda | ≤ -20 | ≤ -100 | ≤ 0 |

Physical Resource Blocks (PRBs)

Son las unidades mínimas de recursos que se le pueden asignar a un usuario dentro de una red celular [45]. Cada PBR ocupa una frecuencia de 180 KHz y cuenta con 12 subportadoras separadas de 15 KHz cada una. Un solo PRB cuenta con 84 elementos que, dependiendo de la modulación asignada al usuario, puede transportar 2, 4 o 6 bits. Uno de sus principios fundamentales es que facilita la transmisión de datos mediante OFDM para el enlace descendente y SC-FDMA para el enlace ascendente [46].

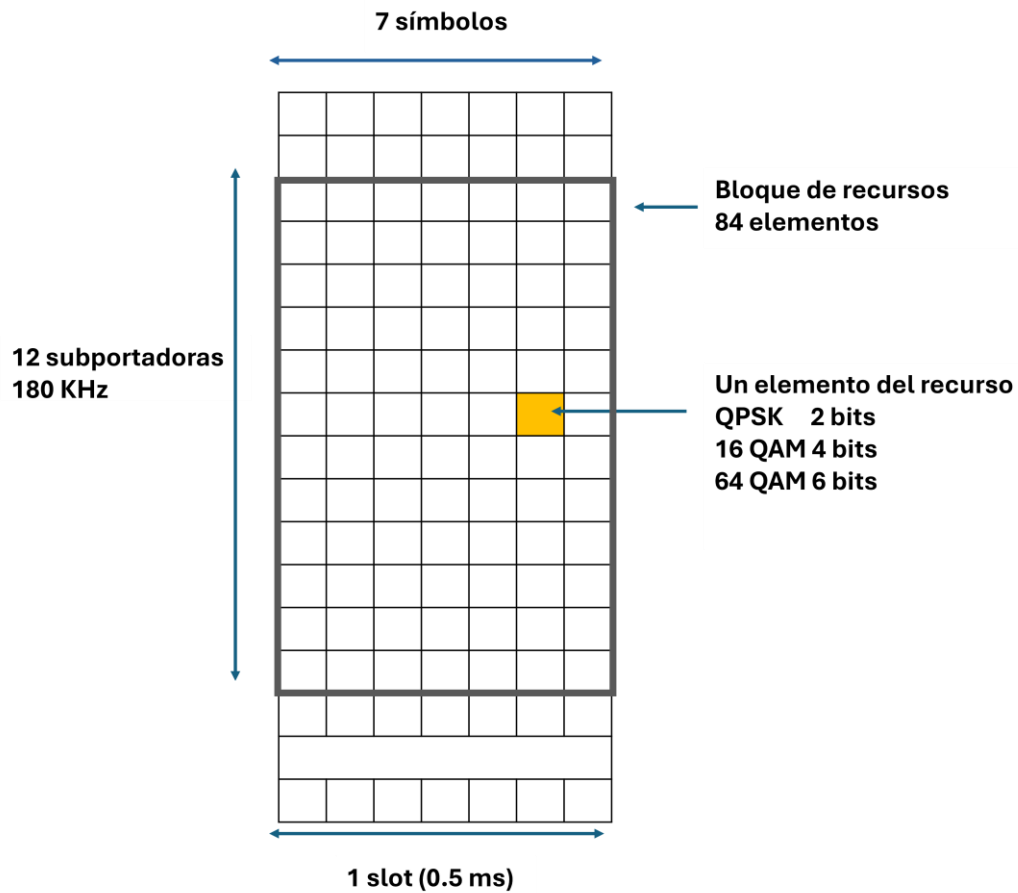


Figura 3. Bloque de recursos físicos.

La cantidad de PRBs que sean asignados al usuario influye directamente en la velocidad de transmisión de los datos. Por ejemplo, las aplicaciones que manejan grandes cantidades de datos, como las usadas por empresas o instituciones suelen experimentar variaciones en la cantidad de PRBs asignados, debido a la congestión de la red, generando así un impacto negativo en la velocidad de descarga y en la calidad de servicio. El ancho de banda de la portadora es una de las características principales para determinar la cantidad de PRBs disponibles en la red, esta relación se presenta en la Tabla 4.

Tabla 4. Número de PRBs disponibles según el ancho de banda [46].

| Ancho de banda | 1.4 MHz | 3 MHz | 5 MHz | 10 MHz | 15 MHz | 20 MHz |
|----------------|---------|-------|-------|--------|--------|--------|
| Nro. de PRB | 6 | 15 | 25 | 50 | 75 | 100 |

El throughput está estrechamente relacionado al número de PRBs disponibles, ya que, a un mayor throughput indica una administración eficiente de los PRBs dando como resultado el desempeño óptimo de la red con una conectividad confiable.

2.2.8 Gestión de redes celulares mediante algoritmos de aprendizaje automático

La aplicación de algoritmos basados en aprendizaje automático actualmente se ha convertido en una de las técnicas más analizadas y utilizadas para lograr mejorar el rendimiento de las

redes celulares [15]. Mediante el uso de redes neuronales, técnicas de lógica difusa, árboles de decisión, entre otros, se ha logrado optimizar y gestionar de forma automática algunos procesos importantes de las comunicaciones móviles, basándose en el objetivo de encontrar soluciones que se adapten a la complejidad y el dinamismo que una red puede presentar en función al comportamiento de los usuarios. Varias investigaciones [9], [21], [22], [33] sugieren usar otros algoritmos diferentes a los ya analizados para observar su eficacia bajo diferentes escenarios, considerando parámetros claves que influyen directamente en el rendimiento de una red.

2.2.9 Clasificación de las técnicas de aprendizaje automático

Aprendizaje supervisado

El algoritmo de aprendizaje automático es una técnica de machine learning que se basa en un conjunto de datos de entrada que incluye información de los datos de salida esperados, cada conjunto de datos es referido como etiquetas debido a que se utilizan para entrenar el modelo [47], según las características del sistema diseñado. A lo largo del proceso de aprendizaje, el modelo ajusta sus parámetros por medio de procesos de optimización con el fin de lograr una mejor precisión a las predicciones realizadas. Se asume la existencia de un supervisor, quien es el encargado de proporcionar la información necesaria para corregir errores y mejorar la capacidad de aprendizaje del modelo [48].

Aprendizaje no supervisado

Este tipo de aprendizaje, a diferencia del aprendizaje supervisado, se caracteriza por que los datos de entrada no cuentan con una salida esperada [47], en su lugar, el algoritmo procesa los datos con el fin de encontrar patrones o relaciones que permitan organizar la información. Debido a sus características, no es posible calcular un error que permita definir la precisión del modelo, en consecuencia, el algoritmo debe hacer uso de técnicas como los métodos de agrupamiento para estructurar la información en un conjunto de datos organizados [48].

Aprendizaje por refuerzo

A diferencia de los otros métodos, el aprendizaje por refuerzo no requiere un conjunto de datos previos para su entrenamiento debido a que funciona de forma autónoma por medio de la interacción de su entorno y su entrenamiento se basa a través de recompensas y castigos [49]. Su objetivo es que, por medio de entornos dinámicos, el algoritmo aprenda y adapte su respuesta bajo diversas situaciones, demostrando su eficacia en casos donde el entorno se vuelve dinámico y es poco probable determinar una acción específica [50].

Debido a que este modelo no cuenta con datos etiquetados, es necesario formular una política que realiza una evaluación de los estados del entorno y las acciones a tomar con el fin de maximizar la recompensa [51]. Para realizar este proceso de manera óptima, se hace uso de algunos elementos fundamentales como: el agente, que representa el modelo a ser entrenado para la toma de decisiones, el entorno, el cual es el espacio en donde se definen las reglas del sistema, el estado, que especifica la configuración que el agente percibe de acuerdo al entorno y la acción, que es la decisión tomada por el agente en función al estado actual [49],

[50]. De forma general, se considera que la interacción con el entorno es una parte fundamental para actualizar las recompensas del modelo. Actualmente, el aprendizaje por refuerzo ha ido tomando relevancia en el área de las telecomunicaciones debido a su capacidad de adaptarse a entornos dinámicos, lo que lo convierte en una herramienta importante para el análisis de la optimización de redes y sistemas inteligentes [51].

▪ Algoritmo Q-Learning

El algoritmo Q-Learning es un método de aprendizaje por refuerzo que se basa en la interacción del entorno para realizar un objetivo específico [49]. De este modo, el algoritmo, por medio de los valores Q (Q-values), aprende a tomar decisiones basándose en la condición que presente un estado por medio de una acción. A medida que el agente explora su entorno, recibe recompensas para ajustar la toma de decisiones con respecto a la política que sigue para aprender, si se realiza un incentivo adecuado, el modelo es capaz de aprender sin supervisión [52]. Al hablar de política, nos referimos a la estrategia que toma el agente para decidir las acciones en base a su estado actual. Otra de las características que presenta este algoritmo es que se considera un algoritmo off-policy, es decir, que el agente usa una política diferente a la óptima para explorar el entorno [51], [53].

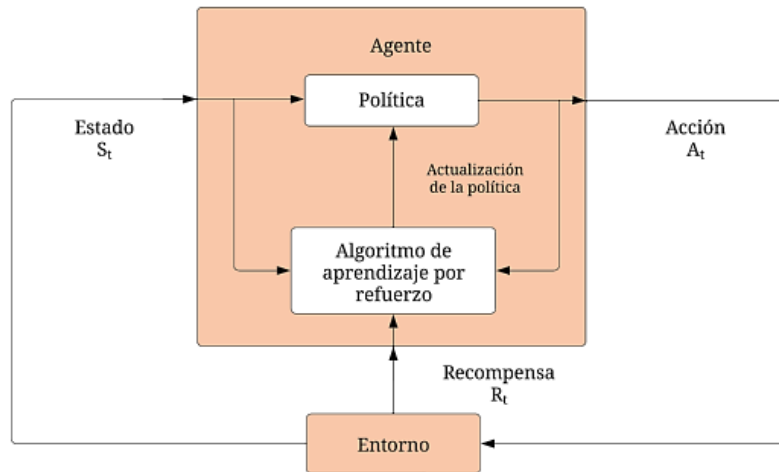


Figura 4. Proceso de actualización de la política en el algoritmo Q-Learning [49].

Los resultados entre la interacción del agente con la función de recompensas se guardan en una matriz denominada matriz Q, donde su organización se representa mediante los estados (filas) y las acciones posibles que puede tomar el algoritmo (columnas) [53], esta tabla se actualiza en cada iteración haciendo uso de la ecuación de Bellman que se define de la siguiente forma:

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a)] \quad (5.3)$$

Donde:

- α se refiere a la tasa de aprendizaje y es el encargado de determinar cuánto se ajusta el valor Q en cada actualización.
- γ es el factor de descuento e indica cuánto se valoran las recompensas futuras en comparación de las recompensas inmediatas.
- r es la recompensa que se obtiene tras la ejecución de la acción a y representa la máxima recompensa esperada.

En la matriz Q se muestra el valor esperado al ejecutar una acción en un estado determinado, con esto se logra cumplir con el objetivo del algoritmo Q-Learning, el cual es elegir la mejor opción para maximizar su recompensa y acercarse a la política óptima a lo largo del tiempo a medida que el agente interactúa con el entorno. Es utilizado comúnmente en aplicaciones de navegación autónoma de robots [54], donde, a diferencia de otros algoritmos, aprende en base a la experiencia obtenida durante la ejecución del proceso de interacción. Una de las limitaciones que presenta este algoritmo es que necesita de múltiples interacciones para lograr alcanzar el estado óptimo. Esto se debe a que las recompensas obtenidas afectan al valor estado-acción [55], mientras que el resto de combinaciones se modifican de forma indirecta, en consecuencia, el valor Q se altera de forma progresiva.

En los sistemas de redes celulares, su implementación radica en optimizar la asignación de recursos en cada estación base según las condiciones del entorno. Estas condiciones varían de forma constante debido a la movilidad y el número de usuarios que se encuentran en una celda, la carga de tráfico, entre otros aspectos. A medida que se optimice el sistema, el algoritmo se adapta a los cambios que se puedan producir y logra buscar la máxima recompensa para garantizar una buena calidad de servicio a los usuarios. En este caso, mediante la implementación del algoritmo, se busca realizar el balanceo de carga en diferentes celdas para lograr optimizar los recursos de una red LTE mediante el proceso de handover horizontal. Para observar su comportamiento, se ha propuesto varios escenarios de prueba con un número de usuarios variable para cada celda.

CAPÍTULO III METODOLOGIA

3.1 Tipo y diseño de la investigación

La presente investigación tiene un enfoque de tipo cuantitativo, ya que mediante simulaciones se obtienen los valores de las métricas del throughput, el número de handovers realizados y el índice de equidad de carga, los cuales están directamente relacionados con el rendimiento de la red. A partir de estos valores, se evalúa el algoritmo de handover horizontal mediante métodos estadísticos en tres distintos escenarios de red.

Tipo de investigación

La investigación se considera de tipo exploratoria, debido a que, al ser desarrollada mediante un modelo de simulación, se exploran distintas configuraciones de red variando el número de usuarios, la distribución en las celdas y la asignación de servicios que simulan distintas cargas de tráfico. Mediante este mecanismo, se busca identificar cómo se comporta la red bajo diferentes condiciones operativas. Para identificar las tendencias o condiciones de operación, se obtienen los valores del throughput, el número de handovers realizados y el índice de equidad de carga. A su vez, debido a que el estudio no se realiza en un entorno real, sino que se modela un entorno controlado, refuerza el tipo de investigación planteada.

3.2 Técnicas de recolección de datos

Observación

La técnica de recolección de datos empleada en la investigación es la observación, ya que a través de los parámetros asignados para la simulación de la red 4G LTE se logra obtener información acerca del comportamiento del sistema antes y después de aplicar el algoritmo de handover horizontal basado en Q-Learning. Con base a los parámetros iniciales de cada escenario, se registran métricas cuantitativas que permitirán validar el algoritmo para la toma de decisiones de handover eficientes. La técnica de observación implica un registro y seguimiento de los resultados generados, tanto numéricos como gráficos. Debido al número de simulaciones realizadas para la obtención de los datos, se presenta un caso representativo por escenario donde se muestran de forma gráfica los resultados obtenidos por simulación.

3.3 Población de estudio y muestra de tamaño

3.3.1 Población

La población conforma el conjunto de datos generados a partir de todas las posibles distribuciones de usuarios móviles y condiciones que pueden presentarse en los 3 escenarios de simulación establecidos, como son: Rural Macro (RMa), Urban Macro (UMa). Urbano Micro (UMi).

3.3.2 Muestra

La muestra se constituye de tres escenarios distintos de red 4G LTE en los que se varía el número de celdas a 5 (RMa), 10 (UMa) y 15 (UMi) para representar diferentes entornos. Para cada escenario propuesto se ejecutan 60 simulaciones independientes, en las cuales la

asignación de usuarios a cada celda se realiza de forma aleatoria, esto con el objetivo de que el sistema logre interactuar con una variedad representativa de configuraciones posibles.

3.4 Hipótesis

Hipótesis nula (H_0)

El algoritmo de handover horizontal basado en aprendizaje por refuerzo no incrementa significativamente el throughput promedio, y por lo tanto no mejora la Calidad de Servicio (QoS) en la red 4G LTE.

Hipótesis alternativa (H_a)

El algoritmo de handover horizontal basado en aprendizaje por refuerzo incrementa significativamente el throughput promedio, lo que implica una mejora en la Calidad de Servicio (QoS) en la red 4G LTE.

3.5 Operacionalización de las variables

| Variable | Concepto | Indicador | Técnica e instrumentación |
|---|--|----------------------------------|---|
| Independiente Carga de la red | Demanda de los recursos de la red en función al entorno y el número de usuarios activos en un momento determinado. | - Número de celdas. | Simulación de una red 4G LTE mediante el software Matlab que permite configurar distintas cargas en la red variando el número de celdas según el entorno. |
| Dependiente Calidad de Servicio | Nivel de rendimiento de la red. | - Throughput - Índice de Jain | Recolección de datos por simulación antes y después del algoritmo de handover horizontal basado en Q-Learning. |

Tabla 5. Operacionalización de las variables.

3.6 Métodos de análisis y procesamiento de datos.

Se utiliza una muestra de 60 simulaciones independientes para cada escenario propuesto, cada simulación presenta una distribución aleatoria de usuarios móviles por celda y condiciones variadas de carga de tráfico a través de la asignación de diferentes servicios. Para el análisis respectivo, en cada simulación se registran métricas como el throughput, la distribución antes y después de aplicar el algoritmo de handover basado en Q-Learning y el número de handovers realizados.

3.7 Diagrama de procesos

La metodología incluye la revisión del estado de arte, el diseño de escenarios para diferentes densidades de carga, la configuración de parámetros de red, la asignación dinámica de usuarios y la definición de estados, acciones y recompensas para el agente de aprendizaje.

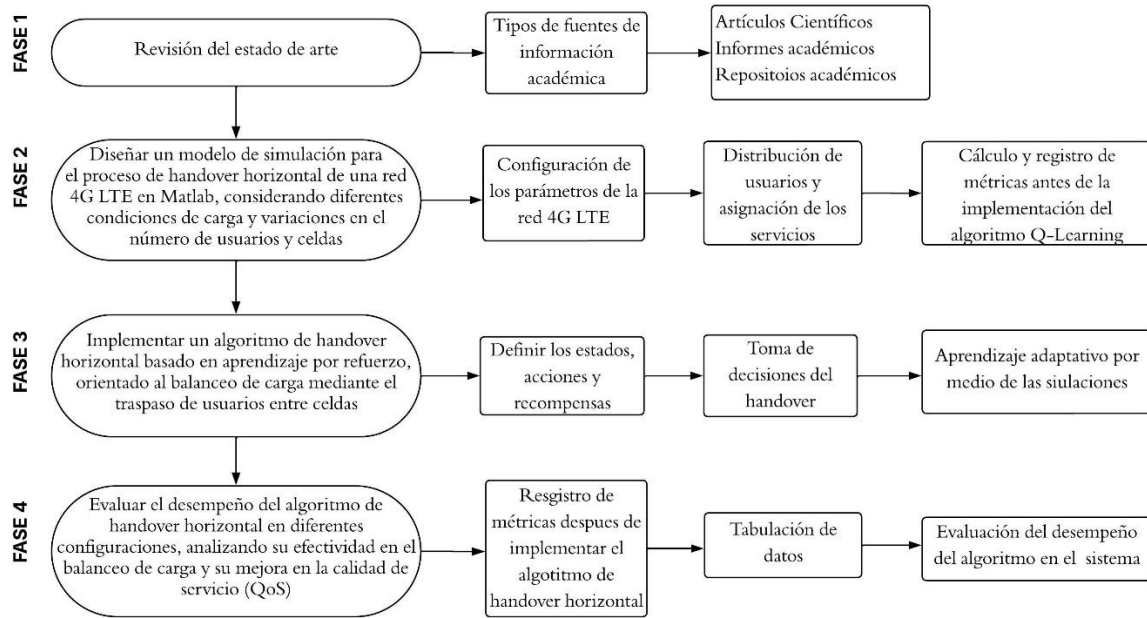


Figura 5. Fases de trabajo.

Fase 1. Revisión del estado del arte

Para esta fase de trabajo se utilizaron herramientas de búsqueda académica como, IEEE Xplore, Google Scholar, repositorios de diferentes instituciones y MDPI para acceder a fuentes documentales que incluyen artículos científicos, artículos de congresos académicos referentes al área de telecomunicaciones, tesis de pregrado y posgrado. Los temas abordados tienen relación con las redes móviles 4G LTE, la gestión del proceso de handover, el balanceo de carga en redes celulares, la aplicación de algoritmos o modelos matemáticos y las métricas relacionadas al rendimiento del sistema. Se recopilieron 90 documentos, de los cuales 57 son usados para el desarrollo de la investigación, 28 de ellos corresponden a tesis de pregrado de diferentes instituciones, 7 a tesis de posgrado, 7 a páginas web encontradas en línea confiables pertenecientes a instituciones como el Ministerio de Telecomunicaciones, 10 a artículos académicos y 5 a artículos de conferencias e informes académicos.

Fase 2. Diseño de un modelo de simulación de una red 4G LTE

En esta fase se diseña un modelo de simulación de una red 4G LTE mediante el software de Matlab y se definen 3 escenarios con 5, 10 y 15 celdas respectivamente. A cada escenario se le asignan parámetros como el ancho de banda, la frecuencia, la potencia de transmisión, los PRBs y su modelo de propagación. En cada celda se distribuyen los usuarios en posiciones aleatorias para simular el comportamiento de una red real y se les asignan servicios de voz, video y datos con distintos niveles de consumo de recursos para obtener diferentes condiciones de tráfico. Antes de implementar el algoritmo basado en aprendizaje por

refuerzo, se calculan y registran las métricas del sistema, como el SINR por usuario importante para realizar el cálculo del throughput individual y global y el uso de PRBS por celda.

Fase 3. Implementación de un algoritmo de handover horizontal basado en Q-Learning

Una vez realizado el modelo de simulación de una red 4G LTE, se implementa un algoritmo de handover horizontal basado en Q-Learning. Para ello, se define un conjunto de estados en función al nivel de SINR que recibe cada usuario y la carga actual de la celda origen. Las acciones se definen en dos: transferir los usuarios a una celda destino que mejore sus condiciones según las métricas calculadas o quedarse en su celda origen. La recompensa es establecida en función de la mejora del throughput, el balanceo de carga y el efecto positivo en el índice de equidad después del handover. Como el aprendizaje del algoritmo se realiza mediante la exploración del entorno y la toma de decisiones por cada simulación realizada, el agente refuerza la mejora del sistema y penaliza las decisiones que afecten a su desempeño.

Fase 4. Evaluación del desempeño del algoritmo

En la última fase se registraron las métricas obtenidas después de aplicar el algoritmo de aprendizaje por refuerzo en 60 simulaciones por cada escenario, luego se tabularon los datos para su análisis respectivo y se verifica que las decisiones realizadas por el agente para el proceso de handover contribuyen a la mejora del sistema de la red. Se aplican métodos estadísticos para comprobar si las métricas obtenidas antes y después de la implementación del algoritmo son significativas estadísticamente.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1 Modelos de propagación

Los modelos de propagación definidos para cada escenario se caracterizan por determinar el comportamiento de la señal cuando se desplaza de una estación base hacia el usuario móvil de acuerdo con los parámetros físicos de la red, como la distancia, la frecuencia y los parámetros de la altura de la estación base. A partir de las recomendaciones del estándar 3GPP TR 36.814 [56], los escenarios en el modelo de simulación se clasifican en tres tipos.

4.1.1 Modelo Rural Macro (RMa)

Este modelo está relacionado con entornos urbanos abiertos, donde las condiciones de línea de vista entre la estación base y los usuarios no se ven afectadas por edificaciones. Generalmente, el número de celdas en este tipo de entorno son bajas por que el modelo se centra más en ampliar su cobertura.

4.1.2 Modelo Urban Macro (UMa)

Para el modelo Urban Macro, las condiciones de línea de vista considera edificaciones de altura media, donde, para equilibrar el rango de cobertura y la capacidad, aumenta el número de celdas con respecto al modelo anterior.

4.1.3 Modelo Urban micro (UMi)

Debido a que este modelo representa entornos urbanizados con alta congestión, las celdas tienen una cobertura más reducida y el modelo de propagación se ve afectado por las condiciones propias del entorno, como las pérdidas por obstáculos.

4.2 Arquitectura del sistema de red 4G LTE simulado

La arquitectura del sistema LTE simulado se divide en 3 componentes: el equipo de usuario (UE), la estación base (eNodeB) y el núcleo de red. Cada componente se comunica entre sí a través de interfaces y cumplen con un papel fundamental para el funcionamiento de la red.

Equipo de usuario (UE)

En la simulación realizada, el equipo de usuario corresponde a todos los usuarios asignados a cada celda, representados por una estructura de datos que contiene información sobre la posición en coordenadas (x,y), un identificador único, la celda a la que está conectado, la cantidad de PRBs que utiliza y la calidad de señal que recibe. El modelo implementado está basado en la capa de acceso por radio, ya que se enfoca principalmente en la interacción de la estación con el usuario.

Estación base (eNodeB)

La estación base se representa en cada una de las celdas y es la encargada de establecer la conexión de los usuarios dentro de la zona de cobertura. Dentro de la simulación, la estación base se encarga de administrar los recursos de radio para los usuarios según el tipo de

servicio, calcula el throughput y ayuda al algoritmo de aprendizaje por refuerzo a tomar decisiones para realizar el proceso de handover.

Evolved Packet Core (EPC)

El Evolved Packet Core es el encargado de gestionar la conectividad entre los usuarios y las redes externas. A pesar de que no se presenta gráficamente en la interfaz, se asume su presencia cuando se realizan los handovers entre una estación base a otra.

4.2.1 Interfaces del sistema

Enlace de radio entre el UE y el eNodeB (Uu)

Es el canal que conecta al usuario con su estación base, mediante esta interfaz se calculan parámetros definidos en el sistema como la pérdida de trayecto y la relación señal-interferencia-ruido (SINR). Estos valores permiten establecer ciertos parámetros para decidir si el usuario debe realizar el proceso de handover.

Interfaz lógica entre estaciones base (X2)

Su función es conectar las estaciones base vecinas para permitir la transferencia de usuarios de una celda a otra sin pasar por el núcleo EPC, facilitando la ejecución de handovers. En el modelo de simulación, esta interfaz se representa por la matriz neighborTable, la cual define las celdas que están conectadas entre sí y, por lo tanto, las que pueden intercambiar usuarios.

Enlace entre el eNodeB y el EPC (S1)

A pesar de que en el modelo de simulación no se representa de forma gráfica, se asume su existencia, ya que después de realizar el proceso de handover los usuarios siguen conectados a la red.

4.3 Entorno de simulación

4.3.1 Software

El presente trabajo de investigación es realizado en el software de Matlab R2025a utilizando la herramienta integrada App Designer, ya que permite crear interfaces gráficas y proporciona una estructura basada en objetos que organiza el código en bloques independientes que se conectan entre sí para un funcionamiento integral [57]. Su elección se basa en la capacidad que tiene la herramienta para la incorporación de componentes que permiten ajustar parámetros del entorno y visualizar gráficos en tiempo real, en este caso de las condiciones de la red 4G LTE.

4.3.2 Interfaz Gráfica

La interfaz gráfica se encuentra organizada por componentes que permiten al usuario seleccionar el entorno y con ello ejecutar automáticamente los parámetros LTE definidos en el código, generar usuarios, visualizar el entorno una vez definidos todos los parámetros y aplicar el algoritmo de handover horizontal.

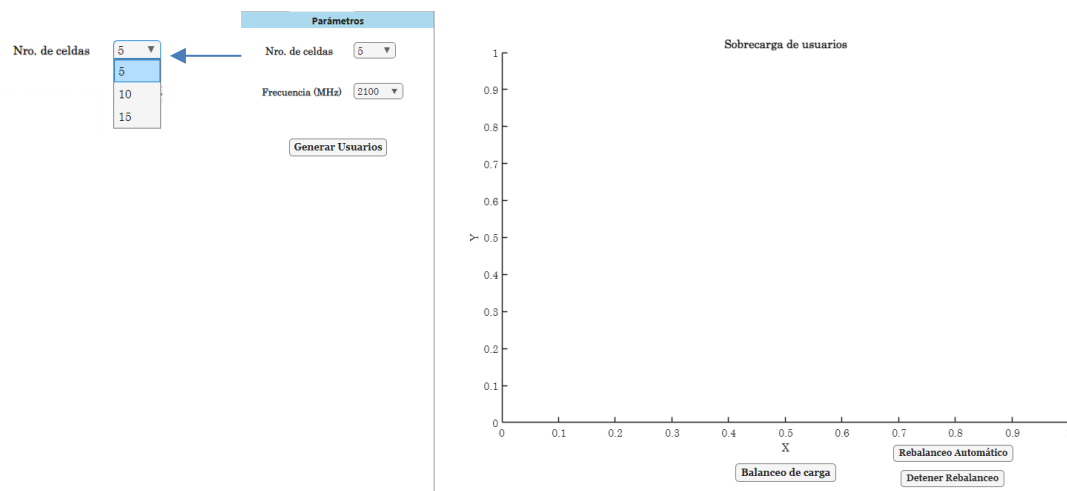


Figura 6. Interfaz gráfica del simulador LTE desarrollado en App Designer.

Como se muestra en la

Figura 6, en el panel lateral izquierdo se puede definir el número de celdas entre 5, 10 y 15 y la banda de frecuencia, inmediatamente después de seleccionar el número de celdas, se gráfica en el UIAxes las celdas hexagonales y los radios de cobertura. Con el botón Generar usuarios, se genera distinto tipo de tráfico agregando distinto número de usuarios a cada celda en posiciones aleatorias. Para interactuar con el algoritmo de handover horizontal basado en Q-Learning, en la parte inferior de la interfaz se encuentra el botón de Balanceo de carga, que ejecuta por única vez el algoritmo implementado, el botón de Rebalanceo automático, que ejecuta un ciclo continuo de acuerdo con un valor definido en el código y el botón Detener Rebalanceo para pausar las simulaciones cuando se ejecute el ciclo.

4.4 Generación y configuración del escenario 4G LTE

La configuración del escenario de red 4G LTE sigue varios procesos que se resumen en: generar usuarios con terminales móviles, sincronizar el canal e inicializar la Q table para la implementación del algoritmo de handover horizontal. Estos procesos se realizan según el tipo de escenario escogido.

4.4.1 Asignación de recursos y topología de la red

Al seleccionar el número de celdas, se ejecuta el Script 1, donde se extrae el número seleccionado de celdas y la frecuencia desde la herramienta dropdown de App Designer.

```
numCeldas = str2double(app.NrodeceldasDropDown_2.Value);
frecuencia = str2double(app.FrecuenciaMHzDropDown_2.Value);
app.configurarAnchoBanda(numCeldas);
app.cellPositions = app.calcularPosicionesCeldas(numCeldas);
app.actualizarVisualizacionCeldas();
```

Script 1. Inicialización de parámetros.

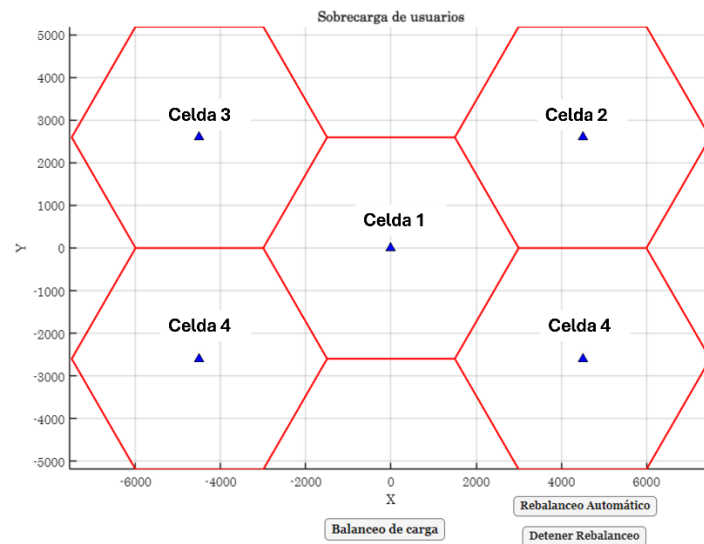
Mediante la función configurarAnchoBanda, según los parámetros definidos en la Tabla 6, se asigna el ancho de banda del sistema, el número de bloques de recursos físicos (PRB) para cada celda y el radio de cobertura.

Tabla 6. Parámetros asignados según el número de celdas de la red.

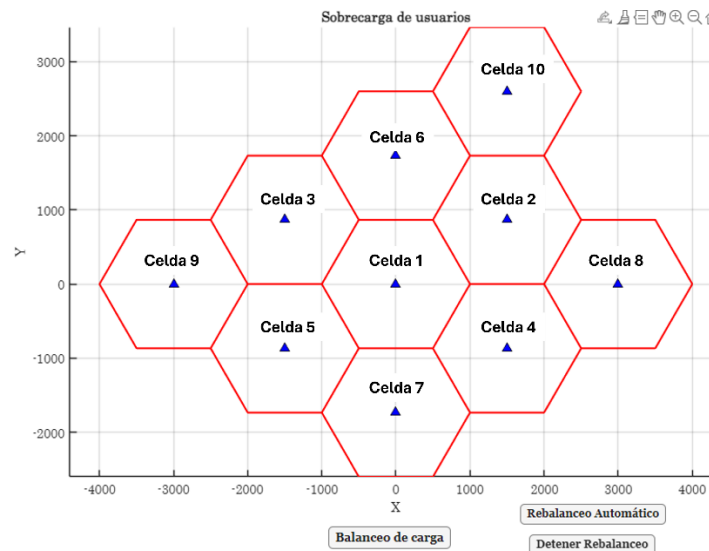
| Nro. de celdas | Ancho de Banda (MHz) | PRBs por celda | Radio (Km) |
|----------------|-------------------------|----------------|------------|
| 5 | 10 | 50 | 3.0 |
| 10 | 20 | 100 | 1.0 |
| 15 | 20 | 100 | 0.5 |

Para calcular las posiciones de las celdas se considera el valor del radio de cobertura r y a partir de una celda central ubicada en el punto (0,0) se colocan las celdas restantes. Para el escenario de 5 celdas las posiciones se definen utilizando una distancia horizontal de $1.5 * r$ y una distancia vertical de $\sqrt{3} * r/2$, en el caso del escenario de 10 celdas se usa una distancia horizontal de $3 * r$ y una distancia vertical de $\sqrt{3} * r$ y para el escenario de 15 celdas se añade una distancia vertical de $2 * \sqrt{3} * r$. Finalmente, con `app.actualizarVisualizacionCeldas()` se grafican las celdas en el UIAxes de la interfaz.

(a)



(b)



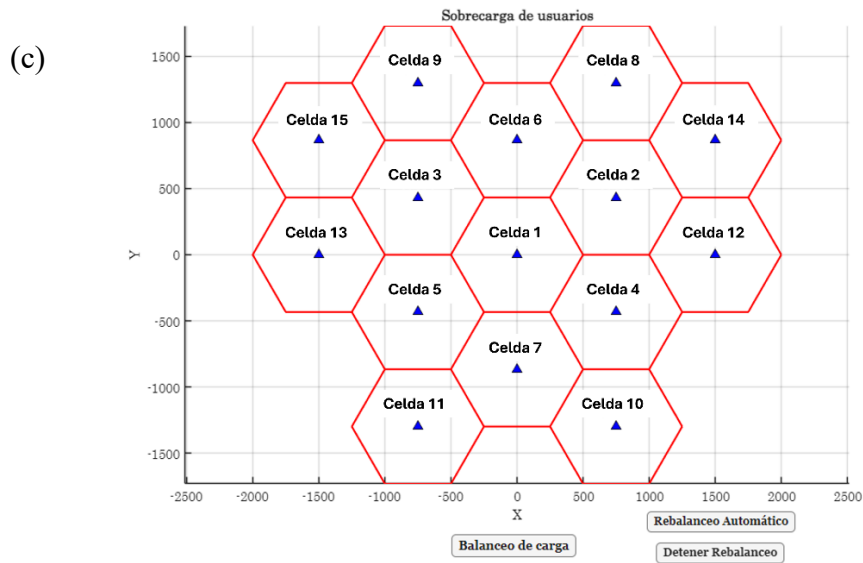


Figura 7. Distribución de celdas para los distintos escenarios simulados: (a) red con 5 celdas (RMa), (b) red con 10 celdas (UMa), (c) red con 15 celdas (UMi).

La selección de los tres escenarios busca probar la capacidad de adaptación del algoritmo de handover frente a diferentes topologías de red y variaciones en la carga de tráfico.

4.4.2 Generación de usuarios

Después de asignar los recursos y graficar la topología de red, se inicia la distribución de usuarios activando el callback `GenerarUsuariosButton_2Pushed`. Antes de iniciar este proceso, el sistema limpia cualquier información que se haya almacenado en simulaciones anteriores. Dependiendo del valor de p y de acuerdo con el rango determinado en el Script 2, cada celda puede tener una cantidad distinta de usuarios.

```
p = rand;
if p < 0.33
    defaultRange = [10, 20];
elseif p < 0.66
    defaultRange = [20, 35];
else
    defaultRange = [35, 60];
end
numUsuarios = randi(defaultRange);
```

Script 2. Niveles posibles de usuarios.

El objetivo es simular usuarios móviles distribuidos de forma aleatoria y lograr celdas congestionadas mediante condiciones reales.

4.4.3 Posicionamiento y asignación de servicios a los usuarios

Mediante la función `dibujarUsuariosEnCelda` se generan coordenadas aleatorias ubicando a cada usuario dentro de su celda correspondiente y se establece un método que evita que los usuarios queden fuera de la topología hexagonal de la celda. El tipo de servicio se clasifica

en voz, video o datos y es asignado a cada usuario con una demanda distinta de recursos según lo establecido en el **Script 3**.

```
servicioConfig = struct(...  
    'VoIP', 1, ...  
    'Video', 4, ...  
    'Datos', 2);
```

Script 3. Asignación de PRBs a cada servicio.

Después, se calcula la carga total sumando los PRBs consumidos por todos los usuarios de cada celda y se almacena en un vector llamado `cargaCeldas` para posteriormente usarlo en la toma de decisiones para el balanceo de carga.

4.4.4 Configuración del modelo del canal

Se establecen los parámetros físicos necesarios para calcular la atenuación de la señal desde el transmisor, en este caso, desde la estación hasta el usuario móvil. Para ello se actualizan los valores dentro de una estructura que contiene los parámetros físicos del canal LTE y se activa después de generar las posiciones de los usuarios. En el Script 4, se muestran las instrucciones utilizadas para actualizar dicha estructura.

```
app.escenario = app.obtenerEscenario();  
% Sincronizar los parámetros físicos en channelModel con el  
escenario actual  
app.channelModel.hBS = app.escenario.hBS;  
app.channelModel.hUT = app.escenario.hUT;
```

Script 4. Asignación de parámetros físicos al modelo de canal.

Como primer paso, se ejecuta la función `obtenerEscenario`, la cual devuelve una estructura con parámetros relacionados a la altura de la estación base (`hBS`), la altura del usuario (`hUT`) y la frecuencia seleccionada desde la interfaz a una variable llamada `app.escenario`. Algunas de funciones del sistema necesitan de todos los datos mencionados para ejecutar su proceso. Sin embargo, existen otras funciones que solo necesitan los datos de la altura de la estación base y la altura del usuario. Para evitar dependencias innecesarias entre estructuras, se toman los dos datos necesarios y pasan a otra estructura llamada `channelModel`.

4.4.5 Movilidad de usuarios

Implementar el comportamiento de movilidad de usuarios en un modelo de simulación de una red 4G LTE resulta significativo para el cálculo de las métricas y para iniciar el proceso de handover. En el sistema presentado, la movilidad se clasifica en dos tipos, vehicular y peatonal. A cada usuario se le asigna un tipo de movilidad mediante una distribución equitativa y, según el tipo asignado, mediante el factor de movimiento descrito en el script 5, se establece un desplazamiento proporcional al radio de la celda.

```
factor Movimiento = [0.001, 0.02]; % [peatones, vehículos]  
proporciones = [0.5, 0.5]; % 50% peatones, 50% vehículos
```

Script 5. Configuración de la función de movilidad.

Para simular la trayectoria de cada usuario, se asigna un ángulo aleatorio entre 0 y 360°. En cada simulación, se determina que el 30% de usuarios cambien su dirección inicial por una nueva. Esto, se realiza con el propósito de modificar la dirección original para simular cambios en la trayectoria por obstáculos. Una vez definido el desplazamiento y la dirección, se toman estos parámetros y la función `simularmovilidad` los convierte en coordenadas (x,y), usando funciones trigonométricas para la nueva posición, si esta sobrepasa el radio de cobertura de la celda, se invierte de forma automática.

4.4.6 Inicialización de la tabla Q

Al finalizar el proceso de configuración de la red, se verifica la existencia de la tabla Q utilizada por el algoritmo de aprendizaje por refuerzo. Esta tabla contiene los valores que le permiten al sistema decidir cuando un usuario debe realizar el proceso de handover o mantenerse en su celda actual.

```
if isempty(app.qTable)
    app.inicializarQTabla();
    disp('Q-table inicializada');
else
    disp('Q-table ya existe, se conserva el aprendizaje
acumulado');
end
```

Script 6. Inicialización de la tabla Q.

Una vez verificado que la tabla haya sido creada se reutiliza su aprendizaje acumulado en simulaciones anteriores y continua el proceso de entrenamiento, caso contrario, se crea una nueva tabla considerando todas las combinaciones posibles de estados y acciones definidas por el sistema.

4.5 Modelo matemático del sistema de red 4G LTE

A continuación, se realiza un análisis de los cálculos utilizados para evaluar el desempeño de la red.

4.5.1 Pérdida de trayecto

La pérdida de trayecto (*path loss*) en el sistema de red 4G LTE modelado, se calcula mediante la función `calcularPerdidaTrayectoria`, según el número de celdas seleccionado, se clasifica el entorno en: 5 celdas (RMa), 10 celdas (UMa) y 15 celdas (UMi).

Entorno Rural Macro

Para este entorno se utiliza un modelo definido en 3GPP [56], que permite estimar la pérdida de trayectoria en función de la distancia entre el usuario y la estación base, así como la frecuencia de operación.

```
case 'RMA'
    % Modelo 3GPP para Radio Macro
    perdida = 128.1 + 37.6 * log10(d) + 20 * log10(fc_GHz / 2);
```

Script 7. Cálculo de la pérdida de trayectoria para la modelo RMa.

Esta expresión considera tres componentes: una constante de propagación, un término logarítmico que representa la atenuación de la señal con respecto a la distancia y un término que incorpora el efecto de la frecuencia.

Entorno Urbano Macro

Dado que el modelo de propagación UMa representa áreas urbanas con densidad moderada, para el cálculo de la pérdida de trayectoria se usa la ecuación presentada en el Script 8.

```
case 'UMA'
    % Modelo 3GPP para Urban Macro
    perdida = 128.1 + 37.6 * log10(d) + 20 * log10(fc_GHz)/2;
```

Script 8. Cálculo de la pérdida de trayectoria para el modelo UMa.

A pesar de la similitud con la ecuación presentada en el modelo RMa, aquí se ajusta el efecto de la frecuencia de manera moderada.

Entorno Urban Micro

En este modelo se simulan entornos urbanos densamente poblados y con condiciones que dificultan la línea de vista.

```
case 'UMI'
    % Modelo 3GPP para Urban Micro
    hBS = app.escenario.hBS;
    perdida = 140.7 + 36.7 * log10(d) + 26 * log10(fc_GHz) - 0.3 * (hBS
- 1.5);
```

Script 9. Cálculo de la pérdida de trayectoria para el modelo UMi.

La ecuación empleada considera una constante elevada en comparación de los dos modelos anteriores, un término que refleja el efecto de la distancia entre el eNodeB con el Usuario y una componente que ajusta la pérdida de trayectoria según la frecuencia de operación. Se incluye, además, un término que reduce la pérdida si la estación base está a baja altura.

4.5.2 Potencia recibida

La potencia recibida calcula la intensidad de la señal útil que recibe el usuario desde la celda conectada. En el modelo de simulación de la red para realizar el cálculo de la potencia recibida, se considera parámetros físicos del sistema presentados en la **¡Error! No se encuentra el origen de la referencia..**

| Descripción | Parámetro | Valor | Unidad |
|------------------------------------|-------------------------------|---|--------|
| Potencia de transmisión del eNodeB | potenciaTx | 46 | dBm |
| Ganancia de antena transmisora | gananciaAntena | 18 | dBi |
| Ganancia de la antena receptora | gananciaUE | 0 | dBi |
| Desviación estándar del shadowing | app.channelModel.shadowFading | 8 | dB |
| Pérdida de trayecto | perdida | Calculado en la sección 4.5.1 según el modelo de propagación. | dB |

Tabla 7. Parámetros utilizados para el cálculo de la potencia recibida.

Inicialmente, se modela el canal de propagación considerando el desvanecimiento lento (*shadowing*) y el desvanecimiento rápido (*Rayleigh*).

- En el desvanecimiento lento se reflejan las variaciones que presenta la potencia debido a obstáculos físicos y se modela como una variable aleatoria con distribución normal de media cero y una desviación estándar de 8 dB.
- Para el desvanecimiento rápido, se considera que las ondas de radio emitidas por la estación base no siguen una misma trayectoria, debido a los objetos que se encuentran en el entorno. Como consecuencia, en algunos casos la señal puede interferirse entre sí, generando fluctuaciones rápidas en la intensidad de la señal.

```
shadowing = randn() * app.channelModel.shadowFading;
rayleighFading = 20 * log10( sqrt(-2*log(rand())) );
fadingFactor = rayleighFading + shadowing;
potenciaRecibida = potenciaTx + app.gananciaAntena +
app.gananciaUE - perdida + fadingFactor;
```

Script 10. Cálculo de la potencia recibida con efectos de fading.

Estos dos componentes se combinan para generar un único factor de desvanecimiento y se suma a la potencia recibida para obtener un valor más real de acuerdo con las condiciones del entorno.

4.5.3 Cálculo de SINR

El SINR (*Signal to Interference plus Noise Ratio*) es una medida que permite evaluar la calidad del enlace entre la celda que brinda el servicio y el usuario. La implementación del cálculo en el sistema se basa en los siguientes pasos:

```
nivelRuido = app.ruidoTermico + 10 * log10(anchoBandaHz);
```

```

noise_linear = 10^(nivelRuido/10);
signal_linear = 10^(potenciaRecibida/10);
interference_linear = interference_linear + activityFactor *
10^(potenciaInt/10);
sinr_linear = signal_linear / (interference_linear + noise_linear);
sinr = 10 * log10(sinr_linear);

```

Script 11. Cálculo del SINR considerando la señal útil, interferencia y el ruido térmico.

Su cálculo se realiza iniciando con la conversión de la potencia de la señal recibida de dBm a escala lineal, posteriormente se calcula el nivel de ruido térmico a partir de la densidad espectral de la potencia de ruido y el ancho de banda del sistema según el entorno escogido. Luego se calcula la interferencia total que recibe el usuario, considerando que realmente las celdas no siempre transmiten completamente el total de su potencia. Se introduce un factor de actividad que, según el valor asignado, representa el promedio del tiempo que la celda está transmitiendo. Finalmente, se aplica la fórmula de SINR para cada usuario.

4.5.4 Cálculo del throughput

Para estimar el valor del throughput, el sistema usa la fórmula de Shannon incorporando un factor de eficiencia. Mediante la función `calcularThroughput` se inicia revisando el valor de SINR y convirtiéndolo a escala lineal. Luego se usa la fórmula de Shannon, para estimar la cantidad máxima de datos que se pueden transmitir por segundo, esta fórmula establece que el throughput es proporcional al ancho de banda y a que tanto bueno resulta ser el canal, para simular un entorno más real, el resultado obtenido se multiplica por un factor que indica la capacidad real efectiva del sistema. Finalmente, el valor obtenido se convierte en megabits por segundo (Mbps).

```

function throughput = calcularThroughput(app, sinr)
    % Verificar que (sinr) sea un valor numérico
    if isnan(sinr) || ~isnumeric(sinr)
        throughput = 0;
        return;
    end
    % Convertir de dB a valor lineal.
    sinr_lin = 10^(sinr/10);
    % Evitar valores demasiado bajos
    if sinr_lin < 1e-12
        sinr_lin = 1e-12;
    end
    % Obtener el ancho de banda en (Hz) y el factor de eficiencia
    B = app.bandwidth * 1e6;
    eta = 0.8;
    % Fórmula de Shannon:
    capacity = eta * B * log2(1 + sinr_lin);
    % Convertir a Mbps y forzar un valor mínimo de 0
    throughput = max(capacity/1e6, 0.1);
end

```

Script 12. Cálculo teórico del throughput a partir del valor de SINR.

Con esta implementación, el sistema se obtiene un valor teórico del throughput, siendo la métrica clave para evaluar el rendimiento del sistema de la red simulada.

4.6 Gestión del proceso de handover

4.6.1 Criterio de cambio de celda basado en la calidad de la señal

Para la decisión de cambio de celda, se evalúa de forma individual la calidad de las celdas vecinas y se realiza una estimación del valor de SINR y del throughput que obtendría el usuario si se conectara a la celda evaluada. Una vez identificada la celda vecina que mejore las condiciones de señal, se verifica que cumpla la condición de histéresis. La cual establece un umbral mínimo de mejora necesario para autorizar el traspaso.

```
condHysteresis = (sinrNuevoOptimo - sinrActual) >=
app.handoverParams.A3Offset;
```

Script 13. Condición de histéresis basada en la diferencia del nuevo SINR y el actual.

El parámetro A3Offset se define como un margen mínimo de mejora de señal que debe cumplirse para realizar el cambio de celda. Esta condición ayuda a evitar cambios innecesarios ante variaciones menores de señal. El en Script 14, se combinan dos criterios que deben cumplirse para que el sistema considere válido el traspaso de celda.

```
if bestImprovement >= improvement_threshold && condHysteresis
```

Script 14. Condiciones para ejecutar el handover.

Se exige que la mejora estimada (*bestImprovement*) supere el umbral mínimo predefinido (*improvement_threshold*) y que la condición de histéresis supere el umbral de A3Offset. Si ambas condiciones se cumplen el sistema determina que existe una opción disponible para el cambio de celda. Sin embargo, la ejecución del handover queda sujeta a validaciones adicionales.

4.6.2 Evaluación de los umbrales de carga de la celda destino

Para evitar que el sistema tome decisiones cuando la celda destino esta congestionada y completar las validaciones para el cambio de celda, se implementó el parámetro de carga como un factor adaptativo para modificar el comportamiento del sistema dependiendo de que tan cargada se encuentre la red.

```
promedioCarga = mean(app.cargaCeldas);
capacidadTeorica = app.cellPRBs(1);
cargaFactor = promedioCarga / capacidadTeorica;
if cargaFactor < 0.3
    app.cargaUmbralBaja = 0.40;
    app.cargaUmbralAlta = 0.75;
elseif cargaFactor < 0.6
    app.cargaUmbralBaja = 0.70;
    app.cargaUmbralAlta = 0.85;
else
    app.cargaUmbralBaja = 0.40;
```

```

app.cargaUmbralAlta = 0.75;
end

```

Script 15. Ajuste dinámico de los umbrales de carga.

Primero, se calcula la carga media de la red y se divide con los PRBs asignados a cada celda según el escenario. Al dividir ambos valores se obtiene un factor de carga que representa el porcentaje de carga promedio que está siendo utilizado por toda la red. Mediante este valor se categoriza la carga en alta, media o baja, en función a esto, se ajustan de forma dinámica, los umbrales que definen cuando una celda está considerada con carga baja o alta.

4.6.3 Adaptación de parámetros de handover según el tipo de entorno

Los parámetros establecidos en la Tabla 8, son adaptados para cada escenario según el tipo de entorno.

Tabla 8. Parámetros de handover y Q-Learning.

| Entorno | TTT (ms) | A3Offset (dB) | Histéresis (dB) | Learning Rate | Discount Factor | Max Delta |
|---------|----------|------------------|--------------------|------------------|--------------------|--------------|
| RMa | 10 | -8.0 | 1.0 | 0.0005 | 0.55 | 0.05 |
| UMa | 10 | -4.5 | 1.0 | 0.004 | 0.82 | 0.05 |
| UMi | 200 | 2.0 | 1.0 | 0.10 | 0.97 | 0.07 |

Para ejecutar el proceso de handover, el parámetro de histéresis define un margen que asegura que los cambios de celda realizados sean estables y de este modo evitar el efecto ping pong. El TTT define el periodo que debe cumplir la condición de handover antes de ejecutar el cambio de celda. Estos tres parámetros relacionados al proceso de handover permiten estabilizar el sistema y realizar los cambios de celda de manera más eficiente, considerando las condiciones reales del entorno. Otros parámetros que se ajustan dentro de esta lógica son los usados por el aprendizaje por refuerzo. El learning rate define el peso que se le asigna a la nueva información al actualizar los valores Q, el discount factor determina el valor que asigna el agente a las recompensas futuras en comparación de las recompensas inmediatas y el valor de max Delta limita el tamaño máximo de cambio que se puede realizar en cada aprendizaje.

4.7 Algoritmo de aprendizaje por refuerzo

4.7.1 Descripción general del proceso del algoritmo Q-Learning

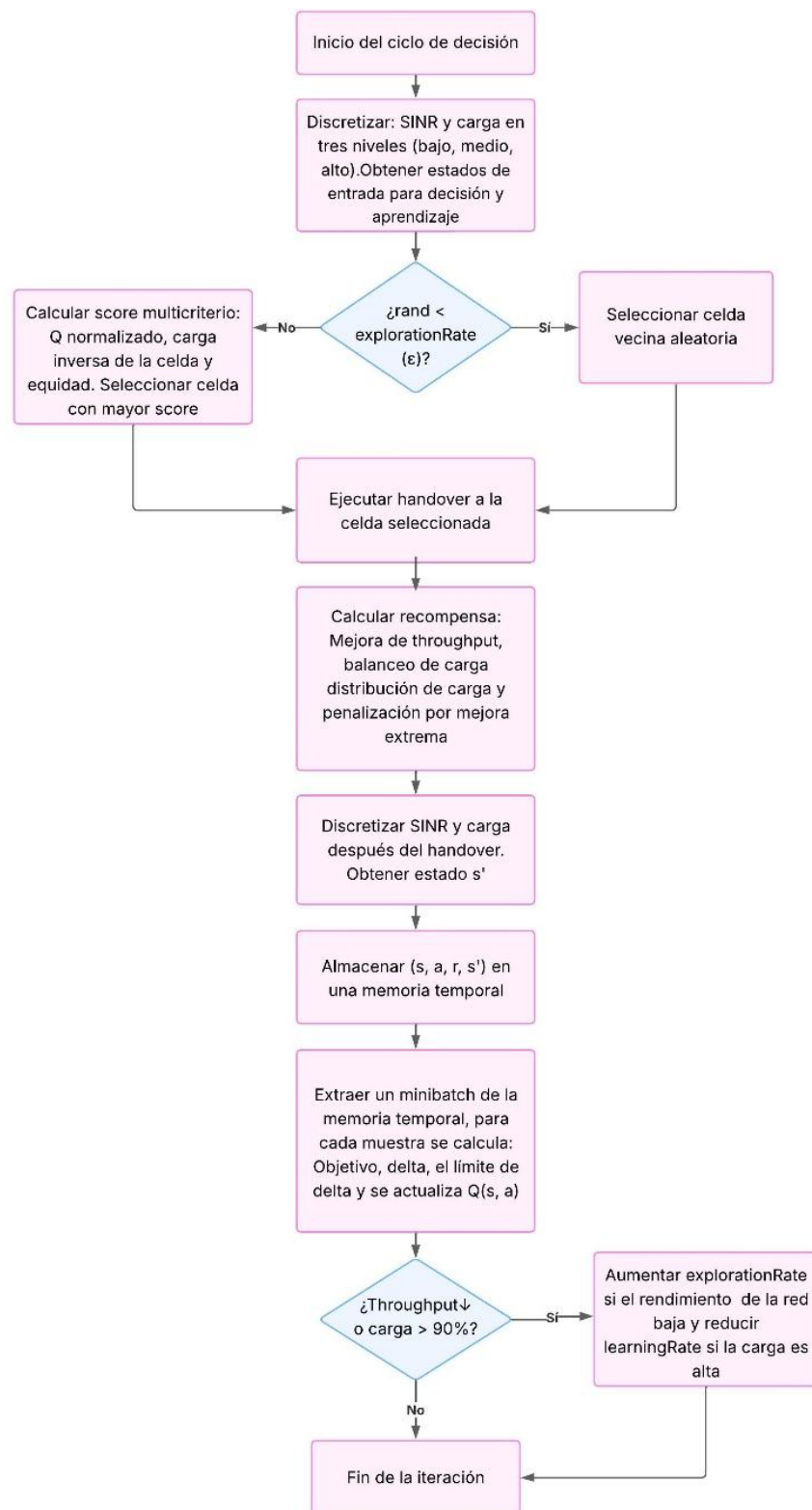


Figura 8. Flujo de ejecución del algoritmo Q-Learning aplicado al proceso de handover.

El diagrama de la Figura 8, presenta las etapas que realizó el agente para tomar decisiones de handover en la red simulada y sirve como referencia para comprender las secciones detalladas a continuación.

Estados

En el algoritmo Q-Learning, el estado representa la situación actual que percibe el agente, en este caso, el usuario de su entorno. En el modelo de simulación realizado, se combinan dos parámetros que afectan a las decisiones de cambio de celda como son: el SINR y la carga. Estos dos parámetros se discretizan en tres niveles para facilitar el aprendizaje del agente.

```
% Discretización del SINR
    if sinr < 12
        sinrLevel = 1; % Bajo
    elseif sinr < 20
        sinrLevel = 2; % Medio
    else
        sinrLevel = 3; % Alto
    end
% Discretización de la carga:
    cargaFactor = carga / capacidadTeorica;
    if cargaFactor < 0.30
        loadLevel = 1;
    elseif cargaFactor < 0.70
        loadLevel = 2;
    else
        loadLevel = 3;
    end
```

Script 16. Discretización de las variables: SINR y carga.

Al combinar los niveles de estos dos parámetros se obtiene 9 combinaciones y se almacenan en un contenedor denominado `app.stateMap`. Este contenedor asigna un identificado único del 1 al 9 a los vectores que representan la combinación de los niveles. Este proceso es importante para que agente Q-Learning pueda identificar en qué situación se encuentra.

Acciones

Las acciones representan las decisiones que puede tomar el usuario desde su estado actual. En este caso, cambiarse desde su celda origen a la celda destino en función a la tabla de vecinos mostrada en el **Script 17**, o quedarse en su celda actual.

```
switch numCeldas
    case 5
        app.neighborTable = { [2,3,4,5], [1,4], [1,5], [1,2],
[1,3] };
    case 10
        app.neighborTable = { [2,3,4,5,6,7], [1,6,8,10],
[1,6,9], [1,7,8], [1,7,9], [1,2,3,10], [1,4,5], [2,4],
[3,5], [2,6] };
    case 15
        app.neighborTable = { [2,3,4,5,6,7], [1,6,8,14],
[1,6,9,13], [1,7,10,14], [1,7,11,13], ...
```

```
[1,2,3,8,9], [1,4,5,10,11],
[2,6,12,14], [3,6,12,13], [4,7,12,15], ...
[5,7,13,15], [8,9,10,15],
[3,5,9,11], [2,4,8,15], [10,11,12,14] };
```

Script 17. Tabla de vecinos en función del número de celdas.

Estas decisiones varían según la posición del usuario, el tipo de escenario y si cumple o no las condiciones establecidas para ejecutar el proceso de handover.

Política ϵ -greedy

Para tomar las decisiones de handover se utilizó una estrategia basada en la exploración-explotación mediante la política ϵ -greedy. En la exploración, el agente realiza acciones nuevas para seguir con el proceso de aprendizaje, mientras que, en la explotación, el agente escoge la mejor acción conocida basado en los valores aprendidos. En el modelo de simulación, se realizó una comparación entre un número aleatorio y el valor de ϵ almacenado en el parámetro `app.qLearningParams.explorationRate`.

```
if rand < app.qLearningParams.explorationRate
    celdaDestino = vecinos(randi(numel(vecinos)));
    return;
end
```

Script 18. Fase de exploración en la política ϵ -greedy.

Si el valor generado por `rand` es menor al parámetro `app.qLearningParams.explorationrate` se selecciona una celda vecina sin considerar otras condiciones y se realiza el proceso de handover. Esta es la fase de exploración que le permite al agente descubrir nuevas opciones que no han sido exploradas, a pesar de que este mecanismo tiene el riesgo de seleccionar celdas que no sean lo suficientemente eficientes, es fundamental para garantizar un proceso de aprendizaje efectivo.

Si la condición del Script 18 no llegara a cumplirse, se implementó una política de selección basada en múltiples criterios. Para cada celda vecina, se calcula un score bajo tres criterios:

- **Valor Q normalizado (normQ):** es el aprendizaje acumulado del agente, según en experiencias pasadas.
- **Carga de la celda (normLoad):** nivel de carga de la celda, su valor se normaliza y luego se invierte para que las celdas con menos ocupación tengan un score más alto.
- **Equidad (fairness):** mide qué tan distribuida está la carga entre las celdas vecinas. Por ejemplo, si la desviación de la carga es baja, habrá mayor equilibrio y, por lo tanto, el valor de equidad es alto, por el contrario, con una desviación alta, el valor de equidad baja.

Estas tres medidas se combinan para calcular un score final para cada celda vecina y de acuerdo con este valor se selecciona la celda destino que tenga el mayor puntaje.

```

        % Obtener los Q-values y carga actual de las celdas vecinas
        qVals = app.qTable(estadoActual, vecinos);
        cargas = app.cargaCeldas(vecinos);
        % Normalizar los Q-values
        normQ = (qVals - min(qVals)) / (max(qVals) - min(qVals) + eps);
        % Normalizar la carga y utilizar 1 - normalización para dar
mayor preferencia a celdas con menor carga
        normLoad = 1 - ((cargas - min(cargas)) / (max(cargas) -
min(cargas) + eps));
        % Nuevo término de equidad: cuanto menor sea la varianza entre
las cargas de vecinos, mejor.
        fairness = 1 - std(cargas) / (mean(cargas) + eps);
        % Combinar las métricas (0.4 para Q-value, 0.4 para carga y
0.2 para equidad)
        scores = 0.4 * normQ + 0.4 * normLoad + 0.2 * fairness;
        [~, idx] = max(scores);
        celdaDestino = vecinos(idx);

```

Script 19. Selección de la celda destino bajo múltiples criterios.

Adicionalmente, se implementó un ajuste dinámico para el parámetro `app.qLearningParams.explorationrate` en función del desempeño del sistema.

```

% Ajuste de exploración basado en resultados globales
if globalThroughputAfter < globalThroughputBefore
    app.qLearningParams.explorationRate = min(0.9,
app.qLearningParams.explorationRate + 0.1);
else
    app.qLearningParams.explorationRate =
max(app.minimalExplorationRate, app.qLearningParams.explorationRate *
0.980);
end

```

Script 20. Ajuste adaptativo de la exploración en función del throughput global.

Si después de ejecutar el handover (acción) y calcular el throughput global, este disminuye, se considera que la decisión tomada no fue óptima y se incrementa el valor de `app.qLearningParams.explorationrate` para fomentar una mayor exploración en futuras decisiones. Por el contrario, si el throughput global aumenta, se reduce el valor de `app.qLearningParams.explorationrate` y se escoge la acción que tenga mayor score según las condiciones descritas en el Script 20.

Actualización de la tabla Q

La actualización de la tabla Q, permite que el agente aprenda cuáles son las mejores decisiones en cada estado, según las recompensas recibidas al interactuar con el entorno. En la tabla Q se almacenan los valores asociados al estado-acción, por lo que se considera el componente central del algoritmo y se actualiza usando la ecuación (5.1). En el sistema, se implementó una ecuación algebraicamente equivalente, para realizar la actualización con un valor incremental controlado llamado delta.

```

Objetivo = recompensa + app.qLearningParams.discountFactor *
max(app.qTable(estadoNuevo, :));
delta = app.qLearningParams.learningRate * (Objetivo -
app.qTable(estadoAnterior, accion));
% Limitar el delta para evitar cambios bruscos: máximo ±0.1
delta = sign(delta) * min(abs(delta), 0.1);
% Actualiza la Q-table en la posición (estadoOld,
accionSample)
app.qTable(estadoAnterior, accion) =
app.qTable(estadoAnterior, accion) + delta;

```

Script 21. Actualización del valor Q.

Los valores definidos para el factor de descuento y la tasa de aprendizaje se encuentran descritos en la Tabla 8, según el tipo entorno escogido. A través de la lógica implementada en el **Script 21**. Actualización del valor Q, se logra actualizar el sistema de forma progresiva para tomar decisiones sobre qué acciones se debe realizar en cada estado, ajustando los valores almacenados en función a las recompensas y la experiencia acumulada de forma controlada.

Es importante mencionar que el sistema no se actualiza en cada acción de forma inmediata, sino que, guarda las experiencias en una memoria temporal hasta alcanzar un tamaño máximo definido. Durante el proceso de aprendizaje se toma experiencias al azar de la memoria temporal y, con ellas, se realiza la actualización de la tabla Q mediante el Script 21.

Recompensa

En la función de recompensa se evaluó el impacto que se genera al realizar el handover de una celda a otra considerando cuatro criterios importantes:

- **Mejora del throughput**

Para cuantificar la mejora que obtiene el usuario al realizar el cambio de celda, se calcula la mejora relativa del throughput antes y después del traspaso.

```
mejora = (Thro_nuevo / Thro_antes) - 1;
```

Script 22. Cálculo de la mejora del throughput después del handover.

- **Balanceo de carga**

Adicionalmente, el sistema evalúa si el cambio de celda contribuye al balanceo de carga. Para ello se compara la carga de la celda destino con respecto a la celda origen.

```

if cargaDestino <= 0.85 * cargaOrigen
    balanceoCarga = 1 - (cargaDestino / cargaOrigen);
else
    balanceoCarga = -0.5;
end

```

Script 23. Evaluación del balanceo de carga entre celdas.

Si la celda destino tiene al menos un 15% menos de carga que la celda origen se premia el traspaso, caso contrario, se asigna una penalización de -0.5.

- **Equidad en la distribución de carga**

Mediante el índice de Jain se mide que tan balanceada esta la carga entre todas las celdas. Un valor cercano a 1 indica una buena equidad de carga mientras que un valor bajo o cercano a 0 indica un desequilibrio entre las cargas de las celdas.

```
JainIndex = (sum(app.cargaCeldas)^2) / (numel(app.cargaCeldas) * sum(app.cargaCeldas.^2));
```

Script 24. Cálculo del índice de Jain para la equidad de carga.

Una alta equidad contribuye a una mayor eficiencia y calidad de servicio, por el contrario, valores bajos reflejan desbalances en la distribución de usuarios en las celdas de acuerdo con los servicios asignados

- **Penalización por mejora excesiva**

En algunas ocasiones, una mejora excesiva en el valor del throughput puede indicar una transición poco realista del sistema. Para evitar que estos casos afecten a la estabilidad del entrenamiento del algoritmo, se implementa un mecanismo de penalización.

```
if abs(mejora) > 0.5
    penalizacion_thpt = -1.5 * abs(mejora);
else
    penalizacion_thpt = 0;
end
recompensa = recompensa + penalizacion_thpt;
```

Script 25. Penalización por mejoras excesivas en el throughput.

El mecanismo implementado regula la función de recompensa, evitando que el algoritmo se vea influenciado por valores atípicos, ayudando así a facilitar su convergencia.

4.7.2 Hiperparámetros de aprendizaje

El algoritmo Q-Learning implementado utiliza una serie de hiperparámetros que influyen de forma directa en la forma que el agente aprende y ajusta su comportamiento durante el entrenamiento.

Tabla 9. Configuración de hiperparámetros en el algoritmo Q-Learning.

| Parámetro | Valor inicial | Función |
|--------------------|---------------|--|
| learningRate | 0.05 | Controla que tanto se actualiza el valor de Q con nuevas experiencias. |
| discountFactor | 0.97 | Asigna un peso a la importancia de las recompensas futuras. |
| explorationRate | 0.25 | Probabilidad de seleccionar una acción aleatoria (exploración). |
| minExplorationRate | 0.05 | Establece un valor límite mínimo para ϵ . |
| explorationDecay | 0.998 | Multiplicador que reduce ϵ de forma gradual a lo largo del entrenamiento. |
| maxDelta | 0.1 | Límite al cambio máximo por iteración en la tabla Q para evitar fluctuaciones. |

Los parámetros como: learningRate, discountFactor y maxDelta a pesar de que se encuentran inicializados dentro de la estructura qLearningParams como un valor fijo, son ajustados de forma dinámica en función al tipo de entorno. Estos valores ajustados se presentan en la Tabla 9.

4.8 Resultados

En el presente estudio, después de haber realizado el modelo de simulación de una red 4G LTE, se valuó el sistema mediante 60 simulaciones para cada escenario, dando un total de 180 simulaciones. Cada simulación se ejecutó bajo diferentes condiciones de distribución de usuarios y condiciones de tráfico. Para analizar las mejoras del rendimiento del sistema después de aplicar el algoritmo de handover horizontal basado en Q-Learning, se calculó el valor del throughput y se lo comparó con los valores obtenidos antes y después de aplicar el algoritmo. Además, se obtuvieron métricas como la distribución de carga medida a través del índice de Jain y el número de handovers.

4.8.1 Análisis de los resultados por escenario

Métricas de desempeño del escenario de 5 celdas

Mediante la Tabla 10, se observa que el throughput promedio obtenido antes de haber aplicado el algoritmo de handover horizontal basado en aprendizaje por refuerzo es de 29.51 Mbps, mientras que, después de aplicar el algoritmo, se alcanza un valor promedio de 36.57 Mbps. Estos valores reflejan que el mecanismo implementado para tomar las decisiones de handover logra que los usuarios mejoren la calidad de servicio experimentada y, a su vez, el mismo sistema distribuya los recursos de forma equitativa para lograr evitar sobrecargas en las celdas. Para medir la distribución de carga entre las celdas se utilizó el índice de Jain, el cual alcanzó un valor promedio de 0.91.

Tabla 10. Resumen de las métricas de rendimiento para el escenario de 5 celdas.

| Métrica | Valor promedio | Valor Mínimo | Valor Máximo |
|---|-----------------------|---------------------|---------------------|
| Throughput antes de aplicar el algoritmo (Mbps) | 29.51 | 22.07 | 36.04 |
| Throughput después de aplicar el algoritmo (Mbps) | 36.57 | 29.63 | 41.51 |
| Número de Handovers | 11 | 2 | 21 |
| Equidad en la distribución de la carga (índice de Jain) | 0.91 | 0.72 | 0.99 |

La variación entre el valor mínimo y el valor máximo referente al número de handovers y al índice de Jain se debe a las condiciones del sistema presentes en cada simulación, debido a que el algoritmo es adaptativo. En simulaciones que tenga escenarios con mayor nivel de congestión de usuarios, el número de handovers aumenta.

Métricas de desempeño del escenario de 10 celdas

Este escenario representa un entorno UMa, por lo que tiene una topología más amplia y un mayor número de conexiones que el escenario anterior. En la Tabla 11, se presenta que el valor promedio del throughput antes de implementar el algoritmo fue de 45.68 Mbps, mientras que, una vez implementado el algoritmo el valor promedio asciende a 59.86 Mbps. El valor máximo del throughput fue de 59.00 Mbps, lo que demuestra que en algunas simulaciones las condiciones de la red ya son buenas en cuanto a rendimiento.

Tabla 11. Resumen de las métricas de rendimiento para el escenario de 10 celdas.

| Métrica | Valor promedio | Valor Mínimo | Valor Máximo |
|---|-----------------------|---------------------|---------------------|
| Throughput antes de aplicar el algoritmo (Mbps) | 45.68 | 38.93 | 59.00 |
| Throughput después de aplicar el algoritmo (Mbps) | 59.86 | 49.91 | 67.52 |
| Número de Handovers | 29 | 11 | 44 |
| Equidad en la distribución de la carga (índice de Jain) | 0.91 | 0.79 | 0.98 |

El promedio del número de handovers realizados fue de 29 y alcanzo un valor máximo de 44. En cambio, en el valor del índice de Jain se observó un promedio de 0.91, similar al escenario anterior.

Métricas de desempeño del escenario de 15 celdas

Este escenario representa un entorno más complejo debido a su mayor número de celdas. En la **Tabla 12**, se observa que el throughput promedio antes de aplicar el algoritmo fue de 38.02 Mbps y después de aplicarlo alcanzó un valor de 53.56 Mbps. Además, se indican

cuáles son los valores máximos y mínimos alcanzados dentro del total de simulaciones realizadas.

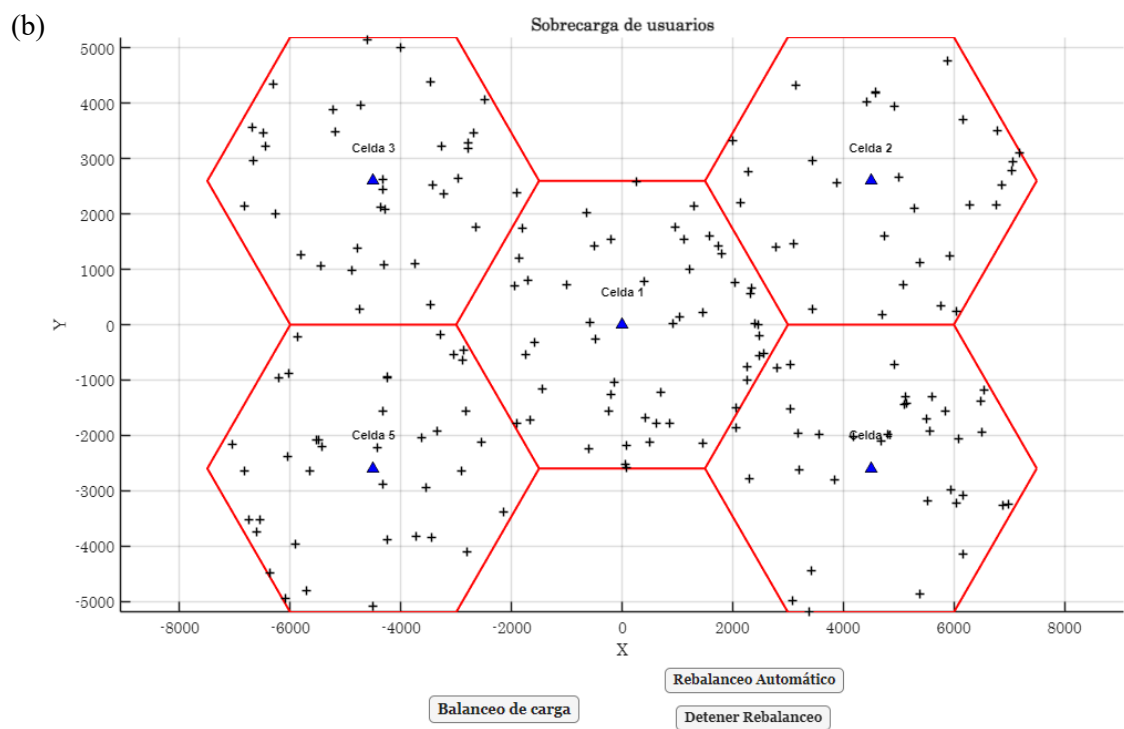
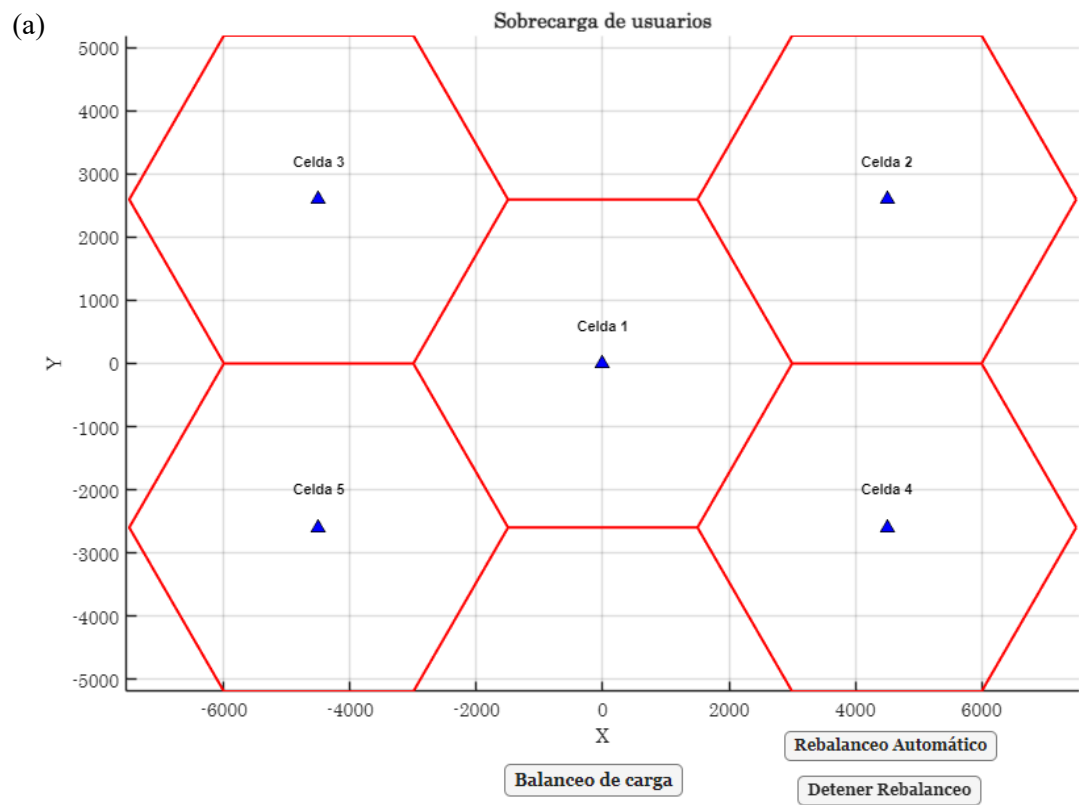
Tabla 12. Resumen de las métricas de rendimiento para el escenario de 15 celdas.

| Métrica | Valor promedio | Valor Mínimo | Valor Máximo |
|---|---------------------------|-------------------------|-------------------------|
| Throughput antes de aplicar el algoritmo (Mbps) | 38.02 | 32.91 | 44.24 |
| Throughput después de aplicar el algoritmo (Mbps) | 53.56 | 48.09 | 60.73 |
| Número de Handovers | 44 | 22 | 66 |
| Equidad en la distribución de la carga (índice de Jain) | 0.90 | 0.82 | 0.96 |

El número promedio de handovers fue de 44, lo cual es un valor razonable considerando el número de celdas de la topología. En cuanto al valor promedio del índice de Jain se observa que alcanza un valor de 0.90, teniendo una pequeña diferencia con los dos escenarios anteriores, lo que demuestra un buen desempeño del algoritmo incluso en condiciones de mayor complejidad.

4.8.2 Análisis gráfico de cada escenario

Las gráficas del sistema son generadas individualmente para cada simulación en todos los escenarios propuestos. Debido al número total de simulaciones realizadas, se selecciona una de forma representativa por cada escenario, donde gráficamente se observa la asignación de usuarios, la trayectoria de los usuarios al realizar el handover y la comparación del throughput antes y después de aplicar el algoritmo de handover horizontal.



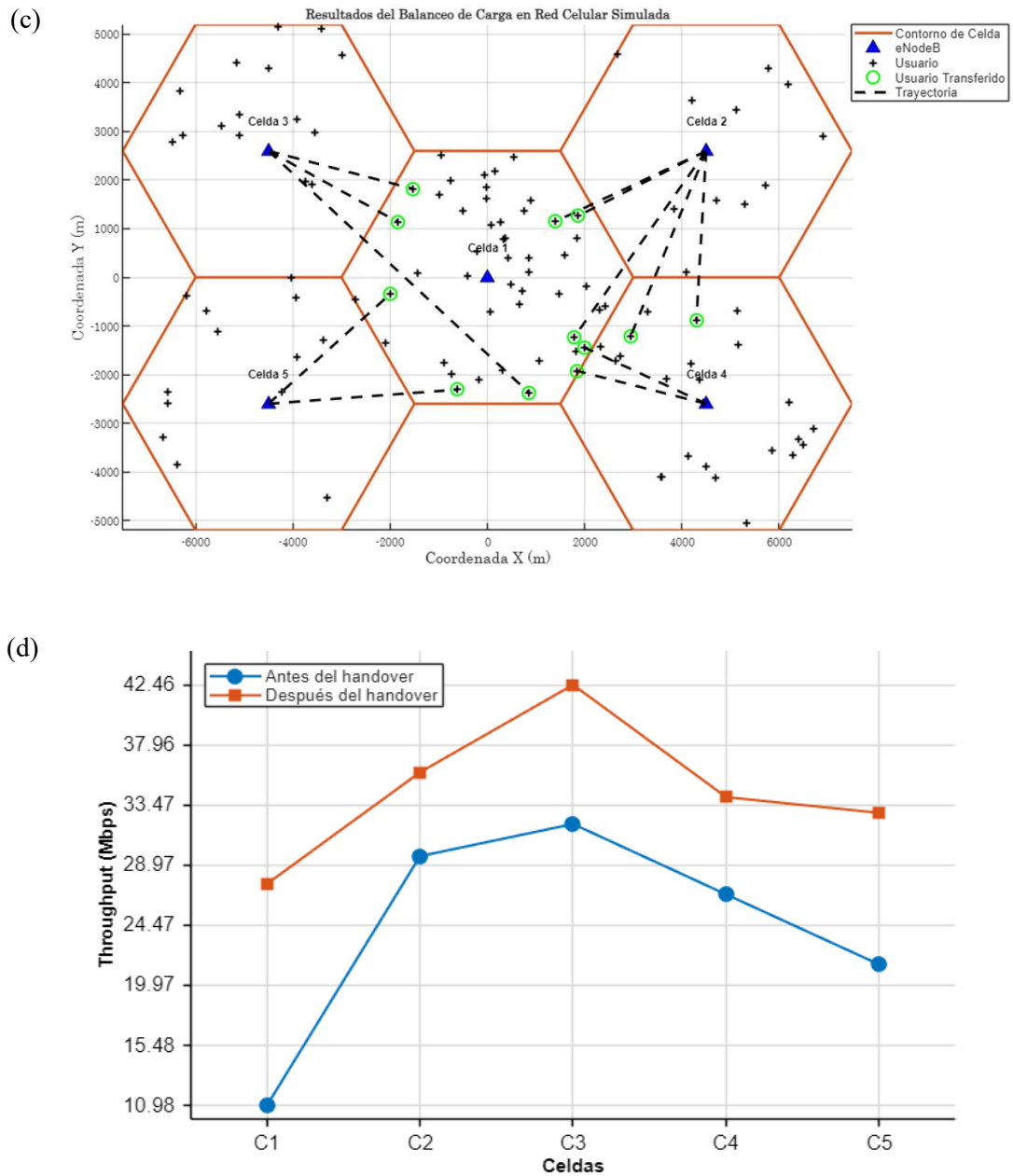
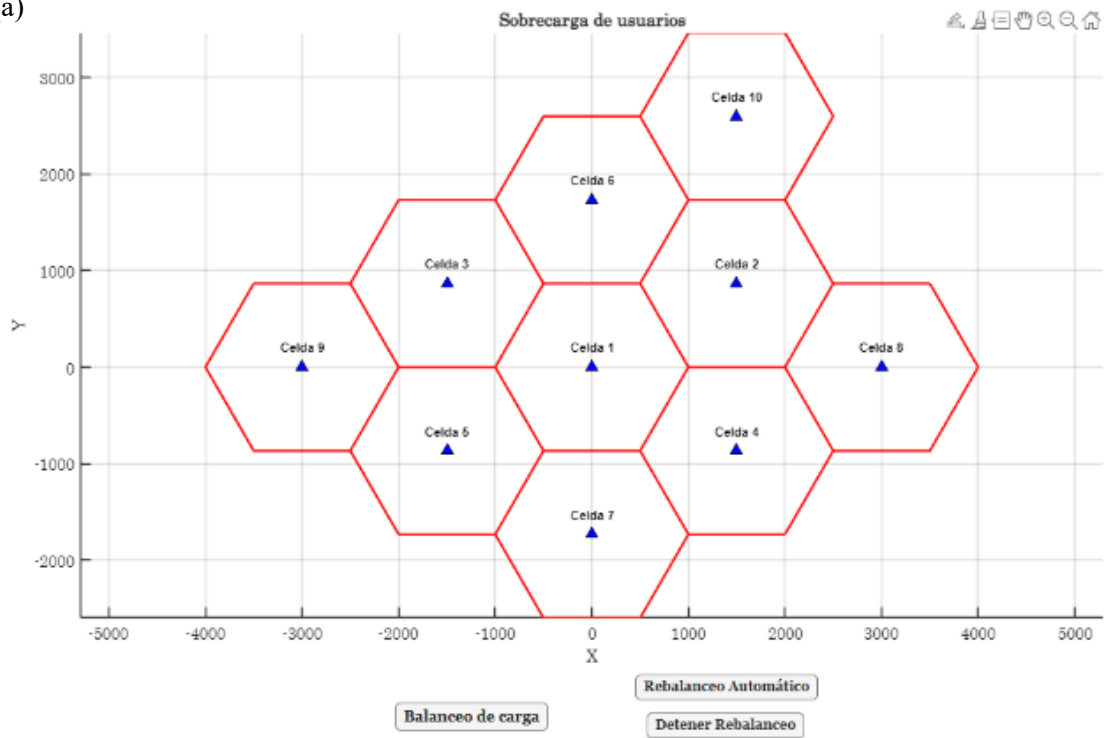
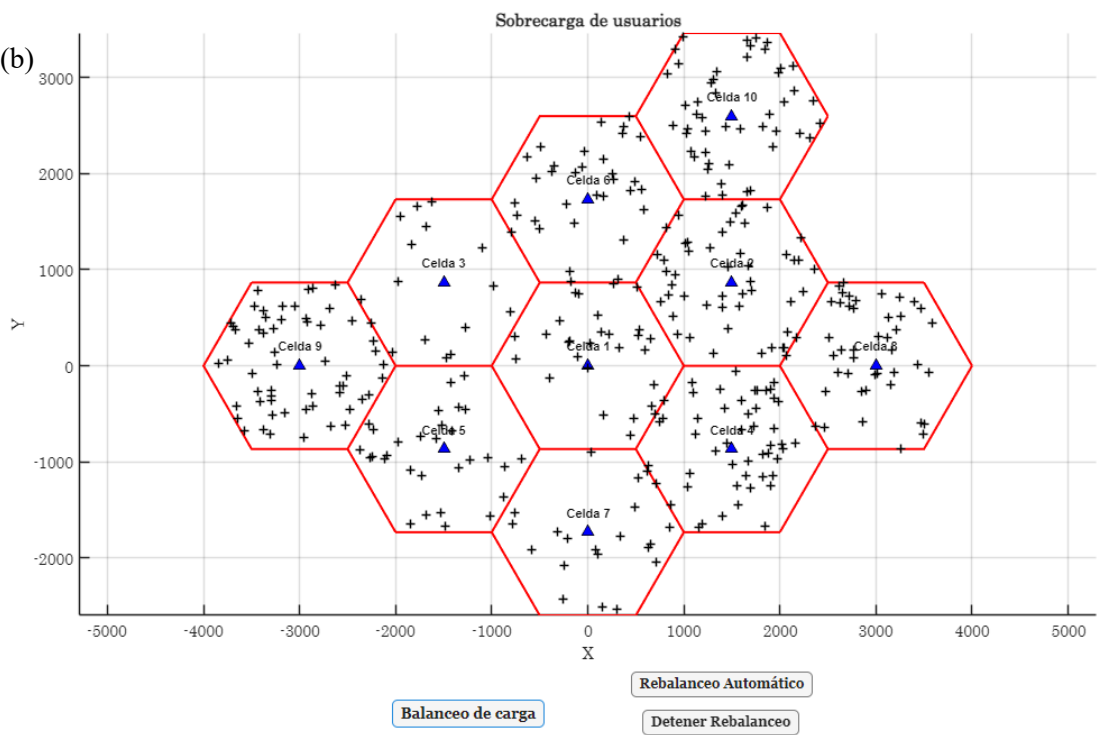


Figura 9. Visualización del sistema de simulación: (a) Topología de red con 5 celdas (RMa). (b) Generación y distribución de usuarios. (c) Ejecución del handover y trayectoria de los usuarios transferidos. (d) Comparación del throughput antes y después de aplicar el algoritmo.

(a)



(b)



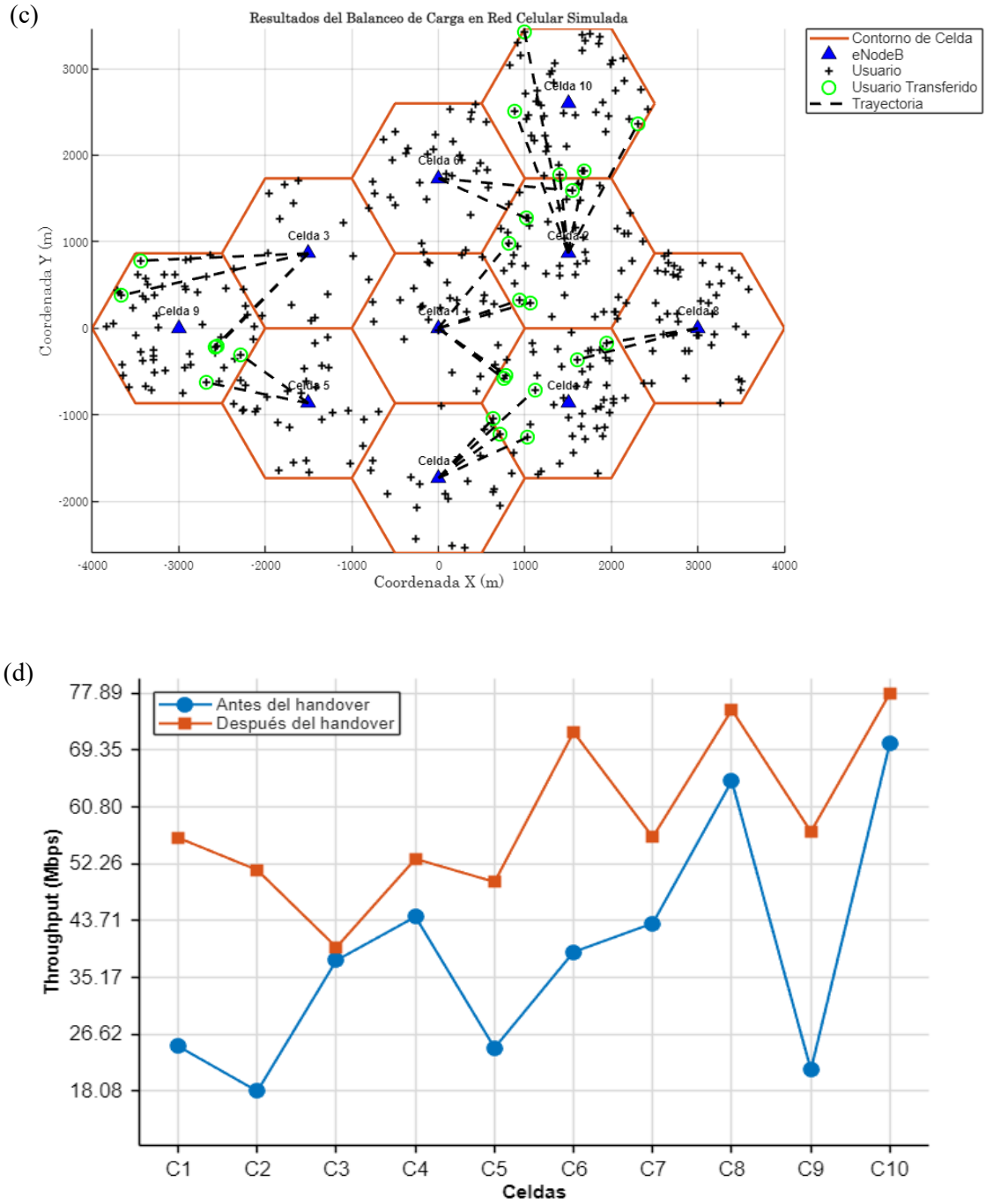
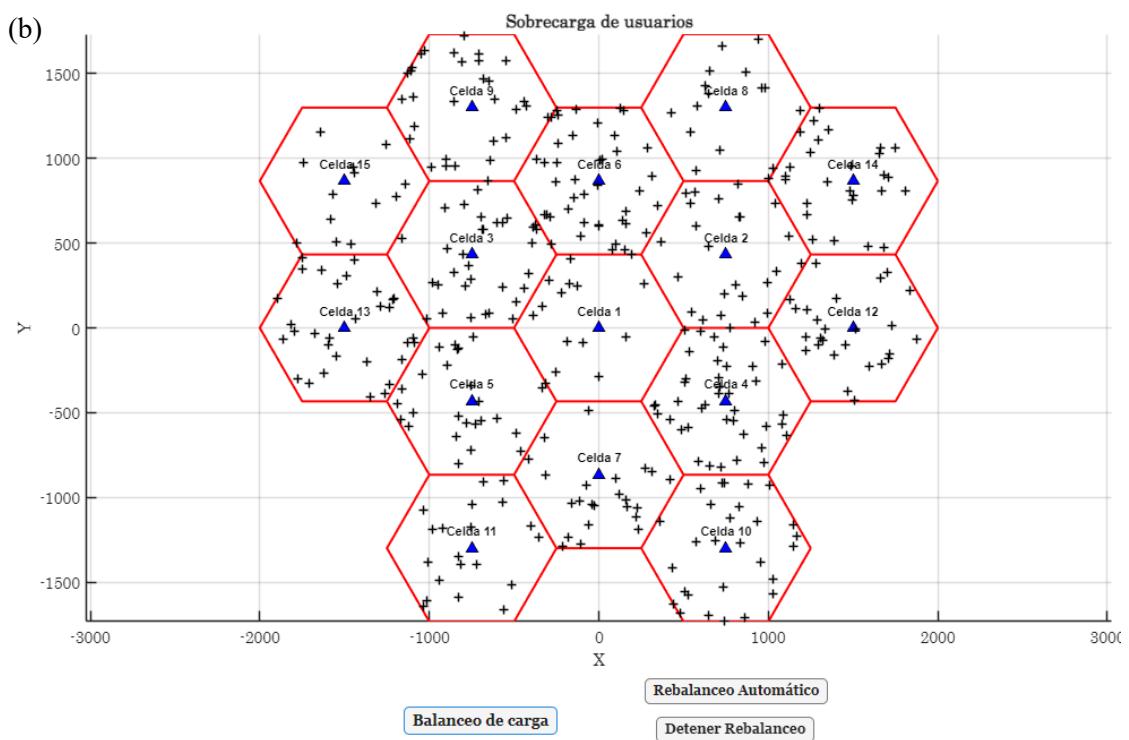
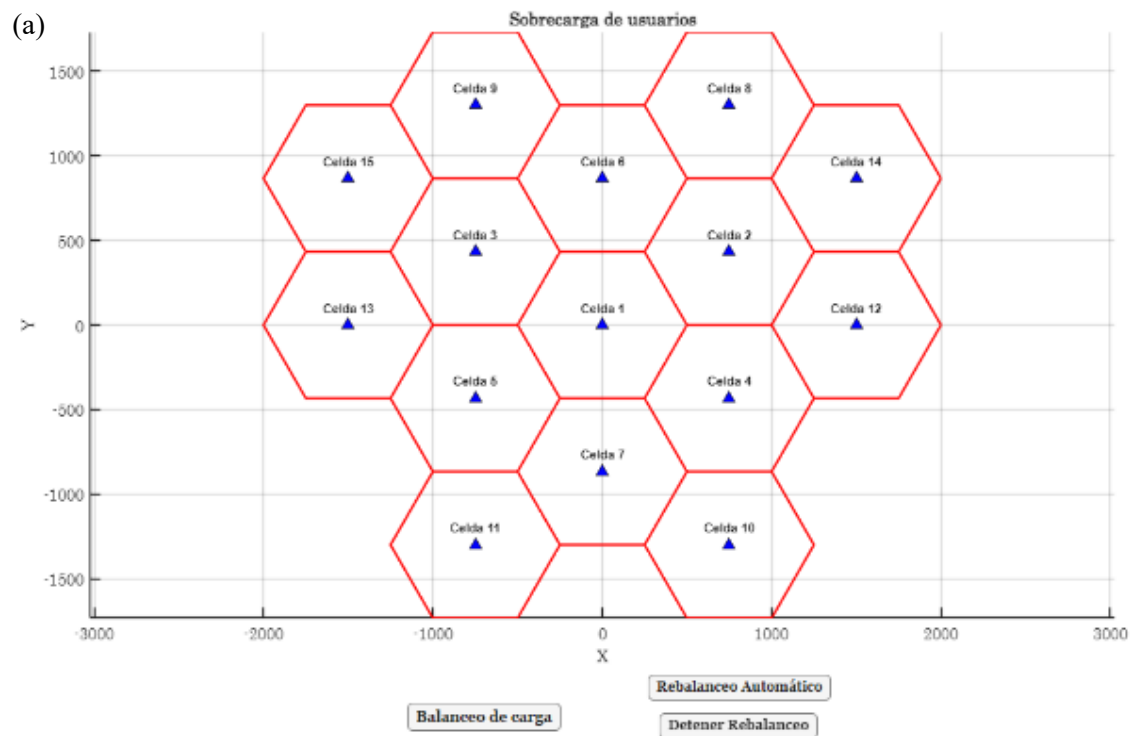


Figura 10. Visualización del sistema de simulación: (a) Topología de red con 10 celdas (UMa). (b) Generación y distribución de usuarios. (c) Ejecución del handover y trayectoria de los usuarios trasferidos. (d) Comparación del throughput antes y después de aplicar el algoritmo.



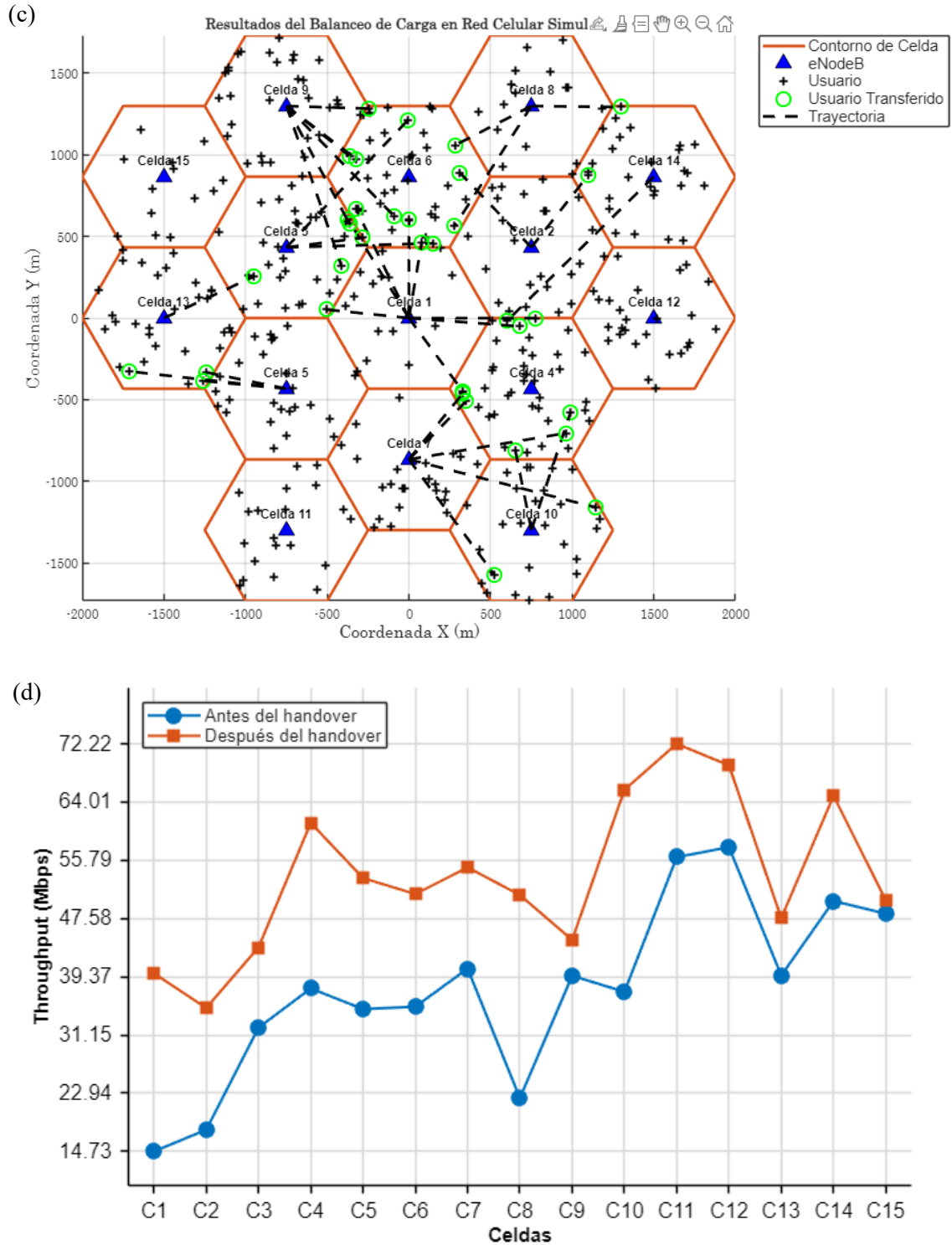


Figura 11. Visualización del sistema de simulación: (a) Topología de red con 15 celdas (UMi). (b) Generación y distribución de usuarios. (c) Ejecución del handover y trayectoria de los usuarios trasferidos. (d) Comparación del throughput antes y después de aplicar el algoritmo.

Las figuras correspondientes a los tres escenarios simulados permiten observar el comportamiento del sistema de forma gráfica. En cada caso, se muestran las posiciones de las celdas, según el entorno escogido desde la interfaz, la distribución y posición de los usuarios. Donde, a diferencia de las celdas, en cada simulación estos dos componentes varían para simular entornos con variabilidad de carga. Para entornos reales se considera que cuando un usuario realiza el proceso de handover, no realiza el traspaso de celda de forma física, es decir, el usuario no se mueve de su celda actual a la celda que realizó el handover. El cambio de celda se realiza en la conexión del dispositivo. Para ello, en la gráfica correspondiente, se proyecta una trayectoria desde el usuario hacia el eNodeB destino. Finalmente, se muestra la comparación del valor de throughput medido antes y después de aplicar el algoritmo de handover basado en Q-Learning.

4.8.3 Análisis comparativo entre escenarios

De acuerdo con los resultados que se muestran en la Tabla 13, el sistema fue capaz de mejorar el throughput global a medida que aumenta el número de celdas.

Tabla 13. Comparación de métricas promedio por escenario.

| Escenario | Throughput antes de aplicar el algoritmo (Mbps) | Throughput después de aplicar el algoritmo (Mbps) | Mejora (%) | Handovers | Índice de Jain |
|-----------|--|--|---------------|-----------|----------------------|
| 5 celdas | 29.51 | 36.57 | 23.92 | 11 | 0.91 |
| 10 celdas | 45.68 | 59.86 | 31.03 | 29 | 0.91 |
| 15 celdas | 38.02 | 53.56 | 40.86 | 44 | 0.90 |

Esto se evidencia en la mejora porcentual del throughput promedio obtenido en el escenario de 15 celdas, donde se alcanza una mejora del 40.86%, siendo superior a los otros dos escenarios. Con esto, se demuestra la habilidad del algoritmo de adaptarse en entornos más complejos. El número de handovers es otra métrica que aumenta en relación con el entorno, ya que la red al tener una topología con mayor número de celdas ofrece una mayor capacidad de aceptación de usuarios, lo que incrementa la posibilidad de que un usuario de acuerdo con las condiciones que presente realice el handover de una celda a otra vecina. El índice de Jain en todos los escenarios es igual o superior a 0.90, demostrando que el sistema no solo se encarga de mejorar el rendimiento de la red, a su vez logra preservar una distribución de carga balanceada.

4.8.4 Análisis de la distribución del throughput

En esta figura se presenta un análisis comparativo del throughput mediante diagramas de caja.

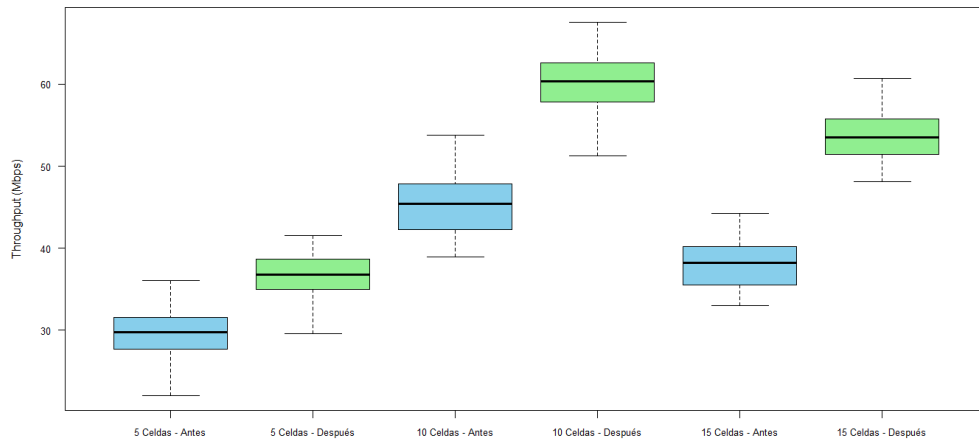


Figura 12. Distribución del throughput antes y después de aplicar el algoritmo de handover en escenarios con 5, 10 y 15 celdas

Para los 3 escenarios, se muestra una mejora en el estado Después, lo que indica un aumento en el throughput después de aplicar el algoritmo de handover horizontal basado en Q-Learning. A pesar de que en el escenario de 10 celdas muestra el valor más alto del throughput, es el escenario de 15 celdas el que presenta una mayor mejora en términos porcentuales, como se describe en la Figura 12. Mostrando así, la efectividad del algoritmo en escenarios densos, donde las condiciones de la red son más inestables por la interferencia entre un mayor número de celdas y los obstáculos propios del entorno.

4.8.5 Prueba de hipótesis

Para determinar si el algoritmo de handover horizontal basado en Q-Learning mejora la calidad de servicio en una red 4G LTE, se realizó el análisis de los 3 escenarios con un número de muestras de 60 simulaciones para cada uno. Debido a que el tamaño de la muestra es moderado, se aplicó la prueba Shapiro-Wilk como el método estadístico para demostrar que los datos de las diferencias del throughput entre los valores obtenidos antes y después de aplicar el algoritmo tienen una distribución normal. Las hipótesis para realizar las pruebas de normalidad a cada escenario son las siguientes:

Hipótesis nula (H_0)

Los datos tienen una distribución normal.

Hipótesis alternativa (H_a)

Los datos no tienen una distribución normal.

En la Tabla 14, se observa que, en el análisis de normalidad se acepta la hipótesis nula. Los datos de las diferencias del throughput entre los valores obtenidos antes y después de aplicar el algoritmo tienen una distribución normal (Valor-p > 0.05).

Tabla 14. Resultados de la prueba de normalidad.

| Escenario | Estadístico W | Valor-P | Decisión |
|-----------------|---------------|---------|--|
| RMa (5 celdas) | 0.9784 | 0.3647 | Se acepta la hipótesis nula. Los datos tienen una distribución normal. |
| UMa (10 celdas) | 0.9922 | 0.968 | |
| UMi (15 celdas) | 0.9883 | 0.8393 | |

Una vez verificado que los datos tienen una distribución normal. Se procedió a establecer las hipótesis correspondientes con el objetivo de determinar si la implementación del algoritmo de handover horizontal basado en Q-Learning mejora la calidad de servicio en una red 4G LTE. Las hipótesis se presentan a continuación:

Hipótesis nula (H_0)

El algoritmo de handover horizontal basado en aprendizaje por refuerzo no incrementa significativamente el throughput promedio, y por lo tanto no mejora la Calidad de Servicio (QoS) en la red 4G LTE.

Hipótesis alternativa (H_a)

El algoritmo de handover horizontal basado en aprendizaje por refuerzo incrementa significativamente el throughput promedio, lo que implica una mejora en la Calidad de Servicio (QoS) en la red 4G LTE.

Luego se aplicó la prueba t de Student para muestras pareadas. Los resultados se muestran en la Tabla 15.

Tabla 15. Análisis T de Student para muestras pareadas.

| Escenario | Valor t | Valor-p | Decisión |
|-----------------|---------|---------|--|
| RMa (5 celdas) | 15.034 | < 0.001 | Se rechaza la hipótesis nula. El algoritmo de handover horizontal basado en aprendizaje por refuerzo incrementa significativamente el throughput promedio, lo que implica una mejora en la Calidad de Servicio (QoS) en la red 4G LTE. |
| UMa (10 celdas) | 25.266 | < 0.001 | |
| UMi (15 celdas) | 31.869 | < 0.001 | |

Mediante los resultados obtenidos se pudo demostrar que las mejoras registradas son estadísticamente significativas y se valida la hipótesis alternativa. Además, este proceso respalda el enfoque de este trabajo, el cual logra, de manera satisfactoria la distribución de usuarios en las celdas de acuerdo con las características de su entorno y la mejora que brinde el proceso de traspaso en las métricas medidas.

Comparación del throughput entre escenarios mediante el método ANOVA

Para verificar que los datos cumplan con el supuesto de homocedasticidad se aplicó la prueba de Levene y se obtuvo un Valor P de 0.4511, siendo superior al nivel de significancia ($\alpha = 0.05$), lo que indicó que no existen diferencias significativas entre varianzas de los grupos

de datos. Por lo tanto, se considera adecuado aplicar el método estadístico ANOVA para determinar si la mejora del throughput experimentada después de la aplicación del algoritmo de handover horizontal basado en Q-Learning varía significativamente según el número de celdas de la red. Como ya fue verificado la distribución normal de los datos anteriormente. En la Tabla 16 se presentan los resultados que se obtuvieron tras aplicar el método estadístico ANOVA.

Tabla 16. Resultados del ANOVA aplicado a los datos del throughput.

| Variación | Grados de Libertad | Suma de cuadrados | Cuadrado medio | Estadístico F | Valor P |
|-----------|--------------------|-------------------|----------------|---------------|---------|
| Escenario | 2 | 2485 | 1242.4 | 80.35 | <2e-16 |
| Error | 177 | 2737 | 15.5 | | |

Como el ANOVA demostró que existen diferencias en la mejora del throughput entre escenarios, se aplicó la prueba de comparaciones múltiples de Tukey HSD para determinar entre que escenarios existen diferencias significativas después de la implementación del algoritmo de handover.

Tabla 17. Comparación de la mejora del throughput entre escenarios mediante la prueba de Tukey HSD.

| Comparación | Diferencia de medias | IC 95% | Valor P | Conclusión |
|-----------------|----------------------|----------------|---------|------------------|
| 15 vs 10 celdas | +1.36 | [-0.34, 3.05] | 0.144 | No significativa |
| 5 vs 10 celdas | -7.11 | [-8.81, 5.42] | < 0.001 | Significativa |
| 5 vs 15 celdas | -8.47 | [-10.17, 6.78] | < 0.001 | Significativa |

Los resultados de la Tabla 17 muestran que la mejora del throughput en el escenario de 5 celdas fue menor en comparación de los escenarios de 10 y 15 celdas. No obstante, no se evidenció una diferencia significativa entre ambos escenarios. Este comportamiento demuestra que el algoritmo es más efectivo cuando la red tiene un mayor número de celdas.

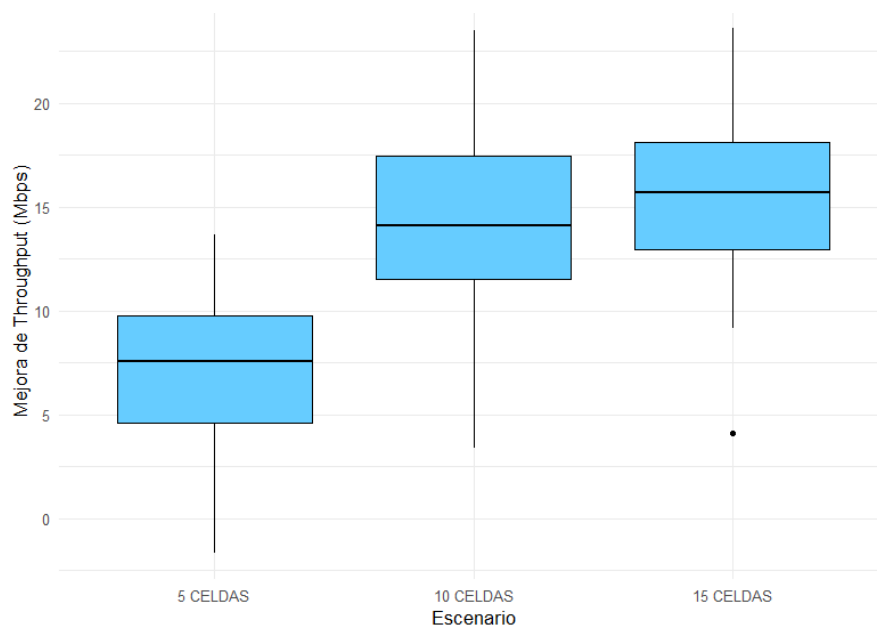


Figura 13. Mejora del throughput por escenario.

En la Figura 13 se presenta la distribución de las mejoras individuales del throughput para cada escenario. Cuando hay un mayor número de celdas se presentan mejoras más altas en comparación con el escenario de menor número de celdas. Este comportamiento valida los resultados obtenidos al aplicar el método ANOVA.

Análisis de índice de Jain

Se realizó el análisis del índice de Jain para evaluar si el algoritmo de handover horizontal basado en el Q-Learning mejora significativamente el balanceo de carga en la red 4G LTE. En primer lugar, se aplicó la prueba de normalidad para determinar si los datos antes y después de aplicar el algoritmo presentan una distribución normal.

Tabla 18. Resultados de la prueba de normalidad para el análisis del índice de Jain.

| Escenario | Estadístico W | Valor-P | Decisión |
|-----------------|---------------|---------|--|
| RMa (5 celdas) | 0.9695 | 0.1461 | Se acepta la hipótesis nula. Los datos tienen una distribución normal. |
| UMa (10 celdas) | 0.9734 | 0.2152 | |
| UMi (15 celdas) | 0.9460 | 0.010 | Se rechaza la hipótesis nula. Los datos no tienen una distribución normal. |

Para los escenarios de 5 y 10 celdas se utilizó la prueba de T de Student para muestras pareadas, mientras que para el escenario de 15 celdas al no tener una distribución normal se usa la prueba no paramétrica de Wilcoxon.

Tabla 19. Análisis T de Student para muestras pareadas y prueba no paramétrica de Wilcoxon.

| Escenario | Prueba aplicada | Valor t | Valor-p | Decisión |
|-----------------|-------------------------|---------|---------|--|
| RMa (5 celdas) | T de Student | 11.459 | < 0.001 | Se rechaza la hipótesis nula. El índice de Jain mejora significativamente después de aplicar el algoritmo de handover horizontal basado en Q-Learning. |
| UMa (10 celdas) | T de Student | 11.823 | < 0.001 | |
| UMi (15 celdas) | No paramétrica Wilcoxon | 1830 | < 0.001 | |

Los resultados de la Tabla 19 muestran que en todos los escenarios el algoritmo incrementa significativamente el índice de Jain, evidenciando una mejor equidad en la distribución de carga entre las celdas. A pesar de que en el escenario de 16 celdas se utilizó una prueba no paramétrica, se validó correctamente la mejora de la equidad en la distribución de carga.

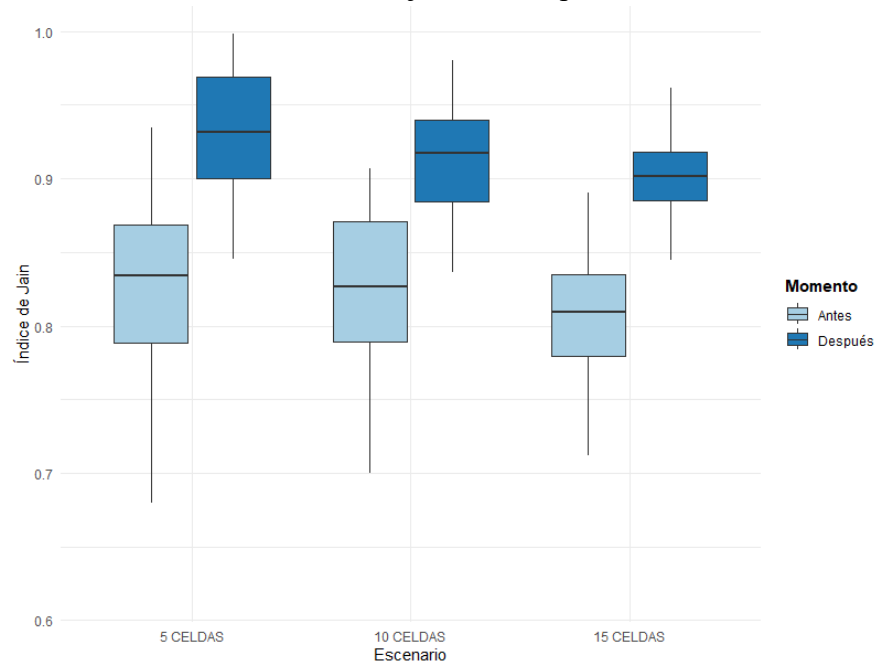


Figura 14. Comparación del antes y después del índice de Jain.

En la Figura 14 se observa un incremento del índice de Jain en todos los escenarios, lo que resulta en una mejora en la equidad de la distribución de carga entre las celdas. A su vez, después del aplicar el algoritmo de handover, la dispersión presenta una reducción considerable en el escenario de 15 celdas lo que evidencia la estabilidad del sistema.

4.9 Discusión

Los resultados obtenidos en los tres escenarios propuestos evidencian la efectividad del algoritmo de handover horizontal basado en Q-Learning propuesto bajo diferentes condiciones, ya sea en el número de celdas, la distribución de usuarios y la carga de tráfico. Usando métricas como el throughput, el número de handovers que realiza el sistema y el índice de Jain para medir la distribución de carga, se evidenció que, tras ejecutar el algoritmo, el rendimiento de la red mejora significativamente.

Los resultados obtenidos en el throughput promedio establecieron una base para realizar la comparación antes y después de aplicar el algoritmo. De este modo, se obtuvieron las mejores porcentuales de cada escenario, registrando así una mejora del 23.92% en el escenario de 5 celdas, mientras que para el escenario de 10 celdas se registró una mejora del 31.03% y para el escenario de 15 celdas se registró una mejora del 40.86%. Como se observa, las mejoras porcentuales presentan un patrón creciente: mientras más grande es la topología de la red y aumenta la complejidad del entorno, el algoritmo sigue siendo funcional y hasta demuestra ser escalable.

La equidad en la distribución de carga entre las celdas es medida por medio del índice de Jain. Para todos los escenarios se mantuvo en un valor promedio igual o superior a 0.90, evidenciando no solo mejoras en el throughput, sino también en el comportamiento global del sistema, ya que las decisiones que toma el algoritmo para el cambio de celda manejan distintas condiciones para que el traspaso de una celda a otra sea lo más efectivo posible. Los resultados estadísticos permiten afirmar que la aplicación del algoritmo de handover produce una mejora significativa tanto en el throughput como en la equidad de la red en todos los escenarios propuestos. Es importante destacar que la comparación entre escenarios evidenció que el escenario de 5 celdas tiene una mejora modesta en comparación con el escenario de 15 celdas, donde sus resultados mostraron mayor incremento en la equidad. En cuanto al número de handovers que presenta cada escenario, se reflejan valores coherentes para descongestionar las celdas sobrecargadas. Para obtener todos estos resultados favorables y guiar el aprendizaje a políticas equilibradas, el desarrollo de la función de recompensa que se aplicó al algoritmo fue fundamental. De este modo se demuestra que el algoritmo contribuye de manera efectiva en mejorar la Calidad de Servicio (QoS) en redes 4G LTE.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

En la presente investigación, se desarrolló un modelo de simulación de una red 4G LTE y se implementó un algoritmo de handover horizontal basado en aprendizaje por refuerzo. A través de una arquitectura estructurada donde se incluyen elementos como la selección del entorno por medio del número de celdas, distribución de usuarios, asignación de tráfico, asignación de estados y recompensas para el algoritmo Q-Learning y el cálculo de métricas. Se obtuvo un sistema que, mediante la toma de decisiones según su entorno, realiza el proceso de handover entre celdas vecinas con el objetivo de mejorar, no solo el balanceo de carga, sino, además, la calidad de servicio (QoS).

El desarrollo del modelo se sustentó en una revisión detallada del estado de arte, la cual permitió identificar las principales estrategias utilizadas para gestionar de forma efectiva el balanceo de carga y el proceso de handover. A través de la revisión de 90 publicaciones científicas, tesis y artículos académicos, se seleccionaron 57 que presentaban información relevante sobre la aplicación de algoritmos tradicionales, heurísticos e inteligentes en redes móviles. Este análisis permitió evidenciar las limitaciones que presentan algunas investigaciones acerca de escenarios más completos y evidenciar fragmentos de códigos, parámetros, condiciones o políticas de entrenamiento en algoritmos inteligentes. En esta etapa se seleccionaron las mejores técnicas para aplicar el algoritmo de handover horizontal basado en Q-Learning y como resultado conseguir una mejora global del sistema.

Los resultados obtenidos mediante las 60 simulaciones realizadas a los tres escenarios reflejan mejoras significativas en las métricas presentadas. En el escenario de 5 celdas, después de aplicar el algoritmo, se alcanza una mejora promedio del 23.92% en el valor del throughput, mientras que, en el escenario de 10 celdas, la mejora promedio alcanza el 31.03%, para el escenario de 15 celdas, la mejora promedio llega al 40.86%. Se calcula también la equidad en la distribución de los recursos entre celdas mediante el índice de Jain, esta métrica para los tres escenarios se obtuvieron valores iguales o mayores a 0.90. Confirmando así que el algoritmo tiene la capacidad de adaptarse desde redes pequeñas a redes más complejas con condiciones menos favorables. El número de handovers realizados también incrementa según el escenario. Sin embargo, esto varía de acuerdo con las condiciones iniciales de la red, ya que, si el sistema inicia con condiciones donde el tráfico de carga está distribuido de forma equitativa, el número de handovers va a ser menor.

Una de las características principales de este sistema es que, mediante el análisis del delta máximo en la Q tabla, se demostró la convergencia estable del proceso de aprendizaje. El valor para los tres escenarios se mantuvo por debajo del 0.025, demostrando así que las decisiones que toma el agente son resultado del aprendizaje progresivo que se estabiliza mediante la experiencia adquirida. Además, por medio de la implementación de la estrategia

ϵ -greedy con decaimiento, el agente balancea adecuadamente la exploración con la explotación sin necesidad de aplicar otros métodos de aceleración de aprendizaje.

A pesar de que el sistema usa una función de recompensa estática combinando los parámetros del throughput, carga y equidad, se generó un comportamiento balanceado. De acuerdo con todos los resultados obtenidos, se evidencia que el uso de algoritmos basados en aprendizaje por refuerzo, en este caso la aplicación de Q-Learning, es una alternativa efectiva y mejorada frente a métodos tradicionales en donde la mejora es aislada. También para reforzar los resultados, el sistema validó la hipótesis planteada. Por lo tanto, este trabajo se presenta como base para futuras investigaciones en redes más avanzadas.

5.2 Recomendaciones

A partir de la investigación realizada y los resultados obtenidos, se recomienda mejorar las condiciones del sistema, específicamente los estados y acciones del algoritmo de aprendizaje por refuerzo. En la metodología actual, los estados están discretizados utilizando los parámetros de SINR y carga, siendo efectivos para cumplir con los objetivos propuestos. Sin embargo, para que el algoritmo sea escalable y mejore su adaptabilidad con el entorno, resultaría conveniente realizar estudios donde se incorporen otras variables, como la velocidad de usuario o el tipo de servicio. A partir de la incorporación de estas variables se podría lograr que el algoritmo tome mejores decisiones con una visión más completa de la red.

En cuanto a la función de recompensa, se comprobó que la combinación del throughput, la carga y la equidad es efectiva para guiar al agente a políticas equilibradas. A pesar de que los valores de entrada a la función de recompensa varían según el entorno, esta función es considerada estática, debido a que su estructura es la misma en los tres escenarios propuestos. Se recomienda que, para futuras investigaciones, se estudien funciones dinámicas, donde su estructura se ajuste de acuerdo con el tipo de escenario y las condiciones generales de la red, logrando así desarrollar un agente más inteligente.

El sistema, para el proceso de aprendizaje, implementa una estrategia ϵ -greedy, la cual permite que el agente realice una exploración al inicio del entrenamiento y gradualmente vaya consolidando una política estable en función de las experiencias acumuladas. A pesar de que por medio de este mecanismo el sistema tiene una convergencia eficiente, el experimentar otras técnicas permitiría una adaptación más rápida del sistema ante picos de tráfico. De este modo, se lograría realizar un análisis sobre las estrategias implementadas y realizar una comparación para tener información sólida que sería de gran relevancia para otras aplicaciones.

BIBLIOGRAFÍA

- [1] L. Carias, Z. Álvarez, y E. Guerra, «Telefonía móvil celular», Trabajo académico, Universidad Nacional Experimental de la Fuerza Armada Nacional, Puerto Piritu, 2021.
- [2] C. Flores, «Estudio de la infraestructura y el espectro radioeléctrico en la evolución de la tecnología 4G y su convergencia a 5G en redes de telefonía móvil en el Ecuador», Trabajo de grado, Universidad Técnica del Norte, Ibarra, 2022.
- [3] J. Villareal, «Análisis del comportamiento del handover en redes celulares LTE mediante modelos de clasificación por árboles de decisión y KNN», Trabajo de integración curricular, Escuela Politécnica Nacional, Quito, 2023.
- [4] C. Martín, «Rendición de cuentas 2023», Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2024. [En línea]. Disponible en: <https://www.telecomunicaciones.gob.ec/rendicion-de-cuentas-2023/>
- [5] K. Quezada, «Análisis comparativo de sistemas de balanceo de carga aplicados a redes móviles de última generación», Maestría en Telecomunicaciones, Universidad Nacional de Loja, Loja, 2023.
- [6] V. Párraga, P. Lupera, y F. Grijalva, «Handover en redes celulares: Machine Learning», Avances de investigación de doctorado, Escuela Politécnica Nacional, XXXI Jornadas en Ingeniería Eléctrica y Electrónica, Quito, 2023.
- [7] E. Udo, U. Oleh, y G. Odo, «Analysis and computer simulation of handover decision in 4G networks», *NIPES Journal of Science and Technology Research* 3(4), pp. 97-108, 2021, doi: <https://doi.org/10.37933/nipes/3.4.2021.10>.
- [8] J. Cervantes, «Protocolo CROSS-LAYER proactivo basado en técnicas de inteligencia artificial para handover sin fisuras en ambientes móviles WLAN», Tesis de grado, Universidad Autónoma del Estado de México, Texoco, 2020.
- [9] B. Caisaguano, «Simulación de un algoritmo predictivo para la toma de decisiones en vertical HANDOVER para redes 4G LTE y Satelitales», Trabajo de titulación, Universidad Nacional de Chimborazo, Riobamba, 2025.
- [10] L. García, «Propuesta para mejorar la cobertura en interiores en redes de telefonía celular en clientes individuales», Trabajo de titulación, Universidad Católica de Santiago de Guayaquil, Guayaquil, 2019.
- [11] T. de Brito, «Machine learning based handover management for LTE networks with Coverage Holes», Tesis de maestría, Universidad Federal de Río Grande del Norte, Natal, 2018.
- [12] «Los beneficios que trae la Tecnología 4G», Ministerio de Telecomunicaciones y de la Sociedad de la Información. [En línea]. Disponible en: <https://www.telecomunicaciones.gob.ec/los-beneficios-que-trae-la-tecnologia-4g/>
- [13] ARCOTEL, «Detalle de Líneas Activas por tipo de servicio y Densidad del SMA». 2024. [En línea]. Disponible en: <https://www.arcotel.gob.ec/lineas-activas/>
- [14] ARCOTEL, «Servicio Móvil Avanzado. Líneas activas por tecnología». 2024. [En línea]. Disponible en: <https://www.arcotel.gob.ec/lineas-activas/>
- [15] M. Marí, «Optimización de la calidad de experiencia en redes celulares mediante el ajuste del traspaso entre celdas», Tesis doctoral, Universidad de Málaga, Málaga, 2021.
- [16] J. Villareal, «Estudio del handover en redes LTE mediante mediciones de campo», Trabajo de integración curricular, Escuela Politécnica Nacional, Quito, 2023.

- [17] C. Casero, «Diseño radio orientado a la implementación de LNR700 y análisis de las degradaciones más comunes», Trabajo de fin de grado, Universidad Politécnica de Valencia, Gandia, 2024.
- [18] S. Hatipoglu, B. Mehmet, M. Akif Yaz, y L. Ata Durak, «Handover-based Load Balancing Algorithm for 5G and Beyond Heterogeneous Networks», presentado en Congreso Internacional sobre Sistemas y Talleres Ultra Modernos de Telecomunicaciones y Control (ICUMT), 2020. doi: 10.1109/ICUMT51630.2020.9222456.
- [19] J. Ramírez, «Mejoramiento del modelo basado en MLB (Mobility Load Balancing) sobre redes LTE», Tesis de maestría, Universidad Distrital Francisco José de Caldas, Bogotá, 2020.
- [20] R. Espinosa, P. Lupera, V. Farre, R. Maldonado, y R. Llugsí, «Statistical Analysis of Handover Process Performance in a Cellular Mobile Network in the City of Quito, Ecuador», presentado en XXXI Conference on Electrical and Electronic Engineering, Quito: Electronics, Telecommunications, and Information Network Department, Escuela Politécnica Nacional, 2023. doi: <https://doi.org/10.3390/engproc2023047019>.
- [21] K. Martínez, «Evaluación de parámetros de optimización de sistemas de balanceo de carga para redes móviles basado en contexto social», Trabajo de titulación, Universidad Nacional de Loja, Loja, 2023.
- [22] V. Párraga, C. Rocha, H. Navarrete, y P. Lupera, «Modelos Predictivos de Zonas de Handover en Redes LTE con Base a Mediciones de Campo y Árboles de Decisión (Caso de Estudio Ciudad de Quito)», *Revista Politécnica*, vol. 52, n.º 1, 2023, doi: <https://doi.org/10.33333/rp.vol52n1.02>.
- [23] V. Párraga, P. Lupera, F. Grijalva, y H. Carvajal, «Brief Survey: Machine Learning in Handover Cellular Network», presentado en XXXI Conference on Electrical and Electronic Engineering, 2023. doi: <https://doi.org/10.3390/engproc2023047002>.
- [24] C. Parreño, P. Lupera, R. Llugsí, y V. Párraga, «Optimized Performance Evaluation of a Q-Learning Hard Handover Algorithm for Load Balancing», *IEEE Microwave Theory and Techniques in Wireless Communications (MTTW)*, pp. 74-79, 2021, doi: 10.1109/MTTW53539.2021.9607052.
- [25] V. Yajnanarayana, H. Rydén, y H. László, «5G Handover using Reinforcement Learning», presentado en IEEE 3rd 5G World Forum (5GWF), 2020, pp. 349-354. doi: 10.1109/5GWF49715.2020.9221072.
- [26] M. Mollel *et al.*, «Intelligent handover decision scheme using double deep reinforcement learning», *University of Glasgow*, vol. 42, n.º 101133, pp. 1874-4907, 2020, doi: 10.1016/j.phycom.2020.101133.
- [27] G. Delin, T. Lan, Z. Xinggan, y L. Ying-Chang, «Joint Optimization of Handover Control and Power Allocation Based on Multi-Agent Deep Reinforcement Learning», *IEEE Transactions on Vehicular Technology*, vol. 69, n.º 11, pp. 13124-13138, 2020, doi: 10.1109/TVT.2020.3020400.
- [28] J. Tianqi, Y. Huiyuan, y X. Zhang, «A Reinforcement Learning Handover Parameter Adaptation Method Based on LSTM-Aided Digital Twin for UDN», *Self-Organized Computing and Network Management for Intelligent Internet of Things*, vol. 23, n.º 4, 2023, doi: <https://doi.org/10.3390/s23042191>.
- [29] S. Nayak, «ns-3 Simulation Based Exploration of LTE Handover Optimization», *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 7, n.º 4, 2023, doi: <https://doi.org/10.4108/eetmca.v7i4.2967>.

- [30] Q. Liu, C. Foong Kwong, W. Sun, L. Li, y H. Zhao, «Reinforcement Learning based Adaptive Handover in Ultra-Dense Cellular Networks with Small Cells», *International Symposium on Artificial Intelligence and Robotics*, vol. 11574, 2020.
- [31] «Características técnicas y ventajas de la red LTE 4G». Universidad Internacional de Valencia, 2023. [En línea]. Disponible en: <https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-lte-y-por-que-difiere-del-4g>
- [32] «LTE: qué es, cómo funciona y diferencias con 4G y 5G». La Universidad Internacional de La Rioja en Ecuador. Revista, 2025. [En línea]. Disponible en: <https://www.unir.net/revista/ingenieria/lte-que-es/>
- [33] K. Escudero, «Análisis y simulación a través de un modelo matemático para reducir el efecto Handover Ping-Pong en una red 4G», Trabajo de titulación, Universidad Nacional de Chimborazo, Riobamba, 2021.
- [34] J. Velez, «Mecanismos para el análisis de indicadores de red LTE», Tesis de maestría, Universidad Católica de Santiago de Guayaquil, Guayaquil, 2018.
- [35] A. Mrozowski, «Implementación del núcleo de red LTE/5G virtualizado», Trabajo de fin de grado, Universidad Politécnica de Valencia, Valencia, 2020.
- [36] R. Rosero, «Diseño de un sistema de recepción de señales LTE 4G utilizando radio definido por software, dirigido a la carrera de Ingeniería en Telecomunicaciones (CITEL) de la Universidad Técnica del Norte, para el desarrollo de prácticas de laboratorio de la asignatura de Radio Definido por Software (SDR)», Trabajo de grado, Universidad Técnica del Norte, Ibarra, 2023.
- [37] S. García, «Diseño e implementación de un prototipo de Estación Base (Nodo B) LTE, como herramienta experimental utilizando el SDR BladeRF», Trabajo de titulación, Universidad Nacional de Loja, Loja, 2023.
- [38] D. Molina, «Estudio de factibilidad para la implementación de la Tecnología LTE en la Banda de 700 MHz como complemento a la Banda de 2100 MHz para mejorar el Servicio de Internet Inalámbrico en la Zona Norte de la Ciudad de Quito», Tesis de maestría, Pontificia Universidad Católica del Ecuador, Quito, 2020.
- [39] «Basic Handover Types in Mobile Network». IPLOOK. End-to-End Mobile Network Solution. Company News. [En línea]. Disponible en: <https://www.iplook.com/info/basic-handover-types-in-mobile-network-i00362i1.html>
- [40] R. Acosta, «Análisis del algoritmo de handover reactivo y proactivo en sistemas femtocelda para la tecnología 4G (Long Term Evolution-Advanced)», Tesis de grado, Escuela Politécnica Nacional, Quito, 2018.
- [41] J. Labajo, «Diseño e implementación de un sistema de medida del throughput en redes móviles LTE», Trabajo de fin de grado, Universidad Politécnica de Madrid, Madrid, 2020.
- [42] M. Curipallo, G. Pozo, P. Lupera, y V. Párraga, «Modelo de aprendizaje automático para la predicción de calidad en modo estático-inactivo y modo móvil-conectado en redes LTE», *Revista de Investigación en Tecnologías de la Información*, vol. 10, n.º 21, 2022. [En línea]. Disponible en: <https://doi.org/10.36825/RITI.10.21.010>
- [43] S. Kazmierczyk, «Implementación de un sistema de análisis de medidas de cobertura en LTE mediante la herramienta QUALIPOC», Trabajo de fin de grado, Universidad Politécnica de Madrid, Madrid, 2020.
- [44] H. Gómez y N. Porras, «Implementación de un sistema de predicción de medidas LTE utilizando la tarjeta de desarrollo zedboard y machine learning», Tesis de grado, Universidad Distrital Francisco José de Caldas, Bogotá.

- [45] «Optimización de PRB en una red LTE con dos portadoras», Ekspres Consulting. [En línea]. Disponible en: <https://ekspres.com/es/optimizacion-de-prb-en-lte/>
- [46] D. Guadalupe, «Estudio y análisis de las tecnologías aplicadas en la interfaz aire de las comunicaciones móviles de 5G», Trabajo de titulación, Universidad Nacional de Loja, Loja.
- [47] A. Páez, «Estudio de la calidad de la conexión en redes de telefonía celular en base a mediciones de campo y técnicas de machine learning», Trabajo de integración curricular, Escuela Politécnica Nacional, Quito, 2023.
- [48] F. Osejo, «Análisis comparativo de algoritmos de machine learning aplicados en gestión de riesgos cuantitativos en proyectos predictivos y adaptativos», Tesis de grado, Pontificia Universidad Católica del Ecuador, Quito, 2023.
- [49] J. Guaita, «Diseño de un algoritmo adaptativo de backoff basado en aprendizaje reforzado para incrementar la probabilidad de acceso exitoso en una red celular con comunicaciones masivas», Trabajo de titulación, Universidad Nacional de Chimborazo, Riobamba, 2023.
- [50] P. Guamán, «Diseño de un algoritmo basado en aprendizaje reforzado para ajustar dinámicamente el tiempo de backoff en un entorno machine-type communication», Tesis de maestría, Escuela Superior Politécnica del Litoral, Guayaquil, 2023.
- [51] M. Rico, «Posicionamiento de estaciones base en tiempo real basado en aprendizaje de refuerzo profundo para futuras redes 6G», Trabajo de fin de grado, Universidad Politécnica de Valencia, 2023.
- [52] H. Casanova, «Diseño de un mecanismo de generación de actualizaciones dinámico utilizando aprendizaje reforzado en comunicaciones IoT celular para optimizar la frescura de la información», Trabajo de titulación, Universidad Nacional de Chimborazo, Riobamba.
- [53] R. Caponera, «Comparación de algoritmos de aprendizaje por refuerzo basados en Q-Learning», Trabajo de fin de grado, Universidad de Málaga. Escuela Técnica Superior de Ingeniería informática, Málaga, 2021.
- [54] K. Moya, «Aplicación del algoritmo de aprendizaje por refuerzo Q-Learning para la generación de trayectorias óptimas en plataformas robóticas», Trabajo de titulación, Universidad Técnica de Ambato, Ambato, 2024.
- [55] N. Maury y S. Villarruel, «Sistema multi-agente basado en Q-Learning para regular dinámicamente la dificultad de juegos de realidad virtual y mejorar la retención de los jugadores», Tesis de grado, Universidad Peruana de Ciencias Aplicadas, Lima.
- [56] 3GPP, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects (Release 9)*, Technical Report, 2010.
- [57] MathWorks, *MATLAB App Designer*. [En línea]. Disponible en: <https://es.mathworks.com/products/matlab/app-designer.html>

ANEXOS

Anexo 1. Códigos del sistema

En esta sección se agregan las principales funciones del sistema utilizadas para modelar la red 4G LTE y para ejecutar el proceso de aprendizaje por refuerzo mediante el algoritmo Q-Learning.

Modelado de entorno de red LTE

```
function posiciones = calcularPosicionesCeldas(app, numCeldas)
    r = app.radiosCelda(1);
    switch numCeldas
        case 5
            posiciones = [0, 0;
                          1.5*r, sqrt(3)*r/2;
                          -1.5*r, sqrt(3)*r/2;
                          1.5*r, -sqrt(3)*r/2;
                          -1.5*r, -sqrt(3)*r/2];
        case 10
            posiciones = [0, 0;
                          1.5*r, sqrt(3)*r/2;
                          -1.5*r, sqrt(3)*r/2;
                          1.5*r, -sqrt(3)*r/2;
                          -1.5*r, -sqrt(3)*r/2;
                          0, sqrt(3)*r;
                          0, -sqrt(3)*r;
                          3*r, 0;
                          -3*r, 0;
                          1.5*r, 1.5*sqrt(3)*r];
        case 15
            posiciones = [0, 0;
                          1.5*r, sqrt(3)*r/2;
                          -1.5*r, sqrt(3)*r/2;
                          1.5*r, -sqrt(3)*r/2;
                          -1.5*r, -sqrt(3)*r/2;
                          0, sqrt(3)*r;
                          0, -sqrt(3)*r;
                          1.5*r, 1.5*sqrt(3)*r;
                          -1.5*r, 1.5*sqrt(3)*r;
                          1.5*r, -1.5*sqrt(3)*r;
                          -1.5*r, -1.5*sqrt(3)*r;
                          3*r, 0;
                          -3*r, 0;
                          3*r, sqrt(3)*r;
                          -3*r, sqrt(3)*r];
    end
end
```

Script 26. Asignación de las posiciones a las celdas.

Distribución y posición de los usuarios

```
function nuevaPos = generarPosicionEnCelda(app, celdaIndex)
    r_scalar = app.radiosCelda(1); % Utilizamos el primer valor
    xCentro = app.cellPositions(celdaIndex, 1);
    yCentro = app.cellPositions(celdaIndex, 2);
    % Se definen los vértices del hexágono usando el radio escalar
    theta = linspace(0, 2*pi, 7);
    xHex = r_scalar .* cos(theta) + xCentro;
    yHex = r_scalar .* sin(theta) + yCentro;
    % Se genera una posición aleatoria dentro del hexágono:
    while true
        xRand = xCentro + (rand()*2 - 1) * r_scalar;
        yRand = yCentro + (rand()*2 - 1) * r_scalar;
        if inpolygon(xRand, yRand, xHex, yHex)
            nuevaPos = [xRand, yRand];
            break;
        end
    end
end
```

Script 27. Generar las posiciones de los usuarios dentro del área de cobertura.

Inicialización del entorno

```
function inicializarQTabla(app)
    % Re-inicialización de la Q-Table
    numCeldas = numel(app.cellPRBs);
    numEstados = length(app.stateMap);
    app.qTable = ones(numEstados, numCeldas) * 0.5;
    % Ajuste basado en la capacidad real de cada celda
    normCap = app.cellPRBs / max(app.cellPRBs);
    for i = 1:numEstados
        app.qTable(i, :) = 0.5 + 0.2 * normCap;
    end
end
```

Script 28. Creación de la tabla Q con dimensiones según el número de estados y acciones posibles

Implementación de la estrategia ϵ -greedy

```
function celdaDestino = seleccionarAccionVecina(app, estadoActual, celdaOrigen)
    vecinos = app.neighborTable{celdaOrigen};
    % Estrategia  $\epsilon$ -greedy: si (rand) es menor que (explorationRate)
    se elige una celda vecina aleatoria.
    if rand < app.qLearningParams.explorationRate
        celdaDestino = vecinos(randi(numel(vecinos)));
        return;
    end
    % Obtener los Q-values y carga actual de las celdas vecinas
    qVals = app.qTable(estadoActual, vecinos);
    cargas = app.cargaCeldas(vecinos);
    % Normalizar los Q-values
    normQ = (qVals - min(qVals)) / (max(qVals) - min(qVals) + eps);
```

```

        % Normalizar la carga y utilizar 1 - normalización para dar
mayor preferencia a celdas con menor carga
        normLoad = 1 - ((cargas - min(cargas)) / (max(cargas) -
min(cargas) + eps));
        % Término de equidad: cuanto menor sea la varianza entre las
cargas de vecinos, mejor.
        fairness = 1 - std(cargas) / (mean(cargas) + eps);
        % Combinar las métricas (0.4 para Q-value, 0.4 para carga y
0.2 para equidad)
        scores = 0.4 * normQ + 0.4 * normLoad + 0.2 * fairness;
        [~, idx] = max(scores);
        celdaDestino = vecinos(idx);
    end

```

Script 29. Implementación de la estrategia ε -greedy para la toma de decisiones.

Recompensa y aprendizaje

```

function recompensa = calcularRecompensa(app, sinrAnterior, sinrNuevo,
cargaOrigen, cargaDestino)
    % Calculo del throughput usando la fórmula de Shannon
    Thro_antes = app.calcularThroughput(sinrAnterior);
    Thro_nuevo = app.calcularThroughput(sinrNuevo);
    if Thro_antes < eps, Thro_antes = eps; end
    mejora = (Thro_nuevo / Thro_antes) - 1;
    % Balance de carga:
    % Recompensa si la celda destino tiene una reducción de al
menos 15%
    if cargaDestino <= 0.85 * cargaOrigen
        balanceoCarga = 1 - (cargaDestino / cargaOrigen);
    else
        balanceoCarga = -0.5;
    end
    % Índice de equidad (Jain):
    % Se calcula globalmente sobre la distribución de carga
    JainIndex = (sum(app.cargaCeldas.^2) / (numel(app.cargaCeldas)
* sum(app.cargaCeldas.^2)));
    % Combinar los términos mejora del throughput, balanceo de
carga y equidad(Jain)
    recompensa = 0.50 * mejora + 0.30 * balanceoCarga + 0.20 *
JainIndex;
    % Penalización para fluctuaciones extremas en la mejora
    if abs(mejora) > 0.5
        penalizacion_thpt = -1.5 * abs(mejora);
    else
        penalizacion_thpt = 0;
    end
    recompensa = recompensa + penalizacion_thpt;
    recompensa = max(min(recompensa, 1), -1);
end

```

Script 30. Calcular la recompensa usando la combinación de la mejora del throughput, balanceo de carga y equidad.

Actualización de la Tabla Q

```
function actualizarQTable(app, estadoActual, accion, recompensa, estadoNuevo)
    persistent buffer bufferPtr numExperiencias
    if isempty(buffer)
        buffer = cell(100, 1);
        bufferPtr = 1;
        numExperiencias = 0;
    end
    % Guardar la muestra actual en el buffer
    buffer{bufferPtr} = {estadoActual, accion, recompensa, estadoNuevo};
    bufferPtr = mod(bufferPtr, 100) + 1;
    numExperiencias = min(numExperiencias + 1, 100);
    % Número de muestras para actualizar en el mini-batch
    miniBatchSize = 10;
    for k = 1:miniBatchSize
        idx = randi(numExperiencias);
        sample = buffer{idx};
        if isempty(sample)
            continue;
        end
        % Desempaquetar la muestra
        [estadoAnterior, accion, recompensa, estadoNuevo] = deal(sample{:});
        % Verificación de los límites:
        if accion > size(app.qTable, 2)
            error('Índice de acción (%d) excede el número de columnas de la Q-table (%d)', ...
                accion, size(app.qTable, 2));
        end
        % Calcular del target y delta usando los valores de la muestra
        Objetivo = recompensa + app.qLearningParams.discountFactor * max(app.qTable(estadoNuevo, :));
        delta = app.qLearningParams.learningRate * (Objetivo - app.qTable(estadoAnterior, accion));
        % Limitar el delta para evitar cambios bruscos: máximo ±0.1
        delta = sign(delta) * min(abs(delta), 0.1);
        % Actualiza la Q-table en la posición (estadoOld, accionSample)
        app.qTable(estadoAnterior, accion) = app.qTable(estadoAnterior, accion) + delta;
    end
    % Reducción adaptativa de la tasa de aprendizaje:
    % app.qLearningParams.learningRate = max(0.01, app.qLearningParams.learningRate * 0.98);
end
```

Script 31. Aplicar la fórmula de Q-Learning para actualizar la Tabla-Q.

Aplicación del handover

```
function [usuariosTransferidos, celdasOrigen, celdasDestino] =  
ejecutarBalanceoDeCarga(app)  
    maxTransfers = 100;  
    usuariosTransferidos = zeros(maxTransfers, 1);  
    celdasOrigen = zeros(maxTransfers, 1);  
    celdasDestino = zeros(maxTransfers, 1);  
    handoverCount = 0;  
    numCeldas = size(app.cellPositions, 1);  
    for u = 1:numel(app.userCellAssignments)  
if app.userCellAssignments(u).handoverLock > 0  
        continue;  
    end  
    celdaOrigen = app.userCellAssignments(u).Celda;  
    if celdaOrigen < 1 || celdaOrigen > numCeldas  
        continue;  
    end  
    sinrActual = app.userCellAssignments(u).SINR;  
    cargaActual = app.cargaCeldas(celdaOrigen);  
    estadoActual = app.discretizarEstado(sinrActual,  
cargaActual);  
    vecinos = app.neighborTable(celdaOrigen);  
    if isempty(vecinos)  
        continue;  
    end  
    celdaDestinoOptima =  
app.seleccionarAccionVecina(estadoActual, celdaOrigen);  
    posUsuario = app.userCellAssignments(u).Posicion;  
    sinrNuevo = app.calcularSINR(posUsuario(1),  
posUsuario(2), celdaDestinoOptima);  
    condHysteresis = (sinrNuevo - sinrActual) >=  
app.handoverParams.A3Offset;  
    cargaDestino = app.cargaCeldas(celdaDestinoOptima);  
    throughputActual = app.calcularThroughput(sinrActual);  
    throughputNuevo = app.calcularThroughput(sinrNuevo);  
    condThroughput = ((throughputNuevo - throughputActual) /  
max(throughputActual, eps)) >= 0.40;  
    condTTT = (rand <= exp(-app.handoverParams.TTT/100));  
    condGeneral = app.cumpleCondicionesHandover(sinrActual,  
sinrNuevo, cargaActual, cargaDestino);  
    if condThroughput && condTTT && condGeneral &&  
condHysteresis  
        estadoNuevo = app.discretizarEstado(sinrNuevo,  
cargaDestino);  
        reward = app.calcularRecompensa(sinrActual,  
sinrNuevo, cargaActual, cargaDestino);  
        app.actualizarQTable(estadoActual,  
celdaDestinoOptima, reward, estadoNuevo);  
        if app.cumpleCondicionesHandover(sinrActual,  
sinrNuevo, cargaActual, cargaDestino)
```



```

        if app.realizarHandover(u, celdaOrigen,
celdaDestinoOptima)
            handoverCount = handoverCount + 1;
            usuariosTransferidos(handoverCount) = u;
            celdasOrigen(handoverCount) = celdaOrigen;
            celdasDestino(handoverCount) =
celdaDestinoOptima;
        end
    end
    app.qLearningParams.explorationRate =
max(app.qLearningParams.explorationRate * 0.995,
app.minimalExplorationRate);
end
end
end

```

Script 32. Para cada usuario se ejecuta el proceso de obtener estado, seleccionar la acción, evaluar las condiciones y actualizar Q.

Anexo 2. Parámetros generales del entorno de simulación de la red 4G LTE

Se presentan los parámetros generales usados en la simulación del sistema de red LTE. Estos valores fueron configurados dentro de las funciones correspondientes.

Tabla 20. Parámetros generales de simulación.

| Parámetro | Descripción |
|--------------------------------------|----------------------------------|
| Escenarios simulados | 3 con 5, 10 y 15 celdas |
| Número de simulaciones por escenario | 60 |
| Número de usuarios por celda | Variable para simular sobrecarga |
| Frecuencia de operación | 2100 MHz Banda LTE |
| Asignación de PRBs por celda | 50 |
| Tipos de servicio | Voz, video y datos |

A pesar de que las simulaciones descritas en la sección de resultados se desarrollaron utilizando la frecuencia de 2100 MHz, el sistema fue diseñado para ser compatible también con la frecuencia 1900 MHz, donde el único cambio requerido, es el ajuste de las fórmulas dependientes del uso del parámetro de frecuencia. La lógica del sistema y del algoritmo no se alteran, sino que mantiene su funcionalidad dando resultados positivos. Con esto se demuestra que la arquitectura del sistema es capaz de operar en diferentes bandas LTE sin necesidad de rediseño.