

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Universidad Distrital Francisco José De Caldas

Maestría en Ciencias de la Información y las Comunicaciones



**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY
LOAD BALANCING) SOBRE REDES LTE**

Autor

Ing. Jhon Jaime Ramírez Ochoa

Tutor

Roberto Cárdenas Castiblanco, D, Ing

Bogotá, Colombia

abril de 2020

Contenido

1.	Resumen	4
2.	Palabras Clave	5
3.	Introducción.....	6
4.	Problema de investigación	8
4.1	Pregunta de investigación	8
4.2	Justificación	8
4.3	Objetivos.....	8
5.	Estado del Arte.....	9
6.	Marco Teórico	15
6.1	Conceptos y Características Principales LTE	15
6.2	Arquitectura de LTE.....	17
6.3	Interfaces	19
6.4	OFDMA (Orthogonal Frequency Division Multiple Access).....	20
6.4.1	Generación de un símbolo OFDMA.....	23
6.5	Estructura de la Trama.....	25
6.5.1	Trama Tipo 1 FDD	25
6.5.2	Trama Tipo 2 TDD	25
6.6	Resource Blocks.....	28
6.7	Protocolos en Red de Acceso (E-UTRAN)	32
6.7.1	Protocolos en la Interfaz de Aire.....	32
6.7.2	Protocolos en la Interfaz S1 y X2	34
6.8	Handover	36
6.8.1	Preparación del Handover (Handover Preparation)	39
6.8.2	Ejecución del Handover (Handover Execution)	40
6.8.3	Finalización del Handover (Handover Completion)	41

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

6.9	Mobility Load Balancing (Balanceo de Carga Móvil)	47
7.	Metodología	53
7.1	Método de Modelación	53
7.2	Método de Experimentación	53
8.	Desarrollo de la propuesta	54
8.1	Definición Tipo de Usuarios	54
8.2	Definición de Vecindad.....	54
8.3	Balanceo de carga de primer orden.....	56
8.4	Balanceo de carga de segundo orden.....	58
8.5	Modelo para la generación de la tabla de rutas de balanceo	65
8.6	Modelamiento Matemático.....	66
8.6.1	Definición del Proceso de Poisson (Llegadas de paquetes al sistema).....	67
8.6.2	Proceso de Llegada de paquetes por Usuario.....	71
8.6.3	Definición de la Cadena de Markov para una Celda de un solo canal.....	74
8.6.4	Función Generadora de Probabilidad	76
8.6.5	Balanceo de Carga entre 2 Celdas	80
9.	Simulación y resultados.....	84
9.1	Definición del modelo de simulación	84
9.2	Verificación del modelo de simulación.....	86
9.3	Condiciones previas del primer escenario a simular	92
9.4	Estabilidad del sistema	96
9.5	Tipo de Distribución para la Tasa de pérdidas y el Throughput promedio	99
9.6	Simulaciones y Análisis.....	101
9.7	Condiciones previas del segundo escenario a simular	115
10.	Conclusiones.....	121
11.	Apéndices	124
	Bibliografía	170

1. Resumen

Como cualquier sistema de comunicaciones, las redes móviles son sistemas propensos a saturación de recursos en cuyo caso la calidad de las transmisiones se deteriora, lo que se traduce en bajo throughput, pérdida de paquetes, latencia en el envío de paquetes, entre otros. Partiendo de esta premisa, es necesario realizar una revisión profunda sobre el uso de los recursos del sistema, que permita incrementar la eficiencia de la transmisión bajo unas condiciones particulares.

El estándar de cuarta generación LTE ha sido desarrollado por la organización 3GPP (The 3rd Generation Partnership Project), dentro del cual se incorporan propuestas como los mecanismos SON (Self Organising Networks), que permiten a las redes poder configurarse y optimizarse por sí solas con el fin de lograr un mejor uso de los recursos disponibles en la red y por ende obtener un incremento en la capacidad de esta. Estos mecanismos han surgido principalmente por la necesidad de automatizar los procesos de configuración, optimización y recuperación de la red. [1]

Dentro del estándar mencionado se han agrupado todas las iniciativas por parte de entes académicos y entes privados con el fin de definir todas las especificaciones técnicas al respecto. Entre estas especificaciones se encuentran las relacionadas con el feature MLB (Mobility Load Balancing), en donde se describe cómo varios eNodeB son capaces de transferir usuarios entre sí con el fin de balancear carga entre sus celdas, cómo debe ser el proceso de comunicación entre estos y cómo se ajustan los parámetros de handover para llevar a cabo el proceso. [2]

Con base en ello, el presente trabajo inicia recopilando la información establecida en la norma de la 3GPP, revisando cómo se implementa este feature en redes reales, cuáles son los algoritmos sobre los cuales se implementa MLB y proponiendo un nuevo modelo con el fin de mejorar los resultados obtenidos con los algoritmos empleados actualmente, en términos de throughput y pérdida de paquetes.

2. Palabras Clave

Redes auto-organizadas, Balanceo de Carga Móvil, Rendimiento, Optimización.

3. Introducción

Actualmente el sector de las Redes Móviles se encuentra en gran dinamismo debido al crecimiento en la cantidad de usuarios y altos requerimientos por servicios que demandan cada vez más Ancho de Banda como lo son Transmisiones Multimedia en tiempo real, IoT (Internet of Things), Tecnologías Cloud entre otros.

El posicionamiento de tecnologías móviles de última generación como lo son **LTE** y **LTE-ADVANCED** obligan al desarrollo e implementación de soluciones necesarias para la prestación adecuada del servicio a los usuarios finales.

Con el fin de incrementar el rendimiento global en la red, los operadores móviles cuentan con 4 herramientas enunciadas a continuación:

- Incremento en la cantidad de estaciones base (eNodeB), con el fin de poder reducir el área de cobertura de cada celda.
- Incremento del espectro radioeléctrico.
- Incremento en la eficiencia espectral de la tecnología.
- Implementación de mecanismos que permitan un uso eficiente de los recursos.

Las dos primeras opciones pueden llegar a no ser viables ya que están restringidas a decisiones de entes regulatorios que pueden ser bastante costosas y tomar un periodo de tiempo extenso en ser aprobadas, y la tercera opción está limitada al desarrollo tecnológico. Es por ende que la atención se centra en la implementación de mecanismos que permitan un uso eficiente de los recursos. [3]

En la norma de LTE desarrollada por 3GPP se mencionan mecanismos que permiten a la red autoconfigurarse con el fin de incrementar la capacidad de esta sacando el máximo provecho de los recursos a nivel espectral. Dentro de estos mecanismos se describen varios **features**, entre los cuales se encuentra MLB (Mobility Load Balancing), cuya finalidad es realizar balanceo de carga entre las diferentes celdas de los eNodeB y poder utilizar de forma más eficiente los recursos de radio disponibles en la red. [4] [5]

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

En la actualidad los mecanismos de balanceo se implementan sobre celdas adyacentes permitiendo el traspaso de usuarios con el fin de equilibrar la carga de ambas celdas, sin embargo, el problema surge cuando dentro de un conjunto de celdas (clúster), las celdas potenciales para realizar el balanceo de carga no son adyacentes. ¿Qué mecanismo se puede implementar en dicho caso para poder balancear la carga y finalmente poder incrementar el throughput global de la red?

En el presente documento se expondrá la propuesta que da solución a este problema mediante la implementación de MLB para balanceo de carga de primer y segundo orden, incluyendo análisis de resultados y comparativos con los mecanismos empleados actualmente.

4. Problema de investigación

4.1 Pregunta de investigación

¿En qué porcentaje puede aumentar el throughput en la red a la hora de implementar balanceo de carga sobre un clúster completo?

4.2 Justificación

Los procedimientos basados en MLB que se plantean en la literatura se limitan a evaluar las celdas vecinas del eNodeB que presentan sobrecarga (**overload**), pero no se observan procedimientos donde se realice un balanceo completo de un clúster de eNodeB, es por ende que se ve una oportunidad de investigación en este tema para proponer un nuevo modelo que mejore la asignación de recursos de radio dentro de un clúster. Para poder desarrollar los objetivos de este trabajo de grado se propone evaluar un método de balanceo de segundo orden la cual se desarrollará en capítulos posteriores.

4.3 Objetivos

4.3.1 Objetivo General

- Desarrollar un modelo basado en MLB (Mobility Load Balancing) sobre un clúster completo de eNodeB en una red LTE.

4.3.2 Objetivos Específicos

- Desarrollo de un modelo matemático que permita sustentar la implementación de un algoritmo basado en MLB sobre un clúster completo de eNodeB.
- Verificar la validez del modelo desarrollado mediante una comparación con los algoritmos de MLB empleados actualmente a través de simulación.

5. Estado del Arte

MLB (Mobility Load Balancing) es una funcionalidad incluida en el estándar desarrollado por la 3GPP para redes LTE (Release 9), que fundamentalmente se basa en la transferencia de tráfico de una celda con alta carga a una celda con menos carga que disponga de recursos de radio (subportadoras) disponibles con el fin de poder balancear el tráfico, cuyo funcionamiento se expone en el subcapítulo 6.9 del presente documento. En este capítulo se mencionan las principales investigaciones que han sido publicadas en revistas como IEEE, y que servirá como base para el desarrollo de la propuesta del trabajo de investigación.

Una de las primeras variaciones del modelo de balanceo se expone en [6]. Dicha propuesta se denomina **CMLB (Cooperative Mobility Load Balancing)**, la cual se basa en obtener la medición “**less-loaded neighbouring cells**” como parámetro de entrada con el fin de calcular la diferencia de carga con respecto a todas las celdas vecinas y decidir a cuál de todas poder transferir tráfico de radio. **MLB** no funciona de la mejor manera cuando varias celdas sobrecargadas intentan balancear carga al mismo tiempo con una celda de baja carga, sin embargo, **CMLB** ofrece una alternativa para evaluar prioridades antes de iniciar el balanceo de tal forma que se balancee la carga de una forma más justa y eficiente. Brevemente el método se puede resumir en los siguientes pasos [6]:

- La celda que presenta alta congestión de tráfico (Hot-Spot Cell) envía un mensaje Request a una celda vecina (Partner Cell). Ésta le responde notificándole el siguiente parámetro:
 - Responde si es apta para recibir tráfico (shifting traffic). En caso de ser una celda apta, envía el valor de la carga actual con el fin de que el eNodeB origen pueda calcular la diferencia de carga (Shifting traffic calculation) y poder decidir si a este nodo se le puede transferir carga.
- El eNodeB origen calcula el valor “**Shifting Traffic Calculation**” de todas las celdas vecinas que son aptas. Se escoge la celda que presente el valor más alto y se envía la información de los usuarios que se encuentran en borde celda al nuevo eNodeB para ser transferidos.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

- El nuevo eNodeB evalúa cómo debe adaptar los recursos de radio para cumplir con los requerimientos de los nuevos usuarios y adicionalmente envía una notificación al eNodeB origen para ajustar el parámetro HOoff (umbral para iniciar el proceso de handover) con el fin de facilitar el proceso de Handover.
- El eNodeB origen recibe la notificación del valor HOoff con el fin de ajustar este parámetro. Inmediatamente el valor ha sido ajustado, los usuarios en borde de celda realizan el proceso de handover para ser trasladados a la nueva celda.

Hay otro modelo basado en MLB similar llamado **NMLB (Neighborhood Mobility Load Balancing)**. El objetivo general del algoritmo es configurar los parámetros físicos de las antenas y los parámetros de handover con el fin de disminuir la cobertura del eNodeB origen y aumentar la cobertura del eNodeB al que se la transferirán los nuevos usuarios. [7]

El eNodeB origen debe liberar tráfico hacia otros eNodeB por ende, debe realizar el siguiente proceso [8] [9]:

- Reducción y Aumento de las coberturas de los eNodeB.
- Reducción y Aumento de la potencia de Transmisión.
- Variación de los parámetros de Handover.

Este método a diferencia de CMLB no tiene en cuenta si hay otras celdas tratando de balancear su propia carga por ende no es recomendable su uso, ya que como se mencionó anteriormente, una celda con recursos de radio disponibles puede verse saturada en un periodo de tiempo muy corto.

En otra investigación del año 2018 [10], se propone un modelo de balanceo sobre una red híbrida (small cells y macro cells). Mediante simulaciones se busca determinar que tan influyente son los patrones de trayectoria y velocidad de los usuarios sobre los resultados de balanceo ejecutados. Este artículo se basa en los resultados obtenidos en [11] y [12] y tiene como objetivo minimizar la desviación estándar de la carga dentro de un conjunto de celdas determinado, es decir lograr que todas las celdas de la red se encuentren con un nivel promedio de carga similar, intentando que la carga a nivel de señalización del protocolo RRC sea la menor posible. Este protocolo se explicará en el desarrollo del marco teórico.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Con la implementación del algoritmo se reduce la desviación estándar de la carga de un 7.58% a un 60.74% dependiendo el tipo de escenario, sin embargo, se ve un incremento de un 10.79% en la cantidad de handovers, comparado con escenarios donde no se aplica el modelo de balanceo de carga. Adicional a ello, se concluye que es importante que la funcionalidad de **MRO (Mobility Robustness Coordination)** deba implementarse en paralelo a la funcionalidad de **MLB** con el fin de reducir los intentos fallidos de conexión entre los usuarios y los eNodeB.

Retomando el modelo propuesto en [10], la ruta que siga un usuario móvil y la velocidad con el que se desplace afecta el resultado de la implementación de MLB, para ello se plantearon escenarios en los que los usuarios se desplazan en trayectorias rectangulares y circulares, adicionalmente usuarios que se desplazan caminando (velocidad promedio menos a 5km/h) y usuarios que se desplaza en vehículo (velocidad promedio de 40km/h). Los resultados del trabajo muestran que bajo un escenario donde los usuarios se desplazan caminando en trayectoria circular se logra una efectividad del 54.7% a la hora de balancear la carga entre celdas, mientras que para un escenario donde los usuarios se desplazan en vehículo en trayectoria rectangular se alcanza una efectividad del 26.8%. La efectividad se mide como la cantidad de balanceos exitosos sobre el total de casos donde se requería realizar balanceo de carga.

En [13] el autor plantea un algoritmo adaptativo para poder implementar el modelo de MLB para small cells (celdas de baja cobertura). Debido a que las celdas solo cubren un área geográfica pequeña, el escenario es propenso a experimentar una alta tasa de handovers cuando los usuarios se encuentran en movimiento. Para ello se plantea un modelo basado en umbrales dinámicos con el fin de minimizar la desviación estándar del RBUR (Promedio de RB asignados en un periodo de tiempo t).

En primer lugar, se debe calcular la carga actual promedio de las celdas mediante la siguiente expresión:

$$\overline{RBUR} = \frac{1}{T * N_{PBR}} \sum_{\tau \in (t-T, t)} RB_I^\tau (a)$$

Donde T corresponde al periodo de observación dado, N_{PBR} corresponde al número total de Resources Block asignados durante el periodo T , y el factor RB_I^τ corresponde a la

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

cantidad de Resource Block ocupados dentro de un periodo de tiempo τ dentro del intervalo $(t - T, t)$. Es evidente observar que un valor alto de \overline{RBUR} indicará un nivel alto de carga dentro de la celda.

Al igual que en el artículo mencionado anteriormente [12], se busca minimizar la desviación estándar de la carga a lo largo de la red mediante variaciones dinámicas del parámetro de umbrales para la ejecución de los handover con el fin de incrementar la eficiencia del mecanismo de balanceo tradicional.

En [14] se menciona la importancia de MLB, brindando un balance entre la distribución de los recursos en la red y la demanda de tráfico por parte de los usuarios. En este artículo se identifican 2 políticas claves que caracterizan conjuntamente estas variables y se proponen algoritmos para estabilizar la red e incrementar la eficiencia de recursos dentro de la red. Se adopta un modelo de sincronización para idear la primera política llamada “Asignación de carga” que determina la cantidad de carga que puede ser balanceada entre las celdas. La segunda política es llamada “selección de destino” en la cual se determina la celda óptima para iniciar el balanceo.

A través del desarrollo matemático se muestra la estabilidad y la optimización del método de MLB empleado. Los resultados obtenidos en la simulación permiten identificar cómo mediante un tratamiento riguroso de las 2 políticas mencionadas anteriormente se logran mejores resultados en términos de throughput y Calidad de servicio ofrecida a los usuarios (User Equipments) en comparación con el método tradicional de MLB.

En [15] se menciona que el estándar de LTE difundido por el 3GPP brinda la base para entender el funcionamiento en términos de protocolos, tasas de transmisión, métodos de acceso al canal, etc de dichas tecnologías, sin embargo, también brinda la posibilidad de poder incorporar nuevas tecnologías que permitan incrementar la eficiencia de los protocolos que se puede cuantificar en un mayor throughput, menor retardo y latencia, mayor balance de carga, menor tasa de pérdida de paquetes, etc.

En este artículo el autor propone incorporar SDN (Software Defined Network) al EPC (Evolved Packet Core) de la red gracias a las bondades que tiene al ser una tecnología más flexible, programable y de menor coste para el operador. SDN permite separar los planos de control y de usuarios de una manera sencilla.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Los principales retos que se presentan en el Core de la red son el enrutamiento de tráfico, tráfico compartido entre diferentes operadores (Traffic sharing) y la administración del tráfico. Este último factor es el más crítico cuando los usuarios experimentan movilidad entre celdas debido a que el intercambio de mensajes y eventos entre celdas debe desplegarse de forma confiable y en el menor tiempo posible con el fin de que los handover se ejecuten correctamente. El autor se centra en la reducción de sobrecarga en la red de Core mediante el uso de balanceo de carga, especialmente sobre el SGW (Serving Gateway). Al implementar este balanceo se obtiene un incremento efectivo en el throughput y un decremento en la latencia de la red.

En el artículo citado en [16] se tiene como objetivo plantear un modelo que permita mitigar el efecto de la disminución del nivel de RSRP (Reference Signal Receive Power) mínimo, que trae consigo la implementación del MLB trasladando usuarios de borde de celda de una celda con alta carga a una única celda con carga baja. Allí se menciona un método en el cual antes de iniciar el balanceo de carga se evalúan las condiciones de los usuarios que serán transferidos, con el fin de que dichos usuarios no queden por debajo del umbral de RSRP mínimo para ofrecer un canal estable y de esta forma no tener un rendimiento bajo en términos de throughput y pérdida de paquetes. Los resultados de este artículo se basan en [17] [18].

En el artículo citado en [19] se valida que el modelo de MLB efectivamente permite balancear cargas entre celdas adyacentes, a costa de la degradación del servicio que experimentan los usuarios de borde celdas (los cuales son trasladados de una celda a otra). Al realizar ajustes en los parámetros de Handover, las coberturas de cada celda varían y causa que los usuarios ubicados en las zonas de borde de celda sean más propensos a experimentar interferencia proveniente de las celdas adyacentes. En este paper se comprueba que para los usuarios de borde celda de celdas que implementaron balanceo de carga tuvieron una disminución del CQI (Channel Quality Indicator) para el canal de bajada e incrementaron el nivel de interferencia en el canal PUSCH (Physical Uplink Shared Channel) de subida. Los resultados del paper (para dicho escenario) muestra que para celdas donde ha habido una modificación sustancial de los parámetros de handover, tanto las conexiones fallidas como los handover fallidos están por encima del 85%.

En el artículo citado en [20] se plantea mitigar los 2 efectos negativos que tiene implementar el modelo MLB. La sobreestimación de carga por celda lo cual puede acarrear handovers

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

innecesarios y capacidad baja a nivel de capa de transporte debido a dichos handovers ejecutados. Esto puede ocasionar escenarios como el traslado de UE a celdas con recursos de radio disponibles, pero con baja capacidad a nivel de red de transporte. El desarrollo de la solución muestra cómo se debe prevenir en cierto grado ejecutar balanceo de carga a celdas con baja capacidad a nivel de capa de transporte con el fin de no bajar el performance de la red. Así mismo es importante tener en cuenta la relación que existe entre MLB y MRO, ya que el efecto de uno tiene efecto sobre el otro como se menciona en [21].

6. Marco Teórico

La evolución de los sistemas de Redes Móviles se ha dado a pasos agigantados. Así como se observa en la figura 1 en prácticamente 20 años se pasó de ofrecer tasas de transmisión de 9.6 Kbps a tener hoy en día con LTE-Advanced tasas de transmisión cercanas a 1Gbps. Este avance se ha dado de forma gradual con la aparición de GSM, GPRS, UMTS, HSPA, HSPA+ y hoy en día LTE y LTE-Advanced.

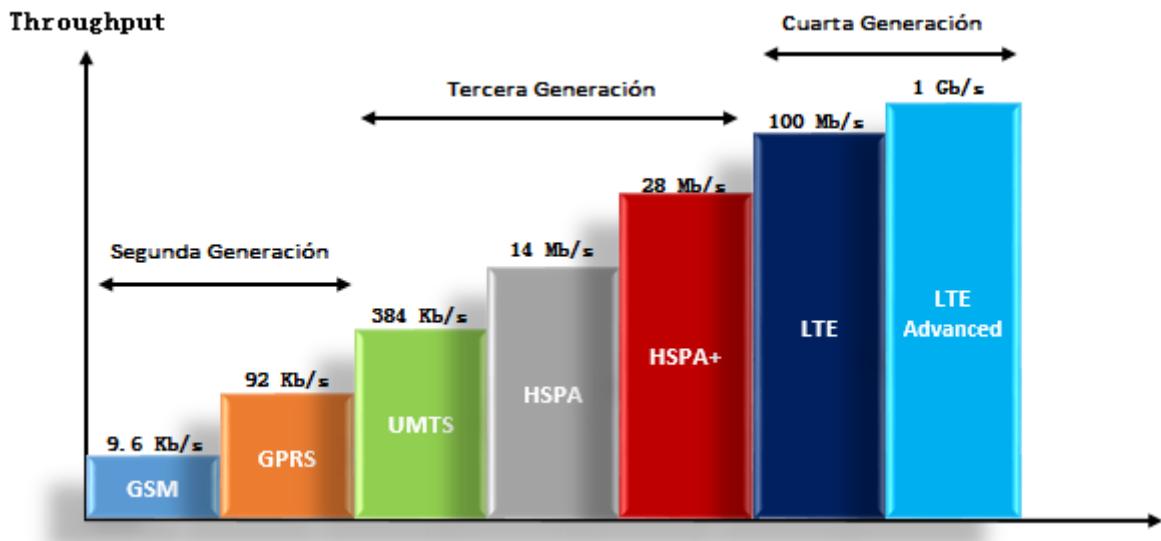


Figura 1. Evolución de Tecnologías Móviles

6.1 Conceptos y Características Principales LTE

LTE (Long Term Evolution) es la tecnología móvil más reciente descrita desde el release 8 hasta el release 12 de la 3GPP actualmente. En el reporte técnico 25.913 de la 3GPP se definen los objetivos principales de LTE [22] [23] [24] [25]:

- Permite trabajar con Ancho de Banda dinámico desde 1.4 MHz hasta 20 MHZ.
- Tasa de Transmisión máxima en el canal de Descarga (Downlink) de 100Mbps si el UE cuenta con 2 antenas de Recepción.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- Tasa de Transmisión máxima en el canal de Subida (Uplink) de 50Mbps si el UE cuenta con 1 antena de Transmisión.
- Latencia entre el UE y el eNodeB menor a 10ms.
- Mayor eficiencia espectral promedio (3 a 4 veces) en el canal de Downlink que el release 6 HSPDA.
- Mayor eficiencia espectral promedio (2 a 3 veces) en el canal de Uplink que el release 6 HSPUA.
- La arquitectura de LTE es plana, lo cual minimiza la cantidad de elementos de red.
- Se definen 3 mediciones importantes que permiten la asignación de recursos a los usuarios, las cuales son:

RSSI – Received Signal Strength Indicator: Es la potencia recibida total promedio percibida únicamente en los símbolos OFDM que contiene señales de referencia.

RSRP – Reference Signal Received Power: Es la potencia de las señales de referencia dividida sobre el ancho de banda total. Esta medida es la que se emplea para generar los mapas cobertura, gestionar los procedimientos de handover, entre otros.

RSRQ – Reference Signal Received Quality: Permite obtener una medición con respecto a la calidad de las señales de referencia recibidas.

- En LTE a celda se la asigna un PCI (Physical Cell ID), el cual es un identificador que permite diferenciarla de las demás celdas.
- La mayoría de los servicios que ofrece LTE se encuentran dentro del Dominio PS (Packet Switched) sin embargo, también incluye funcionalidad que le permiten la interoperabilidad con servicio del dominio CS (Circuit Switched) como lo es el feature de FallBack (servicio de voz tradicional) usado en UMTS.
- LTE soporta el servicio de VoIP (Voice over IP). Sin embargo, por falta de cobertura o por limitaciones en el terminal (UE) es posible acceder al servicio de voz por medio del proceso de Fallback se puede dar continuidad al servicio bajo redes UMTS, GMS o CDMA 2000.
- LTE soporta servicios de Multimedia Broadcast Multicast Services (MBMS) para la transmisión de TV móvil.
- La transmisión entre el UE y el eNodeB puede ser llevada a cabo mediante TDD (Time Division Duplex) o FDD (Frequency Division Duplex).

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- LTE garantiza interoperabilidad con otras tecnologías de comunicación móvil como UMTS, GSM y CDMA 2000.
- Para el canal de bajada (Downlink) la técnica de acceso al medio es OFDMA (Orthogonal Frequency Division Multiple Access) y para el canal de subida (Uplink) es SC-FDMA (Single Carrier – Frequency Division Multiple Access).
- LTE soporta MIMO (Multiple Input Multiple Output) para el canal de bajada (Downlink). Este feature se encuentra especificado del release 10 en adelante de la 3GPP.
- Debido a la llegada de LTE, se puede realizar un refarming (nueva distribución del espectro electromagnético) para las diferentes tecnologías inalámbricas.
- LTE facilita la creación de las celdas vecinas con el fin de garantizar el proceso de Handover.
- LTE incluye funcionalidades SON (Self Organising Network) con el fin de automatizar la configuración de la Red, Optimización, Detección y Corrección de Fallas.
- LTE Advanced se describe desde el release 10 de la 3GPP.
- LTE Advanced incluye el feature de Carrier Aggregation con el fin de ofrecer Anchos de Banda efectivo más amplios e introduce MIMO para el canal de Uplink.

6.2 Arquitectura de LTE

- LTE se refiere a la Red de Acceso de Radio UMTS Mejorada (E-TRAN Evolved UMTS Radio Access Network), mientras que el System Architecture Evolution (SAE) se refiere a la Evolved Packet Core (EPC). En la figura 2 se ilustra la división entre la Red de acceso y la Red de Core.
- LTE usa una arquitectura plana, por lo cual elementos como la RNC o la BSC de tecnologías anteriores como GSM y UMTS ya no existen, y todas sus funciones son ejecutadas directamente en el eNodeB (radio base).
- En LTE el eNodeB es el equivalente al NodeB en UMTS o a la BTS en GSM. Las funciones del control y asignación de los Recursos de Radio son ejecutadas directamente en el eNodeB.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

- Los eNodeB se comunican con la Red de Core mediante 2 entidades: El MME (Mobility Management Entity) para el plano de Control y Señalización, y el S-GW (Serving Gateway) para el plano de Usuario. Esta comunicación se lleva a cabo mediante las interfaces S1-Control Plane y S1-User Plane.
- Dentro de la Red de Core El MME y el S-GW se comunican con el PDN (Packet Data Network), el cual tiene como función realizar el enrutamiento a Redes Externas (Redes de otros operadores, Redes IMS, Internet, etc).
- Dependiendo de los Requerimientos a nivel de cobertura y QoS es necesario implementar celdas Macro, Micro y Femto BTS, con el fin de evitar los Hot Spot (Lugares con Alta concentración de Usuarios y los Cold Spot (Lugares con baja cobertura).

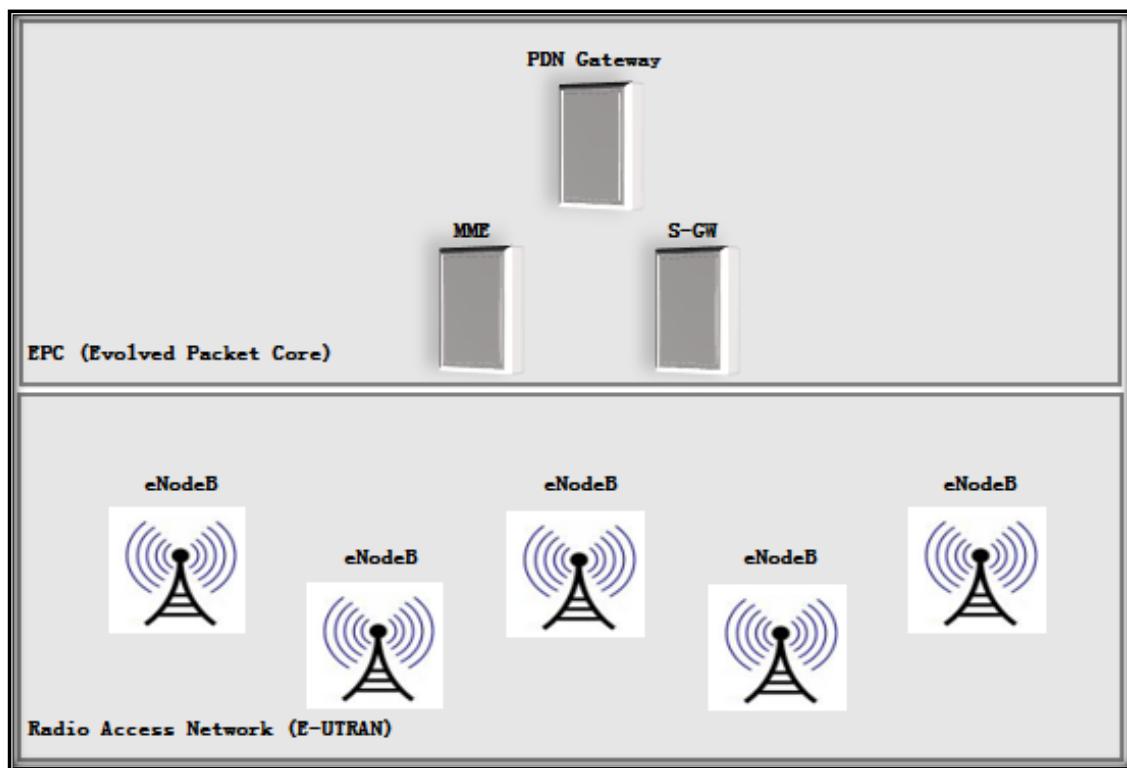


Figura 2. División entre la Red de acceso y la Red de Core [2] [24]

6.3 Interfaces

Así como se definieron los elementos principales que hacen parte de una red LTE-EPC, se deben definir las interfaces, las cuales son las que permiten la comunicación entre dichos elementos. [26]

- Interfaz de Aire (Uu): Esta interfaz permite la comunicación entre el User Equipment (UE) y el eNodeB. Esta comunicación se logra mediante la transmisión y recepción de señales electromagnéticas moduladas sobre una portadora específica asignada por el eNodeB.
- Interfaz X2: Esta interfaz es la encargada de permitir la comunicación entre 2 eNodeB. Sobre esta interfaz se puede enviar información de control y señalización, e información de usuario.
- Interfaz S1: Esta interfaz permite la comunicación entre el eNodeB y el EPC, específicamente con el MME y el S-GW.
- Tanto las interfaces X2 como S1 son basadas en IP.
- En la figura 3 se observa el conjunto total de interfaces empleadas en la Red de Acceso y en la Red de Core sin embargo, para el presente trabajo las interfaces de interés serán: Uu, X2 y S1.

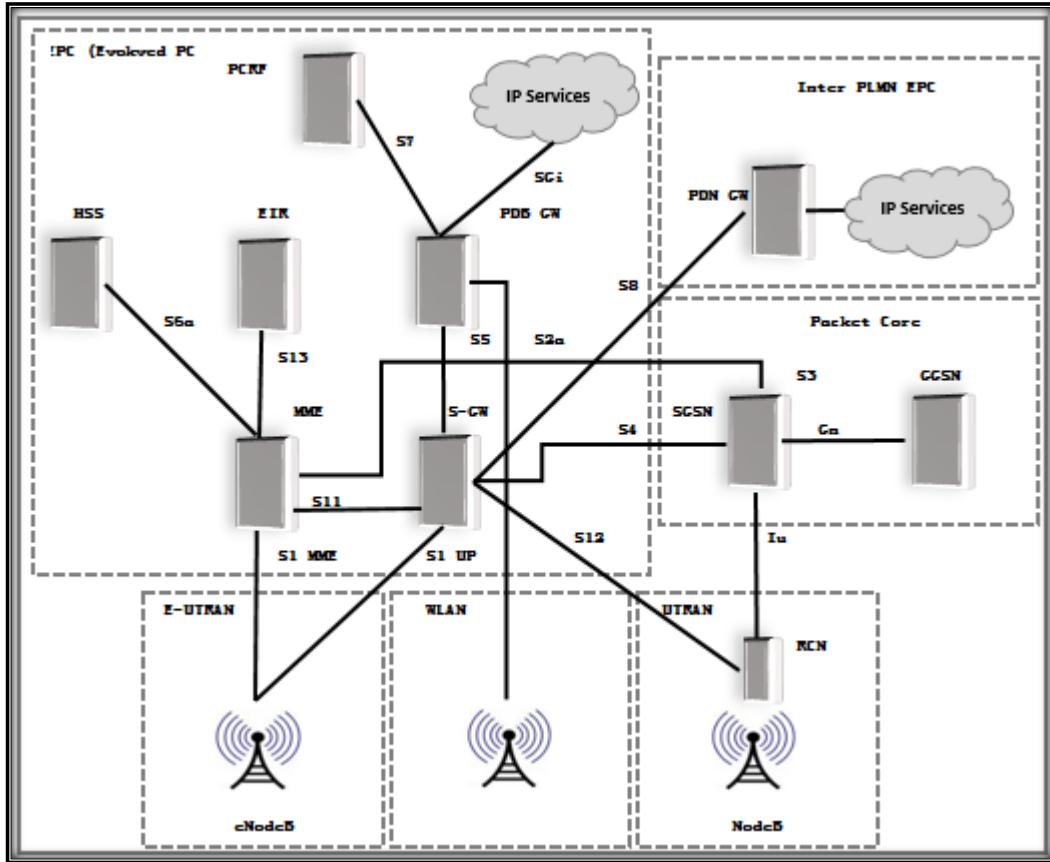


Figura 3 . Diagrama de Interfaces en una Red LTE [27] [28]

6.4 OFDMA (Orthogonal Frequency Division Multiple Access)

La técnica para el acceso múltiple de usuarios basada en el uso de funciones senoidales ortogonales entre sí, permite la asignación de recursos a nivel de Radio de una forma más eficiente que lo que se logra con otras técnicas de acceso (tabla 1) empleadas en tecnologías anteriores.

	FDMA	TDMA	CDMA	OFDMA
Recurso	Portadoras RF	Ranuras de tiempo	Códigos	Sub portadoras
Example System	AMPS	GSM	UMTS	LTE

Tabla 1. Descripción Técnicas de acceso en Tecnologías Móviles [28]

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

En OFDMA se disponen de subportadoras (subcarriers) las cuales son asignadas a los usuarios de acuerdo con el tipo de Servicio que demande y a las políticas de QoS previamente configuradas. [29]

Dichas subportadoras se pueden modelar de la siguiente forma:

$$x(t) = e^{j2\pi k \Delta f t} * rect(t), \quad 0 \leq k \leq K - 1 \quad (1)$$

$$\text{Siendo } f_k = k * \Delta f, \Delta f = \frac{1}{T_s}.$$

Las cuales tienen una respuesta en el dominio de la frecuencia como se observa en la figura 4.

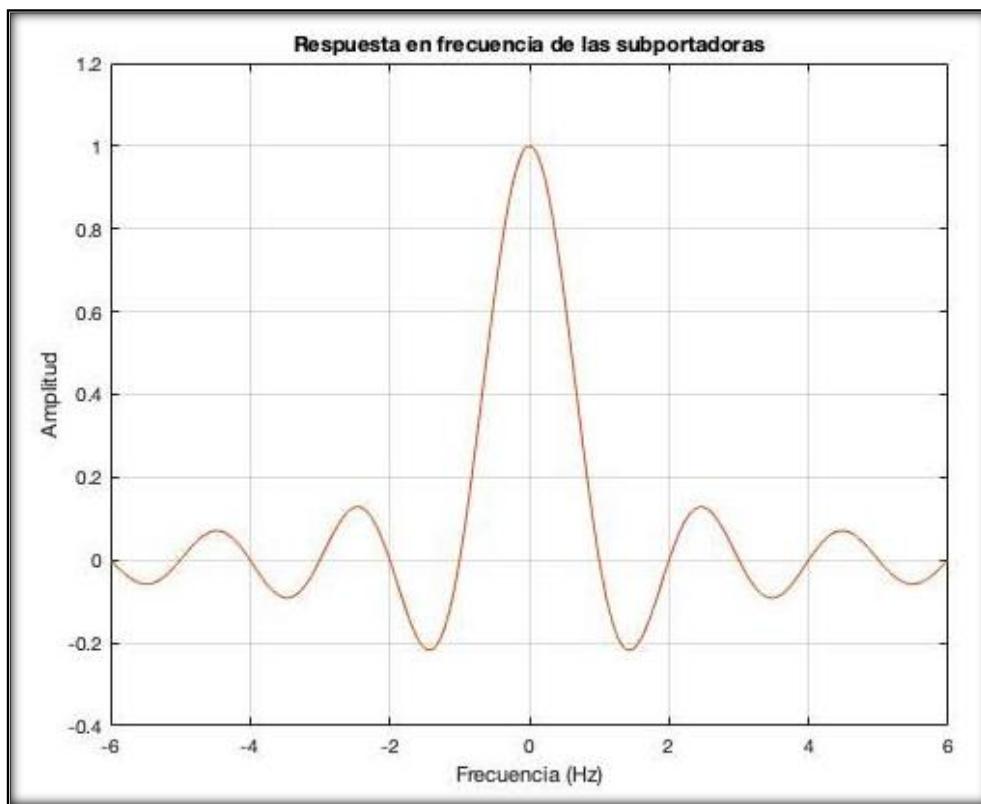


Figura 4. Espectro de una señal rectangular de duración T_s

Al ser funciones ortogonales en el intervalo de tiempo T_s (duración de cada símbolo), la integración del producto de estas es nula exceptuando el caso donde $m=k$, como se detalla a continuación:

$$R_{k,m} = \frac{1}{T_S} \int_0^{T_S} x_m(t) * \overline{x_k(t)} = \int_0^{T_S} e^{j2\pi(m-k)\Delta f t} dt = \begin{cases} 1, & \text{si } m = k \\ 0, & \text{si } m \neq k \end{cases} \quad (2)$$

Como se observa en la figura 5, a pesar de que en el dominio frecuencial las señales se solapen, el hecho de que las funciones portadoras sean ortogonales entre sí permite que el receptor pueda recuperar la señal original enviada por cada una de las subportadoras.

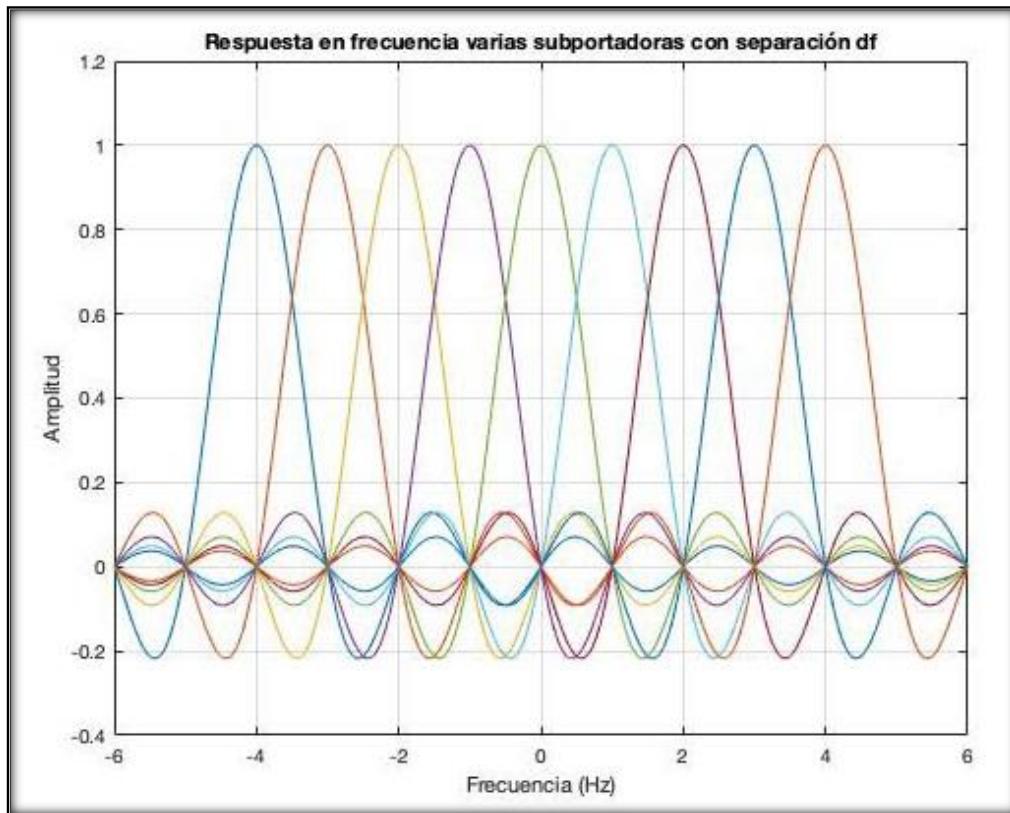


Figura 5. Representación en el dominio de la frecuencia de señales rectangulares moduladas por funciones senoidales ortogonales entre sí.

Dependiendo las condiciones del canal de transmisión se elige una modulación específica (BPSK, 16QAM o 64QAM) y posteriormente se aplica la Transformada Rápida Inversa de Fourier para generar un símbolo OFDMA en el dominio del tiempo, el cual será transmitido por la interfaz de aire.

Una de las ventajas de emplear OFDMA, es que esta técnica es tolerante al efecto de retardo en la propagación debido a las características propias del canal ya que la duración de los símbolos generados es relativamente alta. Esto se puede lograr debido a que la información es transmitida en paralelo (símbolos por subportadora) y no de manera serial.

- Generalmente un símbolo en OFMDA tiene una duración aproximada de $71.35 \mu s$, valor que surge de la siguiente definición numérica:

$$Duración\ del\ símbolo = \frac{1}{15\ KHz} + Duración\ prefijo\ cíclico \quad (3)$$

Donde el valor del prefijo cíclico es aproximadamente $4.683 \mu s$.

6.4.1 Generación de un símbolo OFDMA.

Cada símbolo en banda base es modulado por una subportadora senoidal, de acuerdo con lo observado en la figura 6:

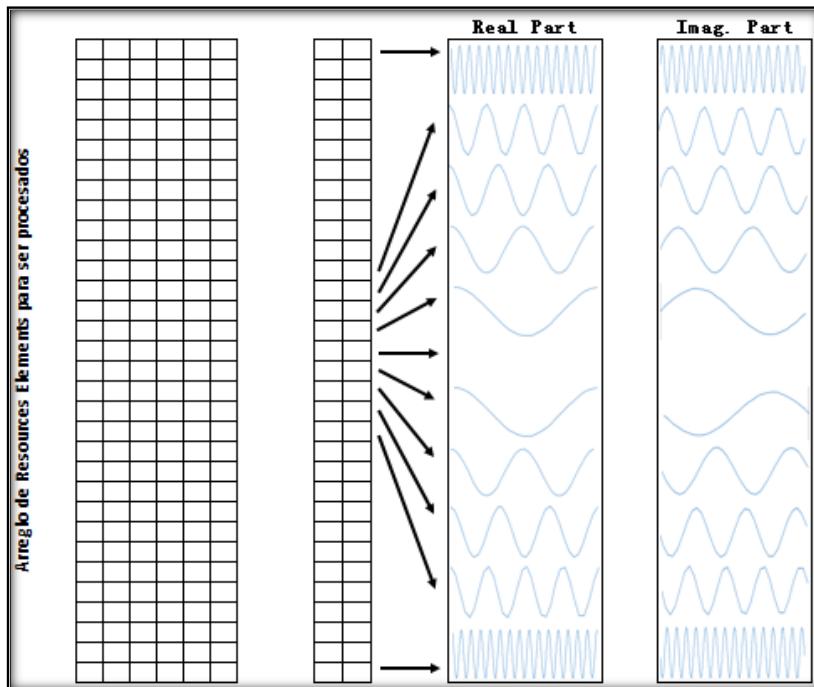


Figura 6. Representación gráfica generación señales OFDM [2] [29]

La información origen se genera de forma serial en el emisor. Esta señal digital pasa por un conversor Serial a Paralelo, donde cada símbolo se modula por una portadora que debe satisfacer la condición ($df = 1/Ts$) con el fin de garantizar la ortogonalidad de las funciones moduladoras, dicha modulación se representa como la multiplicación de cada símbolo por la parte real de la exponencial compleja. Finalmente, se la realiza la suma algebraica de cada

símbolo modulado y se envía al elemento radiante para ser enviado al receptor. Ver figura 7

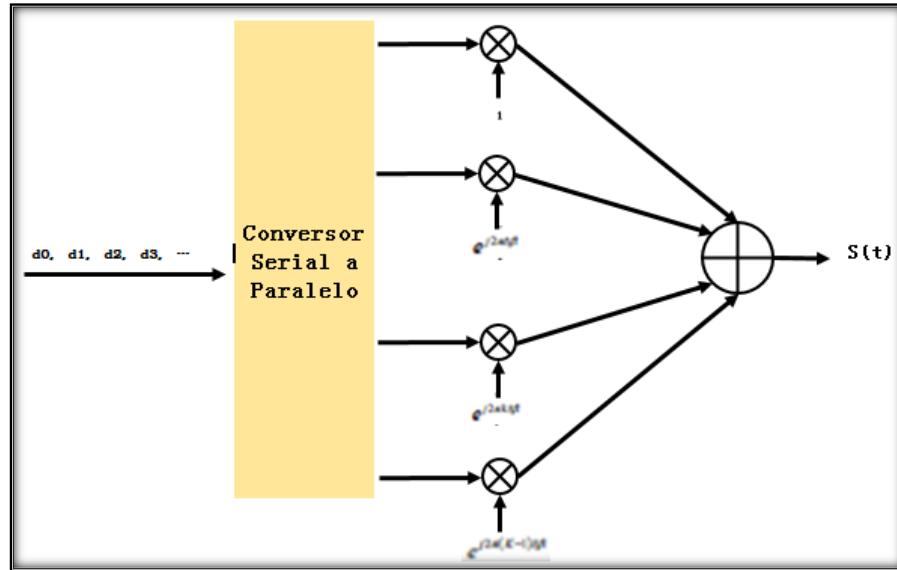


Figura 7. Proceso secuencial generación símbolos OFDM [2] [29]

En el receptor la señal se multiplica por cada una de las portadoras empleadas en el proceso de modulación, con ello se garantiza que cada símbolo se pueda recuperar. Ver figura 8

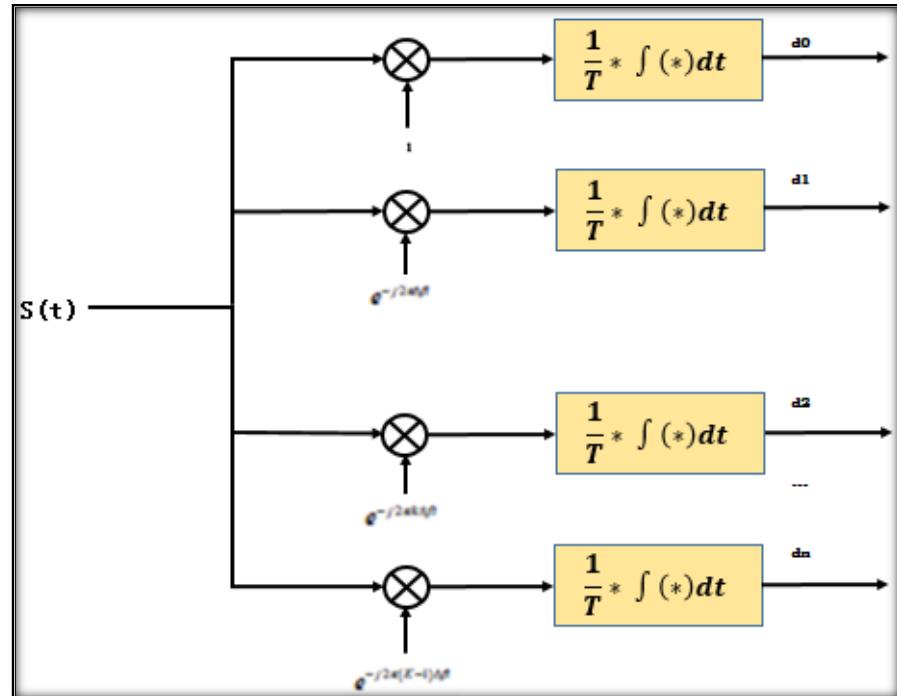


Figura 8. Recuperación señal original a partir de los símbolos OFDM [2] [29]

6.5 Estructura de la Trama

Las tramas de comunicación entre el UE y el eNodeB pueden ser de dos tipos:

6.5.1 Trama Tipo 1 FDD

Este tipo de Trama incluye 10 subtramas, cada una con una duración de 1ms, por lo cual una trama completa tiene una duración de 10ms. Cada Subtrama se divide en 2 slots de 0.5ms de duración cada uno como se puede observar en la figura 9.

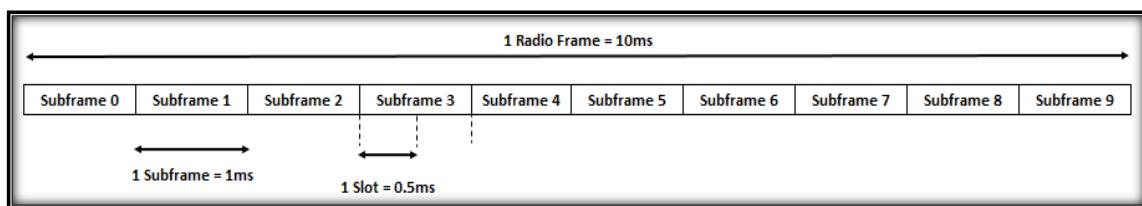


Figura 9. Estructura de la Trama tipo 1 [26]

Este tipo de trama se puede aplicar tanto para el canal descendente (Downlink) como para el canal ascendente (Uplink). Desde el UE, el tiempo de la trama del canal ascendente precedido el tiempo de la trama del canal descendente se debe tener en cuenta para poder calcular el retardo de propagación. Este tiempo es determinado por el “Timing Advance Instructions Signalled by the eNodeB”.

Para el caso de las tramas FDD, los eNodeB adyacentes no tienen que estar sincronizados uno con el otro.

6.5.2 Trama Tipo 2 TDD

Este tipo de Trama incluye 10 subtramas, cada una con una duración de 1ms, por lo cual una trama completa tiene una duración de 10ms. Cada Subtrama se divide en 2 slots de 0.5ms de duración cada uno como se puede observar en la figura 10

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

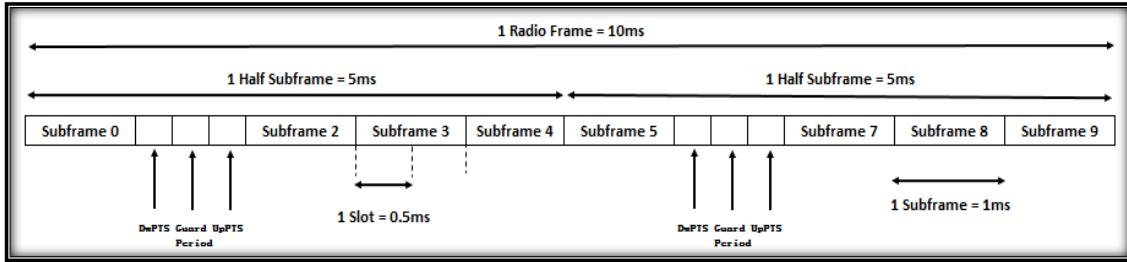


Figura 10. Estructura de la Trama tipo 2 [26] [2]

Para este tipo de Trama se hace la división en 2 partes, cada una con un total de 5 subtramas. Cada subtrama puede ser destinada para el canal descendente, para el canal ascendente o como subtrama especial (Transición entre una subtrama Downlink y una subtrama Uplink).

Uplink / Downlink Configuration	DL to UL Switch Periodicity	Subframe Number									
		0	1	2	3	4	5	6	7	8	9
0	5 ms	D	S	U	U	U	D	S	U	U	U
1	5 ms	D	S	U	U	D	D	S	U	U	D
2	5 ms	D	S	U	D	D	D	S	U	D	D
3	10 ms	D	S	U	U	U	D	D	D	D	D
4	10 ms	D	S	U	U	D	D	D	D	D	D
5	10 ms	D	S	U	D	D	D	D	D	D	D
6	5 ms	D	S	U	U	U	D	S	U	U	D

Tabla 2. Tipos de Configuraciones para las tramas tipo 2 [26] [2]

Como se puede observar en la tabla, las configuraciones 2, 3 y 4 presentan solo 1 subtrama especial (S), lo que quiere decir que a diferencia de las otras configuraciones disponen de una subtrama adicional para la transmisión de información útil, por lo cual estas configuraciones son apropiadas para perfiles de Alto Tráfico.

La trama especial se compone de 3 campos: DwPTS (Downlink Pilot Time Slot), GP (Guard Period) and UpPTS (Uplink Pilot Time Slot), y la duración de cada uno puede variar de acuerdo con la siguiente figura:

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Special Subframe Configuration	Normal Cyclic Prefix DL			Extended Cyclic Prefix DL				
	DwPTS	GP		UpPTS	DwPTS	GP		UpPTS
0	3 sym	714 us	702 us	3 sym	679 us	667 us	1 sym	
1	9 sym	285 us	273 us	8 sym	262 us	250 us		
2	10 sym	214 us	202 us	9 sym	179 us	167 us		
3	11 sym	143 us	153 us	10 sym	95 us	83 us		
4	12 sym	71 us	59 us	3 sym	607 us	583 us		
5	3 sym	643 us	619 us	8 sym	191 us	167 us		
6	9 sym	214 us	190 us	9 sym	107 us	83 us	2 sym	
7	10 sym	143 us	119 us					
8	11 sym	71 us	47 us					

Figura 11. Especificación de la subtrama especial para cada tipo de configuración.

[2]

El Tiempo de Guarda GP (Guard Period) es necesario para permitir la transición de recepción a transmisión tanto en el UE como en eNodeB. En la figura 12 se observa la transición entre subtramas teniendo en cuenta los tiempos de propagación.

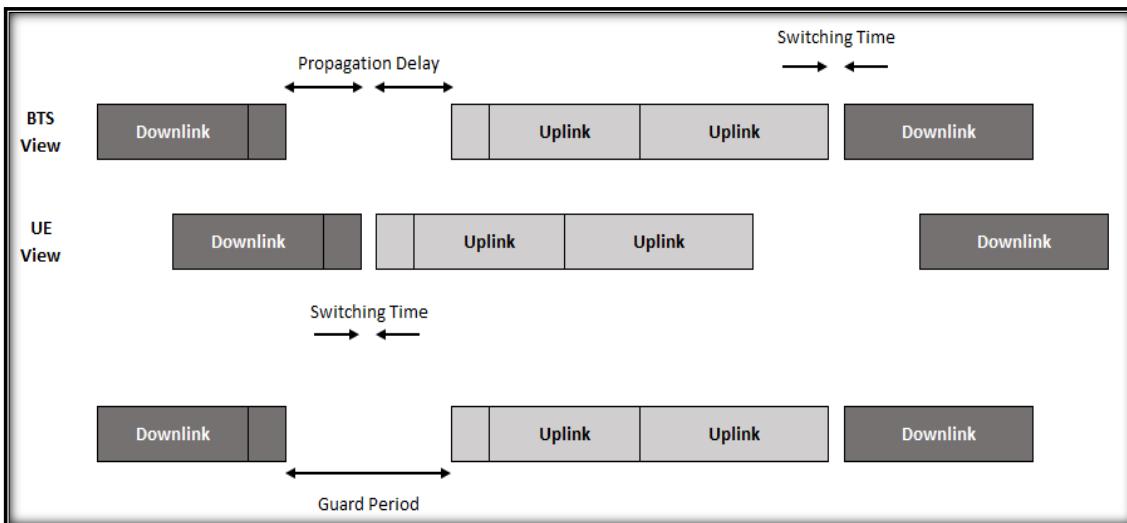


Figura 12. Tiempo de transición para las tramas DL and UL para el modo TDD entre el eNodeB y el UE. [30]

6.6 Resource Blocks

Un Resource Block es la unidad básica dentro de la interfaz de aire (comunicación UE - eNodeB). El eNodeB es el encargado de la asignación de los Resource Blocks a los UE conectados con el fin de permitir la transferencia de datos. [28]

Un Resource Block se define como un recurso en el dominio temporal y frecuencial. Un RB ocupa un periodo de 0.5ms, tiempo durante el cual se pueden transmitir 7 símbolos OFDMA (con prefijo cíclico normal) o 6 símbolos OFDMA (con prefijo cíclico extendido) por subportadora. En el dominio de la frecuencia un RB tiene 12 subportadoras, es decir un Ancho de Banda de 180KHz (12*15 KHz).

Para los Servicios Multimedia Broadcast-Multicast (MBMS) los RB ocupan un periodo de 0.5ms, sin embargo, solo pueden transmitir 3 símbolos (con prefijo cíclico extendido). A nivel frecuencial tienen 24 subportadores, sin embargo el espaciamiento entre ellas es de 7.5KHz, es decir un Ancho de Banda de 180KHz (24*7.5), igual a los dos tipos de RB anteriores.

Los Resources Elements son la unidad de recurso más pequeña y se definen como la activación de una subportadora por un periodo de tiempo suficiente para transmitir un símbolo OFDM, como se puede observar en la figura 13.

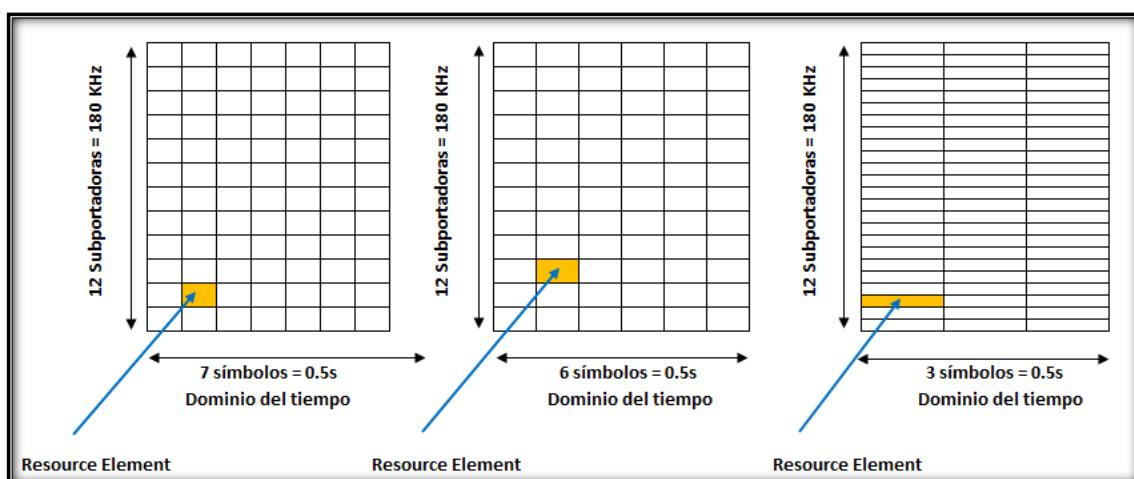


Figura 13. Diagrama de RB usando prefijo cíclico normal y prefijo cíclico extendido.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

El número de Resource Block es proporcional al ancho de banda del canal como se puede observar en la tabla 3 (Asumiendo un Ancho de Banda por subportadora igual a 15KHz).

Ancho de Banda del Canal	1.4MHz	3 MHz	5 MHz	10 MHz	15 MHz	20 MHz
Cantidad de RB	6	15	25	50	75	100
Cantidad de Subportadoras	72	180	300	600	900	1200
Ancho de Banda Útil (MHz)	1.095	2.715	4.515	9.015	13.515	18.015
Tasa de símbolos (CP normal) Msps	1.008	2.52	4.2	8.4	12.6	16.8
Tasa de símbolos (CP extended) Msps	0.864	2.16	3.6	7.2	10.8	14.4

Tabla 3. Especificación de Recursos para diferentes valores de ancho de banda [2]

Para el canal de bajada (Downlink) se tienen 3 tipos de modulación (Tabla 4).

Tipo Modulación	Bits por símbolo
QPSK	2
16 QAM	4
64 QAM	6

Tabla 4. Tipos de Modulación para en canal de bajada [2]

Donde se puede calcular el Throughput máximo de la siguiente forma:

La cantidad de símbolos durante un slot de tiempo es 7.

$$\text{Número de Símbolos por Slot} = 7$$

Si se emplea una modulación de 64 QAM, la cantidad de bits por símbolo es igual a 6.

$$\text{Número de bits por Símbolo} = 6$$

En 1 segundo de transmisión la máxima cantidad de Resource Block que se pueden asignar es 200.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

$$\text{Cantidad de slots por segundo} = \frac{1 \text{ slot}}{0.5 \text{ ms}} = 2000 \frac{\text{slots}}{\text{s}}$$

Por lo tanto, la tasa de transmisión máxima es:

$$\text{Tasa de Transmisión}_{max} = (\# \text{ de Símbolos/Slot}) * (\# \text{ de bits/Símbolo}) * (\# \text{ de slot/s}) * (\# \text{ total subportadoras})$$

$$\text{Tasa de Transmisión}_{max} = 7 \frac{\text{simbolos}}{\text{slot}} * 6 \frac{\text{bits}}{\text{simbolo}} * 2000 \frac{\text{slots}}{\text{s}} * 1200 \text{ subportadoras} = 100.8 \text{ Mbps}$$

Sin embargo, no todos los PRB son empleados para la transmisión de datos. A nivel de capa física se manejan los siguientes tipos de canales:

- PDSCH (Physical Downlink Shared Channel): Este canal es empleado para la transmisión de datos de usuario.
- PDCCH (Physical Downlink Control Channel): Este canal es empleado para la transmisión de datos de señalización, especialmente de control sobre la interfaz S1 que permite comunicar con los elementos MME y SGW (pertenecientes al EPC).
- PBCH (Physical Broadcast Channel): Este canal es empleado para transmitir la información básica de la celda (PCI, ancho de banda empleado, Hybrid ARQ indicator, Master Information Block, etc).
- PCFICH (Physical Control Format Indicator Channel): Este canal permite la decodificación de los canales de control.
- PHICH (Physical Hybrid-ARQ Indicator Channel): Este canal permite la transmisión de los Hybrid ACK para el canal de bajada (Downlink) y los ACK/NACK para el canal de subida(Uplink).
- PRACH (Physical Random Access Channel): Este canal es empleado para el intercambio de mensajes de inicio de conexión entre el UE y la celda.
- PMCH (Physical Multicast Channel): Sobre este canal se define la estructura física para llevar los servicios multimedia Broadcast y Multicast.

Para cada tipo de canal físico se definen un tipo de Modulación definido en la tabla 5.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Tipo de Canal Físico	Esquema de Modulación	Bits por símbolo modulado
PBCH	QPSK	2
PCFICH	QPSK	2
PDCCH	QPSK	2
PFICH	BPSK	1
PDSCH	QPSK, 16QAM, 64QAM	2,4,6
PMCH	QPSK, 16QAM, 64QAM	2,4,6

Tabla 5. Tipos de modulación para los canales de transporte físico. [26]

La máxima modulación alcanzada para el canal PDSCH es 64QAM con la cual se logra el mayor Throughput efectivo, sin embargo, para lograr ello requiere una alta relación Señal a Ruido (mayor a 25dB) de lo contrario la modulación deberá ser menor, proceso conocido como Modulación Adaptativa.

El mapeo aproximado de PRB empleados para cada tipo de canal se especifica en la tabla 6.

Canal	% PRB
PDSCH	88.95
PDCCH	0.11
PBCH	5.05
PCFICH	0.095
PHICH	0.92

Tabla 6. Mapeo aproximado de PRB de acuerdo con el tipo de canal de transporte físico. [2]

Basados en la tabla anterior, y considerando modulación 64QAM y tasa de codificación de 1 (la tasa más baja) se obtiene un Throughput máximo estimado de:

$$Tasa\ de\ Transmisión_{max\ real} = 100.8\ Mbps * 88.95\% = 89.65\ Mbps$$

6.7 Protocolos en Red de Acceso (E-UTRAN)

En la red de acceso (E-UTRAN) se utilizan las siguientes 3 interfaces:

- Interfaz de Aire Uu.
- Interfaz S1.
- Interfaz X2.

Las cuales soportan un plano de usuario y a un plano de control, donde el plano de usuario abarca los protocolos usados en la transmisión de los paquetes IP correspondientes a los servicios demandados por el terminal (UE) en la red y el plano de control abarca los protocolos que sustentan la ejecución (señalización) del plano de usuario. [31]

A continuación, se explica brevemente la pila de protocolos para las 3 interfaces mencionadas.

6.7.1 Protocolos en la Interfaz de Aire

Esta interfaz es la que permite el envío de los paquetes IP entre el UE y el eNodeB. La pila de protocolos se evidencia en la figura 14.

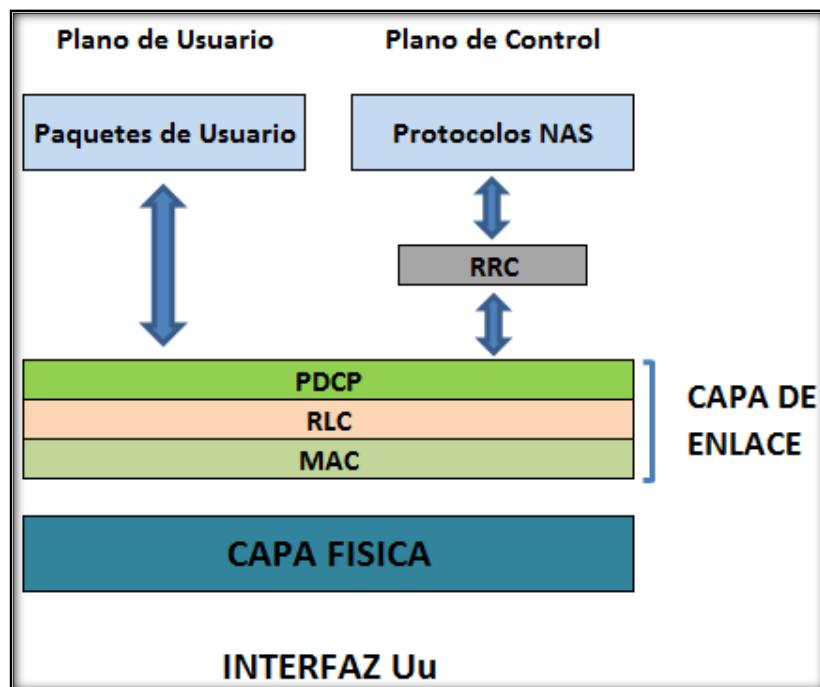


Figura 14. Pila de protocolos para la interfaz Uu [31]

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- Protocolo de convergencia de paquetes de datos (PDCP): Constituye la capa superior de la pila, y es el encargado de proporcionar el acceso al servicio portador (Bearer) de radio, es decir que los paquetes IP se reciben y se envían a través del servicio de transferencia de la capa PDCP. Entre las funciones de esta capa se encuentran la compresión y descompresión de la cabecera de los paquetes IP, cifrado de la información (payload) de los paquetes IP con el fin de garantizar la confidencialidad de la información transmitida y la adición de un número de secuencia dentro de la cabecera PDCP que permite ordenar debidamente los paquetes y en algunos casos detectar paquetes duplicados.
- Capa de control de enlace vía Radio (RLC): Esta capa permite la transmisión confiable de la información entregada por la capa PDCP. Entre las funciones principales se encuentran la corrección automática de errores, segmentación y ensamblado de paquetes PDCP.
- Capa de control de acceso al medio (MAC): Es la capa más importante en la interfaz de aire. Permite manejar el mecanismo de scheduling entre el eNodeB y los diferentes UE conectados a la celda. Realiza la multiplexación de los paquetes RLC en los diferentes canales de transporte relacionados en la capa física. Los servicios que la capa MAC ofrece sobre la capa RLC se denominan canales lógicos.
- Capa Física: Esta capa es la que permite realizar la transmisión de la información entre el UE y el eNodeB a través de la emisión y recepción de señales electromagnéticas. Entre sus funciones se encuentran la codificación del canal, la elección del tipo de modulación, y el proceso relacionado con el método de acceso al medio (OFDMA para el canal de bajada y Single Carrier-FDMA para el canal de subida).

Para el plano de control se añade un protocolo adicional.

- Capa de Control de Recursos de Radio (RRC): Esta capa permite establecer un canal de control entre el eNodeB y el UE. Entre las funciones principales de esta capa se encuentran la gestión de los servicios portadores por cada UE, funcionalidades relacionadas con los procesos de Handover (movilidad), difusión de parámetros del sistema y envío de mensajes de aviso a los terminales que no tienen una conexión activa con el eNodeB (mensaje de paging).

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- Protocolos NAS: Dichos protocolos se extienden al Evolved Packet Core (comunicación entre el UE y el MME). La información de dichos protocolos se transmite de forma transparente por la interfaz de aire sobre el campo de payload del protocolo RRC.
- Entre las principales funciones de dichos protocolos se encuentran la autenticación del UE en la red, autorización, gestión de la movilidad para aquellos UE que no tienen una conexión RRC activa y la gestión de los servicios portadores (bearers) del EPC.

6.7.2 Protocolos en la Interfaz S1 y X2

Los mensajes sobre las interfaces S1 y X2 se soportan sobre diferentes tecnologías de transmisión como SDH, GPON, entre otras por lo cual no se definen protocolos específicos para la capa física y cada de enlace. Para el plano de usuario sobre la capa IP se añade el protocolo GTP-U (GPRS Tunneling Protocol), el cual es utilizado en tecnologías anteriores como GSM y UMTS y el cual es transportado sobre el protocolo UDP. La finalidad de este protocolo es poder realizar la multiplexación de los paquetes IP de varios UE, mediante el uso de etiquetas para diferenciar los servicios de cada portador (bearer). Para el plano de control se hace uso del protocolo de transporte SCTP (Stream Control Transmission Protocol), el cual fue concebido para el envío de señalización sobre redes telefónicas, ofreciendo mecanismos de control de flujo, congestión en la conexión, soporte para multihoming (asignación de varias IP para un único UE), soporte para multistreaming, envío de múltiples flujos a un mismo UE, entre otras funciones. Las pilas de protocolos se pueden observar en la figura 15 y 16.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

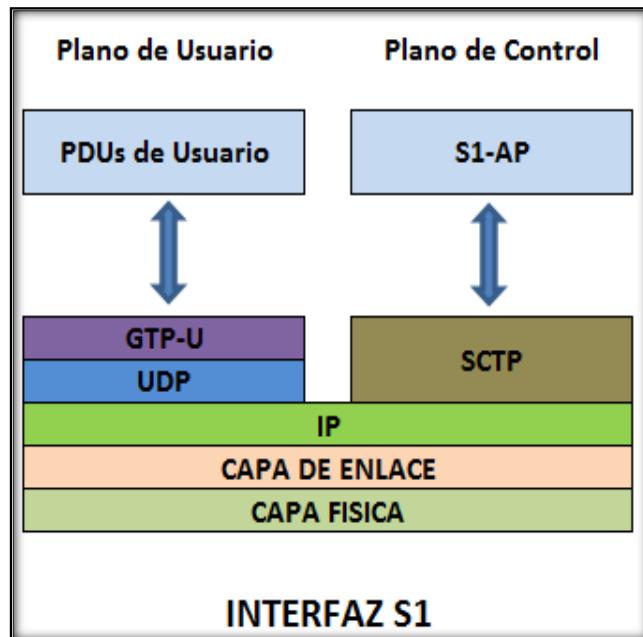


Figura 15. Pila de protocolos para la interfaz S1. [31]

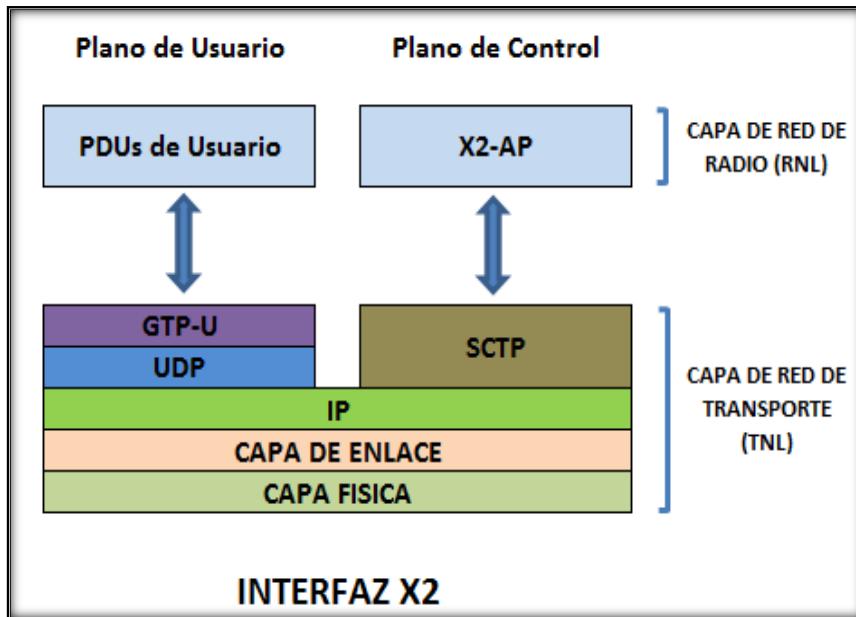


Figura 16. Pila de protocolos para la interfaz X2. [31]

En general para los dos planos mencionados se puede establecer la relación entre las pilas de protocolos. En la figura 17 se evidencia el flujo para el plano de usuario y en la figura 18 el flujo para el plano de control.

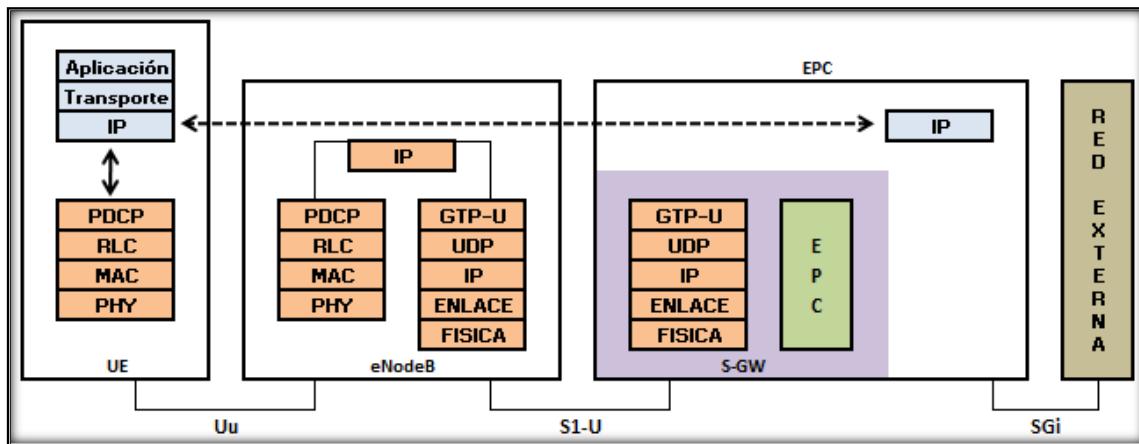


Figura 17. Flujo de protocolos para el plano de usuario [31]

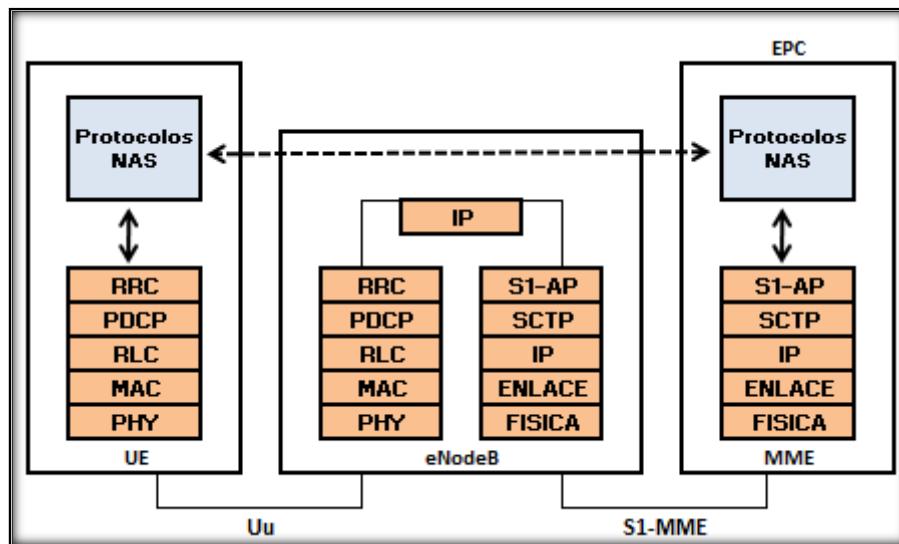


Figura 18. Flujo de protocolos para el plano de control [31]

6.8 Handover

Una de las principales características en un sistema comunicación móvil es la Movilidad. La movilidad es la capacidad que debe tener la red para garantizar el servicio a pesar de que los usuarios se encuentren en movimiento. Para que se pueda garantizar un nivel de QoS a un usuario específico, se debe contar con un nivel de potencia de recepción RSRP mínimo (Reference Signal Received Power) cuyo valor es inversamente proporcional a la distancia

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

entre el sistema radiante y el UE (User Equipment). Si un usuario se encuentra conectado a una celda pero su nivel de RSRP está por debajo del umbral configurado el eNodeB, el usuario deberá terminar la conexión con dicha celda y conectarse a una nueva celda con mejores niveles de potencia. [32]

El handover es el proceso en el cual se transfiere el servicio de una celda a otra cuando la calidad del enlace es insuficiente en una de las estaciones. Este mecanismo garantiza la entrega del servicio cuando un móvil se traslada a lo largo de su zona de cobertura.

Se tienen los siguientes escenarios relacionados con el proceso de Handover:

1. Movilidad Intra-LTE: El handover se ejecuta entre estaciones LTE (E-UTRAN).
2. Movilidad entre redes de acceso 3GPP: El handover se ejecuta entre redes LTE (E-UTRAN), UMTS (UTRAN) y GSM (GERAN) y se conoce con el nombre de Inter-RAT Handover.
3. Movilidad con redes de acceso No 3GPP: El handover se ejecuta entre redes LTE (E-UTRAN) y redes CDMA2000, WLAN o Mobile Wimax.

En el estándar LTE se especifica un Modelo diseñado para la gestión de la movilidad denominado EMM (EPS-Mobility Management), que puede presentar los siguientes estados:

- EMM-Deregistered: En estado el usuario no se encuentra visible para la red por lo cual no podría tener acceso a los servicios de esta. Generalmente este estado se produce cuando el terminal (UE) se encuentra apagado.
- EMM-Registered: En estado el usuario se encuentra visible para la red y puede acceder a los servicios de esta. La localización del usuario viene dada por la resolución de al menos un Tracking Area e igualmente posee al menos una conexión activa con el PDN y un MME asignado encargado de su seguimiento.

Adicional al modelo de movilidad EMM, se define otro modelo de estados para poder validar la existencia o no de un plano de control entre el UE y el MME, el cual se conoce como ECM (EPS Connection Management). Los estados posibles son los siguientes:

- ECM-Idle: En este estado el UE no tiene establecida una conexión de control (señalización) con ningún MME del EPC. La existencia de este estado permite la

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

rápida conmutación hacia el estado conectado (Connected) y que a su vez se disponga de un modo de operación de bajo consumo.

- ECM-Connected: En este estado el UE tiene establecida una conexión de control con algún MME del EPC. Esta Bearer se compone de una conexión RRC a nivel de radio (UE – eNodeB) y de una conexión hacia el MME a través de la interfaz S1.

La información relacionada a los estados de los modelos EMM y ECM se almacena tanto en las entidades del EPC como en el UE. Para el UE estos estados se mapean directamente sobre el protocolo RRC como RRC_Idle y RRC_Connected.

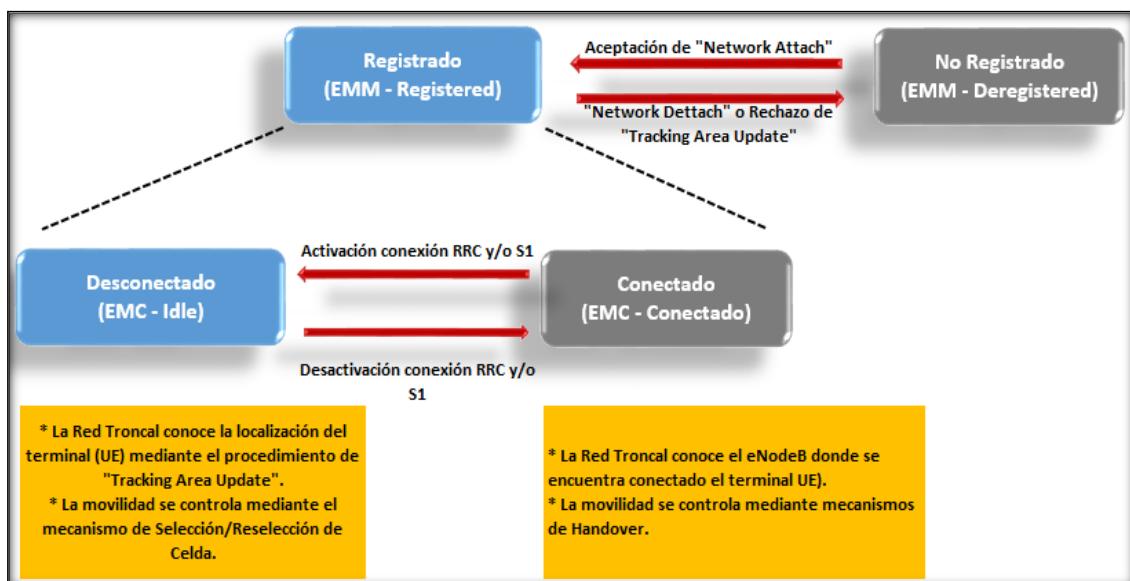


Figura 19. Relación entre los estados de EMM y EMC. [28]

En la figura 20 se evidencia el ciclo de un proceso de Handover. Inicialmente el UE se encuentra en modo **Connected** con la celda A, a medida que el usuario se aleja el nivel de RSRP de la celda A empieza a disminuir y empieza a aumentar el nivel de RSRP de la celda vecina en este caso la celda B. Inmediatamente el usuario sobrepasa la zona de borde celda empieza a experimentar mejores niveles proveniente de la celda vecina, después de ello es necesario que la diferencia de entre los 2 niveles de RSRP sobre pase el valor de HO Hysteresis (valor configurable) para proceder a traspasar los servicios desde la celda A hacia la celda B. El tipo de Handover ejecutado en LTE es del tipo Hard Handover, es decir que se presenta un pequeño grado interrupción de servicio mientras se realiza el cambio de celda Origen.

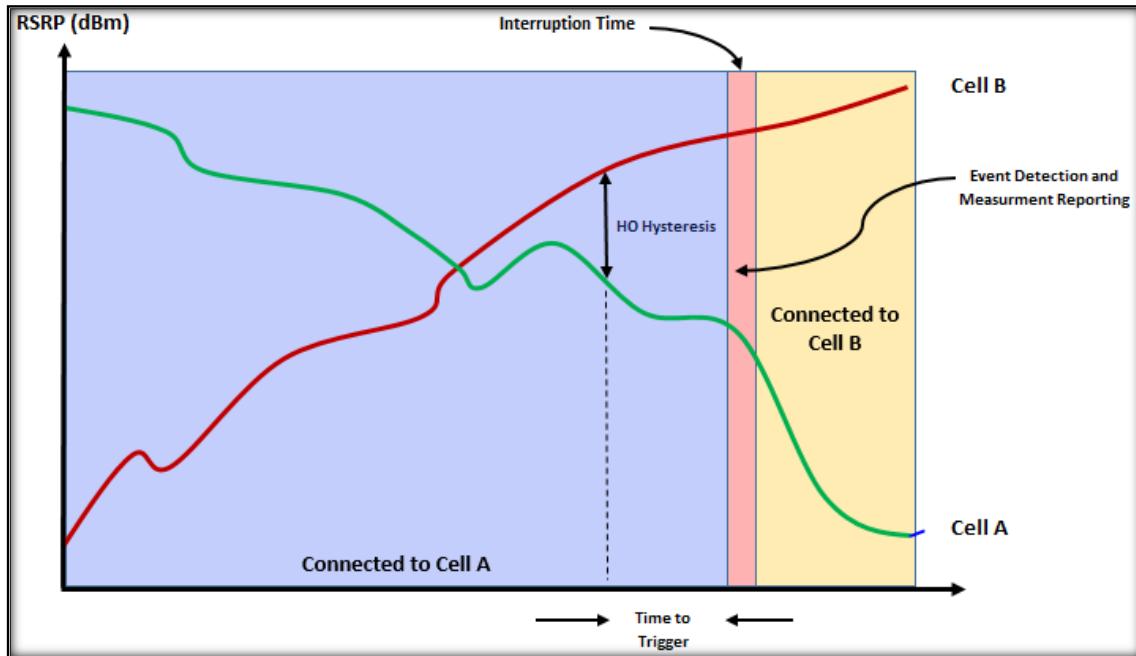


Figura 20. Ciclo de un proceso de Handover [28]

En LTE los handover son del tipo **hard handover**, es decir que todos los datos pendientes por enviar al UE durante el proceso de handover son transferidos desde el eNodeB origen hacia el eNode destino para evitar pérdida de información.

A continuación, se explica la señalización asociada al proceso de Handover.

6.8.1 Preparación del Handover (Handover Preparation)

En la Figura 21 se observa la señalización correspondiente a la preparación del proceso de Handover. Básicamente consiste en el envío de mensajes periódicos desde el UE hacia el eNodeB origen indicando el valor actual de RSRP, con esta información el eNodeB evalúa si la potencia está por debajo del umbral mínimo y de ser así envía una solicitud al eNodeB destino para dar servicio al UE respectivo. El eNodeB destino evalúa bajo ciertas políticas de admisión y responde al eNodeB origen la solicitud.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

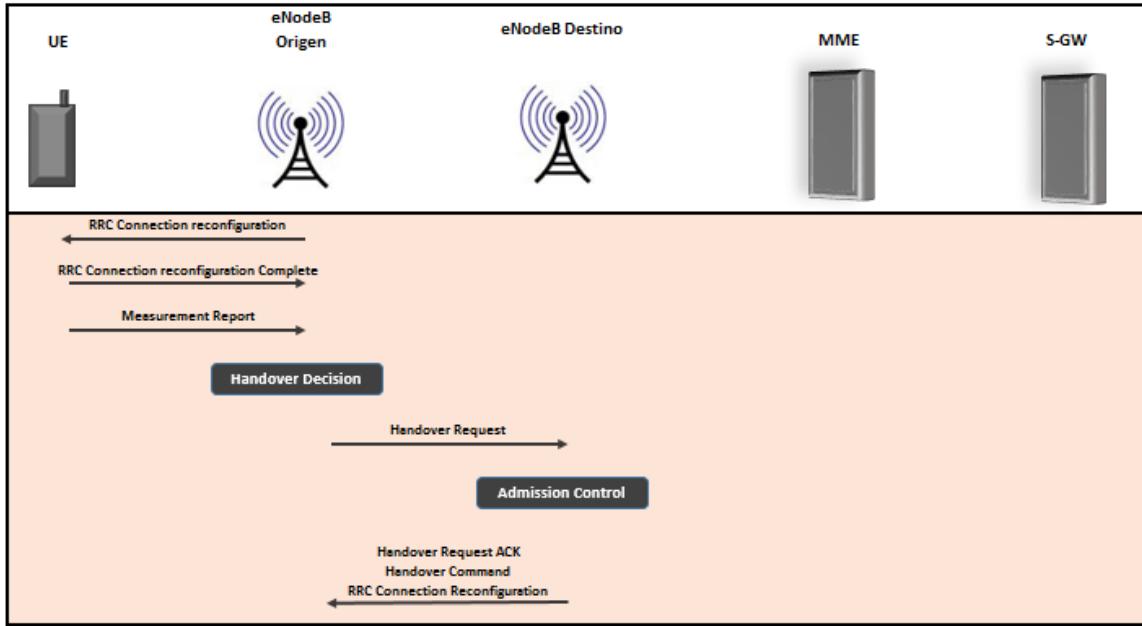


Figura 21. Preparación del Handover [2]

6.8.2 Ejecución del Handover (Handover Execution)

En la Figura 22 se observa la señalización correspondiente a la ejecución del proceso de Handover. Si el eNodeB destino responde al eNodeB origen indicando que puede dar servicio al UE, el eNodeB origen envía un mensaje al UE indicándole que el eNodeB destino será procesará los paquetes desde dicho instante.

El eNodeB origen envía los paquetes que no se han procesado aún del UE respectivo al eNodeB destino por medio de la interfaz X2 y se inicia el proceso de RRC connection entre el UE y el eNodeB destino.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

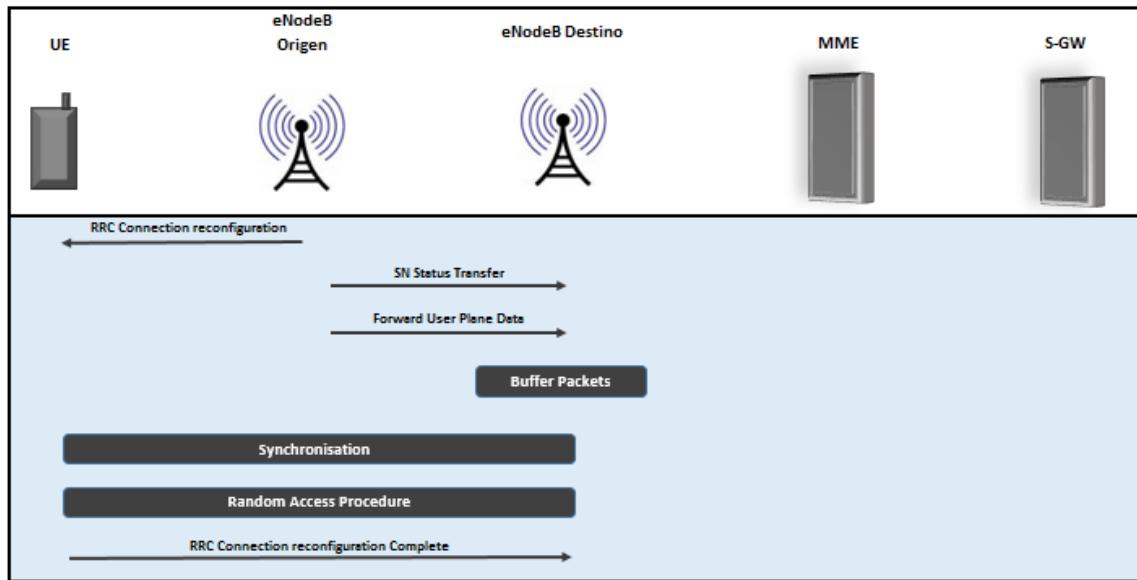


Figura 22. Ejecución del Handover [2]

6.8.3 Finalización del Handover (Handover Completion)

En la Figura 23 se observa la señalización correspondiente a la finalización del proceso de Handover. En esta etapa se modifica el bearer entre el UE y el EPC indicando que el eNodeB ha sido actualizado y se inician a procesar los paquetes del UE dando por finalizado el handover.

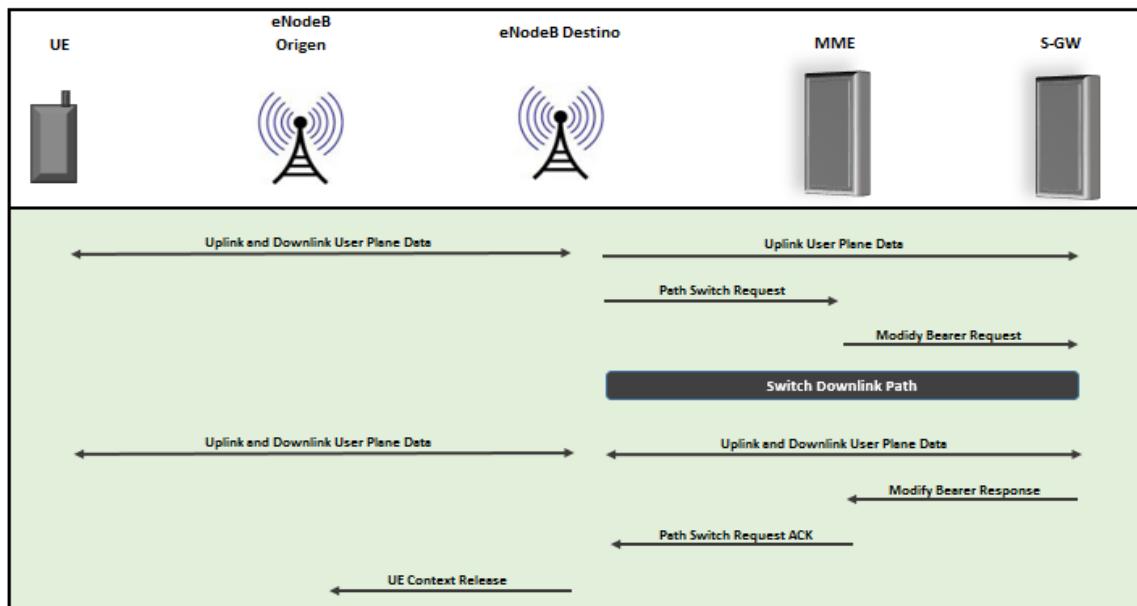


Figura 23. Finalización del Handover [2]

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

A continuación, se explica de forma detallada el contenido de los mensajes enviados en el proceso de Handover evidenciados en la figura 21, 22 y 23.

Se inicia con la configuración en el UE para enviar las mediciones periódicas al eNodeB, mediante los mensajes de RRC Connection Configuration. Estas mediciones reportan generalmente los eventos A3 y A5 los cuales son:

- Evento A3: La celda vecina presenta mayor nivel de potencia que la celda actual.
- Evento A5: La celda actual presenta niveles de potencia por debajo del umbral (treeshold) 1, y la celda vecina presenta niveles de potencia mayores al umbral (treeshold) 2.

El UE envía un mensaje reportando los niveles de potencia al eNodeB cuando alguno de los eventos mencionados anteriormente se presenta o cuando el tiempo configurado periódicamente para el envío del reporte expira. El contenido del mensaje de Reporte de Medición se puede observar en la figura 24.

Information Elements			
Measurement Identity			
Primary Serving Cell Measurement Results		RSRP	
		RSRQ	
Physical Cell Identity			
Neighboring Cell Measurement Results	E_UTRAN Result List	CGI Information	
		Cell Global Identity	
		Tracking Area Code	
		PLMN Identity List	
		Measurement Results	
		RSRP	
		RSRQ	
		Additional System Information	
Measurement Result for ECID		CSG Membership Status	
		CSG Identity	
UE Rx-Tx Time Difference Result			
Current SFN			
Location Information			
Serving Frequency List Measurement Results	Serving Freq Identity	Serving Cell Index	
	Secondary Serving Cell Measurement Results	RSRP	
	Best Neighbour Cell Measurement Results	RSRQ	
		Physical Cell Identity	
		RSRP	
		RSRQ	

Figura 24. Mensaje de reporte de medición [26] [2]

Estos reportes se identifican mediante un ID, el cual permite relacionarlo con el mensaje RRC Connection Reconfiguration y poder extraer la información correspondiente a la configuración. Las celdas actuales siempre están incluidas en los reportes de medición enviados por los UE, lo cual le permite evaluar la cobertura y compararla con respecto las coberturas de las celdas vecinas.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

La lista de E-UTRAN sobre los resultados de mediciones identifica a cada celda vecina con su respectivo PCI (Physical Cell ID).

Las medidas RSRP y RSRQ de celdas vecinas son reportadas para permitir al eNodeB evaluar su cobertura. Si el eNodeB puede identificar una celda vecina candidata a través de su PCI y si las condiciones de cobertura sugieren que se debe ejecutar el handover. El eNodeB actual inicia el proceso de Handover enviando a la celda destino un mensaje de solicitud de Handover (Handover Request) a través de la interfaz X2. La estructura de dicho mensaje se puede observar en la figura 25.

Information Elements		Presence
Message Type		Mandatory
Old eNodeB UE X2AP ID		Mandatory
Cause		Mandatory
Target Cell ID		Mandatory
GUMMEI		Mandatory
UE Context Information	MME UE S1 AP ID	Mandatory
	UE Security Capabilities	Mandatory
	AS Security Information	Mandatory
	UE Aggregate Maximum Bit Rate	Mandatory
	Subscriber profile ID for RAT/Freq priority	Optional
	E-RAB Identity	Mandatory
	E-RAB Level QoS parameters	Mandatory
	Downlink Fowarding	Optional
	Uplink GTP Tunnel Endpoint	Mandatory
	RRC Context	Mandatory
	Handover Restriction List	Optional
	Location Reporting Information	Optional
	Management based MDT allowed	Optional
	UE History Information	Mandatory
	Trace Activation	Optional
	SRVCC Operation Possible	Optional
	CSG Membership Status	Optional

Figura 25. Mensaje de Solicitud de Handover [26] [2]

- El campo Cause dentro del mensaje es generalmente configurado como Handover deseado por Razones de Radio (Handover Desirable for Radio Reasons) sin embargo, hay otras razones como lo son Tiempo crítico de Handover (Time CriticalHandover), Handover por Optimización de Recursos (ResourceOptimisationHandover) or Reducción de carga de la celda actual (Reduce Load in Servig Cell).
- La identidad de la celda destino es identificada como E-UTRAN CGI. Esta información es extraída de la base de datos del eNodeB las celdas vecinas a partir del PCI. El procedimiento de Relación Automática de Celdas Vecinas (ANR) es

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

ejecutado si el PCI y el correspondiente CGI no puede ser encontrados dentro de la base de datos.

- El identificador único global MME (GUMMEI) es incluido en el mensaje para informar al eNodeB destino de cuál es el MME actual de los UE a migrar de una celda a otra. Esto permite al eNode destino direccionar el mensaje de Solicitud de Cambio de Ruta S1-AP al MME apropiado después de ejecutar el Handover.

El campo UE ContextInformation informa al eNodeB destino cuáles bearers necesitan ser establecidos para soportar los handover entrantes. El conjunto de E-RAB pertenecientes al UE es especificado con sus correspondientes requerimientos de Calidad de Servicio (QoS).

El eNodeB destino aplica su propio control de admisión después de recibir la solicitud de Handover, enviada por el eNode origen. Asumiendo que el eNodeB destino puede aceptar dicha solicitud, éste retorna un mensaje de respuesta que tiene la estructura evidenciada en la figura 26.

Information Elements		Presence
	Message Type	Mandatory
	Old eNodeB UE X2AP ID	Mandatory
	New eNodeB UE X2AP ID	Mandatory
E-RAB Admitted List	E-RAB Identity	Mandatory
	Uplink GTP Tunnel Endpoint	Optional
	Downlink GTP Tunnel Endpoint	Optional
	E-RAB Not Admitted List	Optional
Target eNodeB to Source eNodeB Transparent Container		Mandatory
Criticality Diagnostics		Optional

Figura 26. Mensaje de respuesta al mensaje de solicitud de Handover [26] [2]

El eNodeB destino usa el mensaje ACK para informar al eNodeB origen de cuales E-RAB han sido admitidos. Los endpoint del túnel GTP son especificados para permitir al eNodeB reenviar la información de plano de usuario (canal de subida y bajada) a través de la interfaz X2.

El container entre el eNodeB origen y el eNodeB destino incluye un mensaje de comando de Handover. Dicho mensaje encapsula un mensaje de reconfiguración de una conexión RRC el cual es generado por el eNodeB destino y enviado al eNodeB origen para posteriormente ser enviado al UE por medio de la interfaz de aire (uU), donde se proveen las indicaciones

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

para poder completar el proceso de handover. El mensaje de reconfiguración de una conexión RRCse puede ver en la figura 27.

Information Elements	
Target Physical Cell ID	
Carrier Frequency	Uplink Carrier Frequency Downlink Carrier Frequency
Carrier Bandwidth	
Additional Spectrum Emission Requirements	
T304	
New UE Identity	C-RNTI RACH Configuration PRACH Configuration PDSCH Configuration PUSCH Configuration PHICH Configuration PUCCH Configuration Sounding Reference Signal Configuration Uplink Power Control Antenna Information Pmax Uplink Cyclic Prefix Length
Common Radio Resource Configuration	Random Access Preamble Index Random Access PRACH Mask Index
Dedicated RACH Configuration	

Figura 27. Mensaje de reconfiguración de una conexión RRC [26] [2]

El PCI es especificado directamente al UE. Las subportadoras para el canal de subida y de bajada son especificadas en términos de sus E-UTRAN ARFCN (absolute radio-frequencychannelnumber). El ancho de banda del canal también es especificado junto a requerimientos adicionales correspondientes al espectro.

El T304 es un timer cuyo valor puede ser 50, 100, 120, 200, 500, 1000 o 2000ms. Este timer se inicia una vez el UE recibe el mensaje sobre control de movilidad, si el timer expira antes de que se complete el proceso de acceso aleatorio (Random Access) en la celda destino, el UE inicia el restablecimiento de una conexión RRC con la celda.

El eNodeB destino asigna un nuevo C-RNTI al UE, el cual es un identificador dentro de la celda con el fin de diferenciar a cada UE.

La configuración común de recursos de radio (Common Radio ResourcesConfiguration) provee información correspondiente a varios canales físicos y de transporte en la celda

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

destino. En este campo también se especifica la máxima potencia de transmisión en el canal de subida y la longitud del prefijo cíclico en el canal de subida.

La configuración del canal de acceso aleatorio (Dedicated RACH) provee soporte para permitir el procedimiento de acceso aleatorio basado en no contención (non-contention based). Al UE se le asigna una secuencia de preámbulo dedicada, lo cual hace más confiable todo el proceso de handover.

El eNodeB origen procede a enviar un mensaje de estatus de la transferencia del Número de secuencia al eNodeB destino sobre la interfaz X2. Este mensaje provee al eNodeB destino información relacionada al número de secuencia PDCP para el canal de subida y de bajada, lo cual permite asegurar que este número de secuencia sea preservado durante el proceso de handover. Este mensaje tiene la siguiente estructura:

Information Elements		Presence
	Message Type	Mandatory
	Old eNodeB UE X2AP ID	Mandatory
	New eNodeB UE X2AP ID	Mandatory
E-RAB Subject to Status Transfer List	E-RAB Identity	Mandatory
	Receive Status of Uplink PDCP SDU	Optional
	Uplink COUNT value	Optional
	Downlink COUNT value	Optional

Figura 28. Mensaje de estatus de transferencia del número de secuencia [26] [2]

El campo opcional ReceiveStatusofUplink PDCP SDU define un conjunto de bits para diferenciar cuales SDU en el canal de subida han sido recibidos correctamente y cuáles no.

El Uplink Count define el número de secuencia PDCP y el número de HyperFrame (HFN) del primer SDU faltante en el canal de subida. El Downlink Count define el número de secuencia PDCP y el número de HyperFrame (HFN) que el eNodeB destino debería asignar al siguiente SDU del canal de bajada al cual no se la ha asignado un número de secuencia.

El UE procede a sincronizarse con el eNodeB destino a través de la interfaz de aire mediante el uso de las señales de sincronización primarias y secundarias. Posterior a ello, se inicia el procedimiento de Acceso Aleatorio entre el UE y la celda destino finalizando con un mensaje de reconfiguración de la conexión RRC.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

En este punto los datos pendientes por transmitir tanto para el canal de bajada como para el canal de subida pueden ser reenviados entre el UE y la celda destino.

Finalmente, después de recibir el mensaje completo de reconfiguración de la conexión RRC, el eNodeB destino inicia el proceso para cambiar el túnel GTP del canal de bajada modificando el eNode origen por el eNodeB destino. Esto se logra mediante el envío de un mensaje solicitud de cambio de ruta (PathSwitchRequest) al MME, el cual tiene la siguiente escritura (figura 29):

Information Elements		Presence
Message Type		Mandatory
eNodeB UE S1 AP ID		Mandatory
E-RAB to be switched in Downlink List	E-RAB Identity	Mandatory
	Transport Layer Address	Mandatory
	GTP-TEID	Mandatory
Source MME UE S1 AP ID		Mandatory
EUTRAN CGI		Mandatory
TAI		Mandatory
UE Security Capabilities		Mandatory
CSG Identity		Optional
Cell Access Mode		Optional
Source MME GUMMEI		Optional

Figura 29. Mensaje de Solicitud de cambio de Ruta [26] [2]

El MME decide si existe o no un Serving Gateway que permita proveer servicio al UE. Asumiendo que el Serving Gateway no cambia, el MME envía un mensaje de solicitud de modificación del Bearer (GTP v2 C) al Serving Gateway, el cual procede a cambiar la ruta y enviar los datos del canal de bajada hacia la nueva celda.

Por último el eNodeB completa el procedimiento de handover enviando un mensaje de liberación (UE ContextRelease) al eNodeB origen, en donde informa que puede liberar los recursos que tenía dedicados el/los UE que fue trasladado a la nueva celda.

6.9 Mobility Load Balancing (Balanceo de Carga Móvil)

El Feature de Mobility Load Balancing fue incorporado en el estándar de la 3GPP sobre el release 9 y con una versión mejorada en el release 10 [26] [33]. El objetivo de MLB es poder

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

balancear la carga sobre la red, mediante el traslado de usuarios de borde de celda desde celdas congestionadas a celdas con recursos de radio disponibles.

MLB se lleva a cabo mediante el intercambio de información de la carga actual entre las celdas, a partir de lo cual se pueden variar los umbrales de inicio de Handover de acuerdo con el balanceo que se desee ejecutar.

La información intercambiada entre celdas vecinas se evidencia en la figura 30.

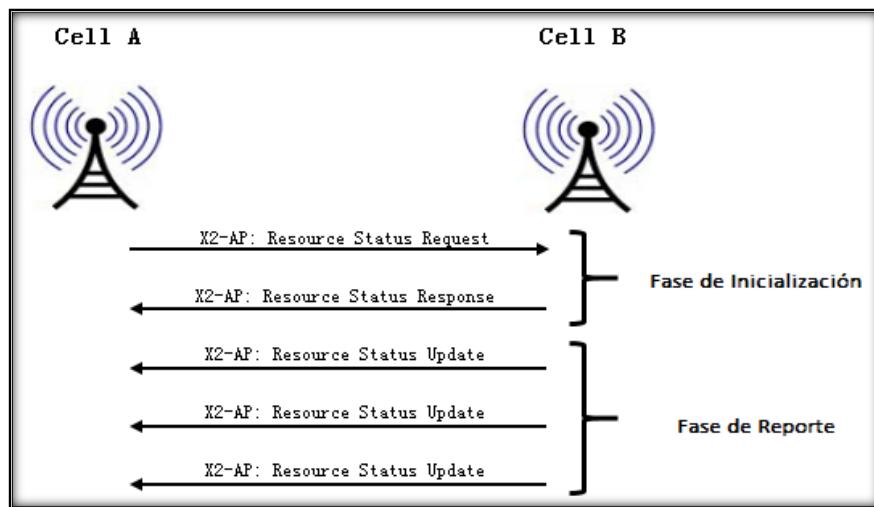


Figura 30. Intercambio de mensajes de señalización [33] [2]

El mensaje de Request tiene la estructura que se observa e la figura 31:

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Information Elements	
Cell Measurement 1 Identity	
Cell Measurement 2 Identity	
Registration Request	
Report Characteristics	PRB Load Indication Periodic
	TNL Load Indication Periodic
	Hardware Load Indication Periodic
	Composite Available Capacity Periodic
	ABS Status Periodic
Cells to Report	LIST
	Cell Identity
Reporting Periodicity	
Partial Success Indicator	

Figura 31. Mensaje de request en el proceso de MLB [33] [2]

- La medida Cell Measurement 1 Identity es asignada por la celda origen.
- La medida Cell Measurement 2 Identity no está incluida dentro del mensaje de Request del estado actual de Recursos. Es asignada por la celda destino.
- La solicitud de Registro es usada para invocar tanto el inicio como finalización de las medidas en la celda destino.
- El listado de las mediciones dentro del mensaje es el siguiente:
 - PRB (Physical Resource Block Load Indication): Cantidad de RB que se encuentran en uso en el momento de la medición en una determinada celda.
 - TNL (Transport Network Layer Load Indication): Porcentaje de Recursos usados a nivel de capa de Transporte en el momento de la medición en una determinada celda.
 - Hardware Load Indication: Indicación del uso a nivel de hardware en el momento de la medición en una determinada celda.
 - Composite Available Capacity: Capacidad disponible por celda en el momento de la medición.
- Almost Blank Subframes (ABS) Status: Estado de las subtramas vacías.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

- El campo de Cells to Report, permite indicar la lista de eNodeB destino a los que se enviarán los mensajes de Respuesta (Response).
- La periodicidad del reporte puede ser definida en el eNodeB. Este valor puede ser de 1, 2, 5 o 10 segundos.
- El campo Partial Success Indicator permite indicar cuales mediciones se realizaron exitosamente y cuáles no.

El mensaje de Respuesta (Response) toma como base el mensaje de Solicitud (Requested) y añade el campo eNodeB 2 Measurement Identity. Esto permite conocer cuales mensaje de Solicitud no fueron respondidos por loseNodeB destino.

El mensaje de Actualización (Update) tiene el siguiente formato (figura 32):

Information Elements		
eNodeB 1 Measurement Identity		
eNodeB 2 Measurement Identity		
LIST		
Cell Identity		
Cells To Report	Hardware Load Indication	Downlink Load
		Uplink Load
	S1 TNL Load Indication	Downlink Load
		Uplink Load
	Radio Resource Status	Downlink GBR PRB usage
		Uplink GBR PRB usage
		Downlink Non-GBR PRB usage
		Uplink Non-GBR PRB usage
		Downlink Total PRB usage
		Uplink Total PRB usage
Composite Available Capacity	Downlink	Cell Capacity Class
		Capacity Value
	Uplink	Cell Capacity Class
		Capacity Value
ABS Status	Downlink ABS Status	
	Usable ABS Pattern Information	

Figura 32. Mensaje de Actualización en el proceso de MLB [33] [2]

Este mensaje contiene información similar a la del mensaje de Solicitud.

- El campo Radio Resources Status provee información sobre el canal de subida y bajada con respecto a:

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

- Uso actual de los PRB para tráfico con marca GBR (Guarantee Bit Rate): Arroja un valor (porcentaje) sobre todos los PRB disponibles usados para transmisiones DTCH con identificadores de clase de QoS (Quality of Service) – QCI 1,2,3,4.
 - Uso actual de los PRB para tráfico con marca non-GBR (non-Guarantee Bit Rate): Arroja un valor (porcentaje) sobre todos los PRB disponibles usados para transmisiones DTCH con identificadores de clase de QoS (Quality of Service) – QCI 5,6,7,8 o 9.
 - Uso total de PRB: Corresponde al valor porcentual de todos los PRB disponibles para la transmisión de subida y la transmisión de bajada.
-
- El campo Composite Available Capacity provee la información sobre los recursos disponibles globales en la celda:
 - La inclusión del campo Cell Capacity Class es opcional. Provee una medición de la capacidad de la celda usando un valor entre 1 y 100. 1 denota baja capacidad y 100 la más alta capacidad.
 - La opción Capacity Value provee un valor entre 0 y 100. 0 denota que la celda no cuenta con recursos disponibles, mientras que 100 denota que la celda cuenta con todos los recursos disponibles.
 - El campo Almost Blank Subframe Status puede ser usado por el eNodeB origen para evaluar si existe o no un parón de ABD que bería ser modificado.

El mensaje de Solicitud de Cambio de Movilidad (Mobility Change Request Message) es usado para hacer la petición de ajuste de umbrales de Handover al eNode destino. Dicho mensaje tiene la estructura observada en la figura 33.

Information Elements	
eNodeB 1 Cell Identity	
eNodeB 2 Cell Identity	
eNodeB 1 Mobility Parameter	Hand Trigger Change
eNodeB 2 Proposed Mobility Parameter	Hand Trigger Change
Cause	

Figura 33. Estructura del mensaje de solicitud de cambio de movilidad [33] [2]

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- Los campos eNodeB 1 y eNodeB 2 Cell Identity son usados para identificar las 2 celdas involucradas en el proceso de balanceo.
- El campo eNodeB 1 Mobility Parameter es opcional. Este campo especifica un cambio para el umbral del handover para el cual el eNodeB origen ha aplicado por handovers para el eNode origen y eNodeB destino. El ajuste para el disparo del Handover es señalizado con un valor entre -20 y 20, de tal manera que el valor real se calcula de la siguiente forma:

$$\text{Actual Value} = \text{Signalled Value} * 0.5dB$$

Actual Value, es un valor sugerido para ajustar el umbral de Handover.

- El campo eNodeB 2 Proposed Mobility Parameter especifica un cambio sugerido en el umbral de los handovers llevados a cabo desde el eNodeB destino al eNode Origen. El ajuste para el disparo del Handover es señalizado con un valor entre -20 y 20.
- El campo Cause especifica la razón por la cual se solicita el ajuste de umbral para el disparo del Handover, por ejemplo Load Balancing.

Si el mensaje de Solicitud de Cambio de Movilidad es aceptado, el eNodeB destino responde con un mensaje de Acknowledge (ACK). Si por el contrario el campo Proposed Handover Triggering Threshold es rechazado el eNode destino envía un Mensaje de falla al eNodeB origen. Sobre este mensaje se especifica el límite superior e inferior disponible para ejecutar el cambio de umbral.

7. Metodología

En el presente trabajo de grado se llevaron a cabo los siguientes pasos metodológicos:

7.1 Método de Modelación

A partir de la información que se tomó como base para la construcción del marco teórico, el estado del arte y la literatura relacionada con el estudio probabilístico aplicado en sistemas de comunicaciones se desarrolla un modelo analítico que permite estimar el número promedio de paquetes en una celda bajo ciertas tasas de llegada y procesamiento, así mismo se obtiene la condición de estabilidad sobre la cual el sistema se estabiliza en un periodo estacionario.

Adicionalmente se obtiene un modelo que permite estimar la tasa de bloqueo para el tráfico de voz en escenarios con y sin balanceo de carga. Este modelo permite realizar una comparación teórica entre ambos escenarios y servir de base para poder validar el funcionamiento del simulador para escenarios más complejos.

7.2 Método de Experimentación

Después de tener como base el modelo analítico que permite estimar número promedio de paquetes en el sistema y tasas de bloqueo para el tráfico de voz, se diseña una herramienta de simulación que permite incrementar la complejidad de los escenarios y poder obtener resultados más interesantes. Después de diseñar e implementar el simulador, el primer paso es confrontar los resultados de este con los resultados que arroja el modelo analítico con el fin de evidenciar la validez de la herramienta de simulación.

Finalmente se procede a incrementar la complejidad de los escenarios (mayor cantidad de celdas, mayor cantidad de usuarios, mayor cantidad de canales, entre otros), con el fin organizar, procesar los datos y proceder con el análisis respectivo. Dicho análisis incluye el uso de herramientas estadísticas que permiten consolidar los resultados finales del trabajo desarrollado.

8. Desarrollo de la propuesta

En la primera parte de este capítulo se explicarán conceptos claves que permitirán comprender la propuesta del mejoramiento de balanceo de carga planteada y con base en ello generar un modelo matemático sobre un escenario acotado bajo ciertas restricciones descritas más adelante.

8.1 Definición Tipo de Usuarios

Los usuarios conectados a una celda se clasifican de acuerdo con la distancia a la que se encuentran de dicha celda de la siguiente forma:

- **Usuarios de cerca y media celda:** Usuarios que se encuentran en una zona cercana a la celda y donde generalmente el nivel de potencia RSRP es mayor a los -83.5 dBm. Para dichos usuarios el esquema de modulación es superior en comparación a los usuarios de borde de celda.
- **Usuarios de borde de celda:** Usuarios que se encuentran en la zona del borde de la celda y donde generalmente el nivel de potencia RSRP es menor a los -83.5 dB. Para dichos usuarios el esquema de modulación es inferior en comparación a los usuarios más cercanos a la celda.

Es importante hacer claridad que no hay ningún umbral puntual para discriminar entre usuarios de media celda y borde de celda. La elección de -83.5 dBm se hizo en base a valores habituales de RSRP y SNR en redes LTE que se pueden encontrar en [34], y se empleará para el desarrollo de la presente propuesta únicamente.

8.2 Definición de Vecindad

Se le denomina vecindad a la celda que es adyacente a otra celda. Para la propuesta a desarrollar se definirán 2 tipos de vecindades:

- **Vecindades para Movilidad:** Estas vecindades son definidas para los handover propiciados por la movilidad de los usuarios, es decir en este caso todas las celdas adyacentes entre sí son consideradas vecindades para movilidad. Por ejemplo, en la figura 34 las vecindades de la celda 1 serán la celda 2 y la celda 3 debido a que un usuario ubicado en la celda 1 se puede mover hacia cualquiera de estas celdas, así mismo las vecindades de la celda 2 serán la celda 1, la celda 3 y la celda 7 y de forma sucesiva para las demás.
- **Vecindades para Balanceo de Carga:** Estas vecindades son definidas para los handover propiciados por el balanceo de carga entre celdas. En este caso no todas las celdas adyacentes son aptas para admitir recibir de borde celda ya que sus coberturas no se solapan con las demás y por ende aplicar un handover sobre dicha celda sería ineficaz. Por ejemplo, en la figura 34 la celda 1 no tiene vecindades para balanceo, la celda 2 tiene como vecindad a las 7, la celda 7 tiene como vecindad la celda 2 y 3 y así sucesivamente.

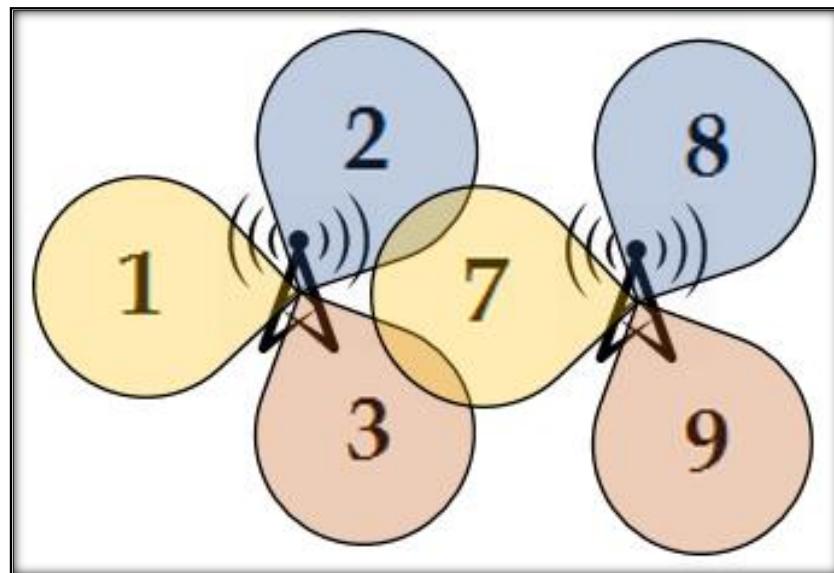


Figura 34. Imagen de ejemplo para evidenciar el concepto de vecindad

8.3 Balanceo de carga de primer orden

En primer lugar, se definirá la carga dentro de una celda como la demanda de Resources Block sobre la cantidad total de Resources Block disponibles dependiendo del ancho de banda que se haya asignado.

$$\text{Current Load} = \frac{\text{User Demand [Resources Block]}}{\text{Total number of Resources Block [Resources Block]}} * 100\% \quad (4)$$

Cuando la carga (Current Load) excede el 100% la celda presenta sobrecarga.

Una celda presenta saturación de recursos de radio cuando la carga alcanza el valor máximo de los que dispone. Es claro que al llegar a este umbral la celda tiene que interrumpir algunas conexiones con los usuarios que se encuentran conectados al eNodeB implicando afectación de servicio, bajo throughput por usuario, pérdida de paquetes, retardos elevados (delay), entre otros lo cual no es deseable en la red.

A partir de esta definición, el balanceo de carga móvil brinda la posibilidad de que una celda con una carga alta cercana al 100% (medida en uso de RB) pueda liberar usuarios de borde de celda hacia una celda vecina con una carga menor. Este método se implementa reduciendo los valores de umbral (**threshold**) propios de cada celda con el fin de promover el handover de dichos usuarios hacia la nueva celda.

De forma gráfica este proceso se evidencia en las figuras 35 y 36. En la figura 35 se observa un clúster de 4 eNodeB, donde cada eNodeB posee 3 celdas para un total global de 12 celdas. Supóngase que la celda número 2 presenta sobrecarga y que la celda 7 la cual es vecina tiene Resource Block disponibles, entonces los usuarios de borde de la celda 2 pueden ser transferidos a la celda 7 como se ilustra en la figura 36.

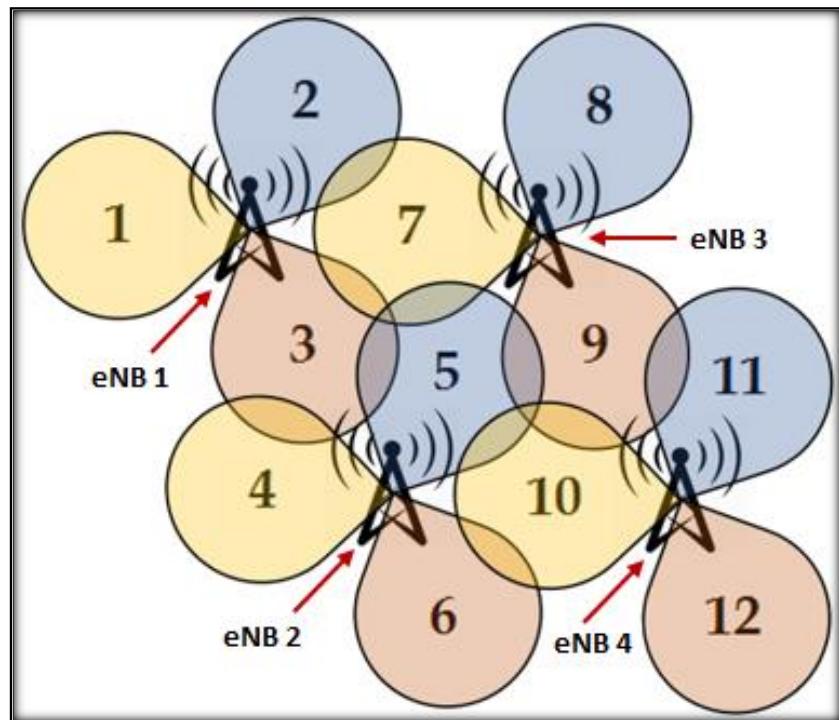


Figura 35. Organización de un clúster ejemplo de 12 celdas

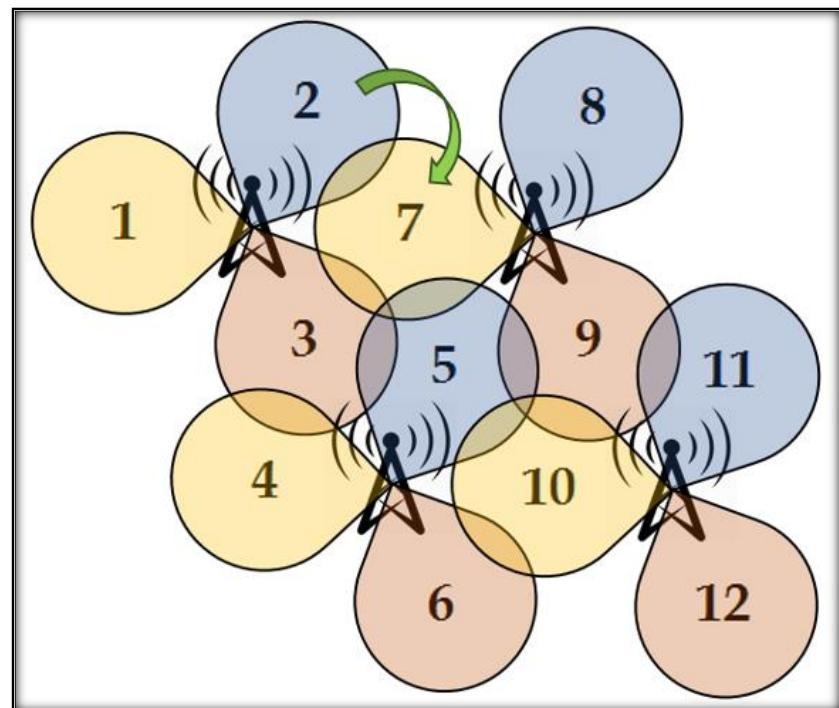


Figura 36. Ejemplo de balanceo de usuarios de borde entre la celda 2 con sobrecarga y la celda 7 con Resources Block disponibles.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

El proceso para elegir a que celda transferir los usuarios de borde celda se realiza enviando un paquete Solicitud (Request) desde la celda origen a las posibles celdas destino, en donde se obtiene la información del porcentaje de recursos de radio disponibles por cada vecindad, dado mediante la siguiente ecuación:

$$\text{Available resources} = \text{Total Number of RB} - \text{Number of RB in used} \quad (5)$$

Cada celda envía un paquete Respuesta (Response) a la celda origen, y la celda con el valor más alto de Recursos Disponibles será la celda a la que se transferirán los usuarios de borde de celda.

Cabe aclarar que en la topología de la figura 35 cada eNodeB posee 3 celdas, sin embargo, no es obligatorio que sea de esta manera, un eNodeB podría tener más o menos celdas de acuerdo con los criterios de diseño y dimensionamiento de la red.

8.4 Balanceo de carga de segundo orden

Basados en la figura 36, supóngase la situación donde la celda 2 presenta sobrecarga y la celda 7 (única vecindad de balanceo de carga para la celda 2) no tiene Resources Block disponibles, sin embargo, la celda 5 (vecindad de la celda 7) cuenta con Resources Block disponibles. ¿Cómo se podrían utilizar estos recursos con el fin de reducir la carga de la celda 2?

La propuesta para lograr ello es hacer un balanceo encadenado entre la celda 2, la celda 7 y finalmente la celda 5 como se observa en la figura 37.

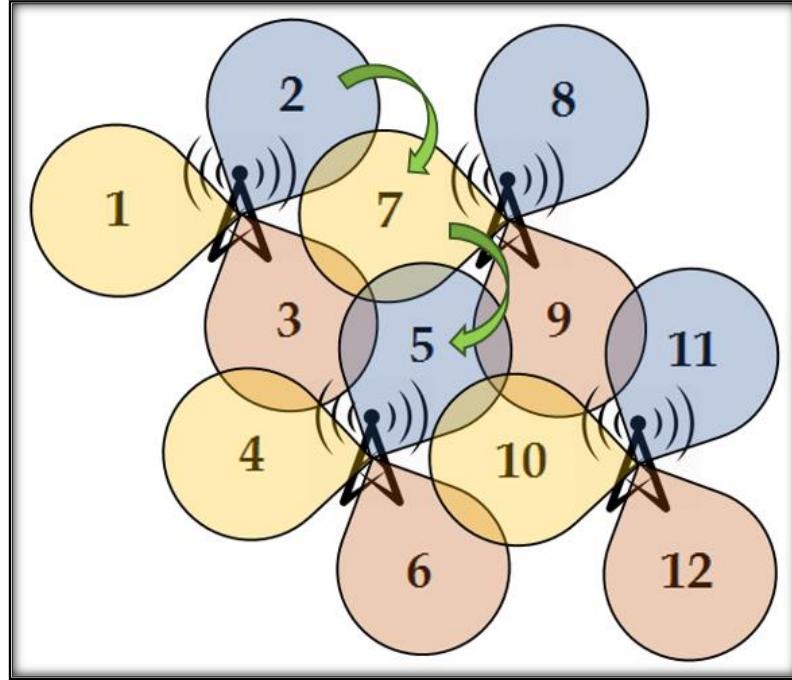


Figura 37. Balanceo de carga encadenado entre las celdas 2, 7 y 5. Un porcentaje de usuarios de borde de la celda 7 se transfieren a la celda 5, posteriormente la celda 2 transfiere usuarios de borde a la celda 7 reduciendo su carga actual.

Si la celda 2 tuviese una carga del 120% (sobrecarga del 20%), la celda 7 una carga del 100% y la celda 5 una carga de 50%, la celda 7 podría transferir un cierto porcentaje de usuarios a la celda 5 liberando Resources Block. Dependiendo de la cantidad de recursos liberados se podrá transferir otro porcentaje de usuarios desde la 2 a la celda 7 reduciendo la carga de la celda con sobrecarga y aumentando el throughput global de la red.

Para el presente trabajo se propone implementar un servidor central que se encargue de administrar las funciones relacionadas con el balanceo de carga. Dicho servidor recibirá de cada eNodeB mensajes de actualización sobre los estados de cada celda incluyendo la carga respectiva por medio de las interfaces X2. Bajo esta premisa es factible que el servidor establezca las rutas más convenientes para proceder a realizar el balanceo de carga de segundo orden.

El proceso bajo el cual el servidor puede tener conocimiento pleno sobre la topología y los Resources Block disponibles de cada celda de forma periódica es el siguiente:

Cada eNodeB envía un mensaje al servidor central donde informa el valor de Recursos de Radio disponibles de cada una de sus celdas (Radio Resources Available). A modo ejemplo

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

el eNodeB1 enviaría al servidor central la siguiente tabla correspondiente a las 3 celdas que tiene asignadas, en este caso la celda 1, 2 y 3.

Physical Cell ID	Resources Block Disponibles
PCI celda 1	Valor 1
PCI celda 2	Valor 2
PCI celda 3	Valor 3

Tabla 7. Almacenamiento de la información de las celdas correspondientes al eNodeB1

Adicional a esta información cada eNodeB debe enviar al servidor central las vecindades de balanceo de carga por celda tal y como se refleja en la tabla 8.1, con el fin de que el servidor establezca la topología de la red y defina los saltos que existe entre una celda y otra.

PCI de la Celda Origen	PCI de las celdas que son vecindad de balanceo
PCI celda 1	PCI j, PCI i, ...
PCI celda 2	PCI m, PCI l, ...
PCI celda n	PCI q, PCI t, ...

Tabla 8. Tabla generada por cada eNodeB indicando las vecindades de balanceo por cada celda

A modo de ejemplo, la información que enviaría el eNodeB 1 al servidor central sería la siguiente:

PCI de la Celda Origen	PCI de las celdas que son vecindad de balanceo
PCI celda 1	No tiene vecindades de balanceo
PCI celda 2	PCI de la celda 7
PCI celda 3	PCI de la celda 4, PCI de la celda 5, PCI de la celda 7

Tabla 8.2. Tabla ejemplo generada por eNodeB 1 indicando las vecindades de balanceo por cada celda

Una ventaja importante es que esta información se transmite por la interfaz X2 y no por la interfaz S1 (Datos de Usuario y Señalización), por lo cual no se congestionaría el canal de transmisión (plano de usuario y control) entre los eNodeB y el EPC.

En el momento en el que el servidor central recolecta los mensajes de actualización de cada eNodeB procede a construir la matriz de adyacencia de la red (donde se indican las vecindades por celda) y la tabla de posibles rutas de balanceo.

Con la información de las vecindades de todas las celdas de la red el servidor puede construir un grafo de relaciones entre las celdas como se observa en la figura 38, donde las

relaciones azules corresponden a vecindades de movilidad y las relaciones rojas corresponden a vecindades de movilidad y a su vez de balanceo.

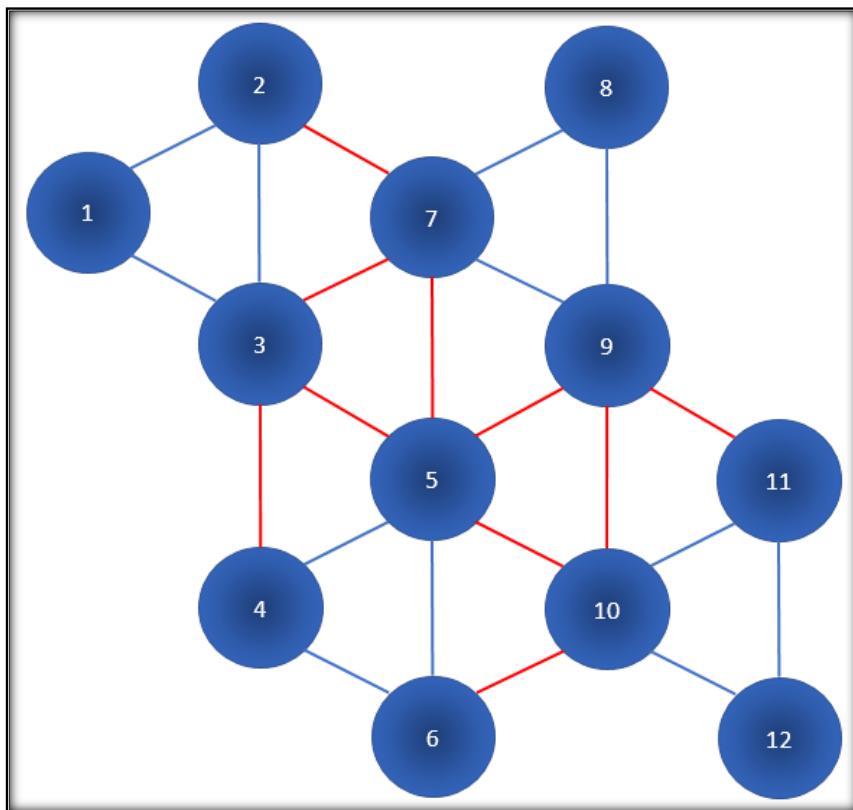


Figura 38. Grafo donde se evidencian las vecindades de movilidad y las vecindades de balanceo de las celdas correspondiente a la red de estudio.

Este grafo puede ser plasmado en forma matricial de tamaño $n \times n$ donde n es el número de celdas en la red que en este caso es 12 [35]. Las filas indican las celdas y las columnas las posibles vecindades, donde un 1 indica que si existe vecindad de balanceo y un 0 indica que no hay vecindad de balanceo. La matriz M para la topología de la figura 35 queda expresada de la siguiente forma:

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Tomando como base la información de la matriz M y la información de la tabla 7, el servidor central puede calcular las posibles rutas de balanceo desde cada una de las celdas de la red y asignar un valor a cada ruta cuya medida será la suma de Resources Block disponibles en las celdas contenidas en cada ruta, en cuyo caso si la ruta es de 1 solo salto (primer orden) el valor de la ruta será la cantidad de Resources Blocks en la celda de posible destino y si la ruta es de 2 saltos (segundo orden) el valor de la ruta será la suma de los Resources Blocks de la celda intermedia y de los Resources Blocks de la celda de posible destino. Es claro concluir a partir de ello, que entre mayor cantidad de recursos disponibles mayor será el valor de la ruta y por ende será la ruta más opcionada para ejecutar el balanceo.

Con el fin de aclarar como se calcula el valor de la ruta, se toma como ejemplo de nuevo la ruta expuesta en la figura 37 donde la celda 2 tiene una carga del 20%, la celda 5 tiene una carga del 100% y la celda 7 tiene una carga del 50%. Desde la celda 2 (celda origen) el valor de la ruta sería la suma de los recursos disponibles en la celda 5 más los recursos disponibles en la celda 7, por lo cual el valor de la ruta sería igual a:

$$\textbf{Valor de la ruta} = 0\% \text{ (recursos disponibles celda 5)} + 50\% \text{ (recursos disponibles celda 7)} = 50\%$$

Como el valor de la ruta es 50% y la sobrecarga de la celda 2 es del 20%, esta ruta se almacena como una ruta potencial para ejecutar el balanceo de carga.

Después de calcular las rutas de balanceo desde cada celda, el servidor central consolida una tabla resumen de las posibles rutas de balanceo por celda como se expone en la tabla 9.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Celda Origen	Métrica de Saltos	Celda de posible Destino	Celda intermedia en la ruta	Valor de la ruta
PCI celda 1	0	PCI celda i	PCI celda j	Valor ruta 1
PCI celda 2	0	PCI celda p	PCI celda q	Valor ruta 2
PCI celda n	0	PCI celda m	PCI celda r	Valor ruta n
...

Tabla 9. Tabla base de rutas almacenada en el servidor

La tabla base de rutas incluye los siguientes 5 campos:

- Celda Origen: En este campo se incluye el PCI (Physical Cell ID) de la celda origen. La celda origen es la celda sobre la que se inicia el proceso de balanceo de carga en caso de llegar a presentar saturación de recursos.
- Métrica de saltos: Indica la cantidad de saltos que existe entre la celda origen y la celda de posible destino. 0 indica que es una ruta de 1 solo salto (primer orden), 1 indica que la ruta es de 2 saltos (segundo orden).
- Celda de posible Destino: En este campo se incluye el PCI (Physical Cell ID) de la celda de posible destino. La celda de posible destino es la celda sobre la que se finaliza el proceso de balanceo de carga en caso de llegar a presentar saturación de recursos la celda origen.
- Celda intermedia en la ruta: Este campo se refiere a la celda intermedia entre la celda origen y la celda de posible destino en la ruta calculada por el servidor central.
- Valor de la ruta: Este campo se refiere a la cantidad de recursos disponibles por ruta. Como se mencionó anteriormente si la ruta es de 1 solo salto (primer orden) el valor de la ruta será la cantidad de Resources Blocks en la celda de posible destino y si la ruta es de 2 saltos (segundo orden) el valor de la ruta será la suma de los Resources Blocks de la celda intermedia y de los Resources Blocks de la celda de posible destino.

A modo de ejemplo, se tomará como base la celda 3 y se describirá como el servidor central calcula las rutas y las consolida en la tabla base de rutas. El primer paso es calcular las rutas de primer orden, es decir las rutas de un solo salto. De acuerdo con el grafo de la red (figura 38) la celda 3 tiene 3 vecindades de balanceo, las cuales son agregadas a la tabla base.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Celda Origen	Métrica de Saltos	Celda de posible Destino	Celda intermedia en la ruta	Recursos de la ruta
3	0	4	-	Costo ruta 1
3	0	5	-	Costo ruta 2
3	0	7	-	Costo ruta 3

El siguiente paso es hallar las rutas de segundo orden, es decir las rutas de 2 saltos. De nuevo bajo la referencia la figura 38 se agregan las rutas a tabla base.

Celda Origen	Métrica de Saltos	Celda de posible Destino	Celda intermedia en la ruta	Costo de la ruta
3	0	4	-	Costo ruta 1
3	0	5	-	Costo ruta 2
3	0	7	-	Costo ruta 3
3	1	9	5	Costo ruta 4
3	1	10	5	Costo ruta 5
3	1	2	7	Costo ruta 6

Tabla 10. Tabla de rutas de balanceo para la celda tomando como referencia la topología de la figura 35 y 38.

De esta manera, celda por celda el servidor consolida la tabla base de rutas de toda la red.

Ahora bien, cuando el servidor central detecta que una de las celdas presenta sobrecarga, revisa inmediatamente la tabla base de rutas e inicia el proceso de balanceo de carga a través de la ruta de mayor costo. En primera instancia el eNodeB tratará de realizar el balanceo de carga con las celdas adyacentes (cantidad de salto igual a 1) y posteriormente con las celdas de segundo orden (cantidad de saltos igual a 2), esto debido a que las rutas con 1 salto requieren una cantidad menor de handovers que las rutas con 2 saltos.

Después de realizar el balanceo de carga, de nuevo se realiza la difusión de actualizaciones al servidor central por parte de los eNodeB y el proceso se normaliza hasta experimentar nuevamente sobrecarga en alguna de las celdas correspondientes al clúster de celdas estudiado.

A modo aclaratorio, es evidente que dentro de la red de la figura 35 hay celdas que no cuentan con vecindades de balanceo, motivo por el cual en caso de presentarse sobrecarga en estas celdas, no es posible balancear carga con otras celdas y por lo tanto es inevitable que se descarten paquetes de voz y datos de acuerdo con las políticas de servicio establecidas en la red.

8.5 Modelo para la generación de la tabla de rutas de balanceo

Con el fin que el servidor central pueda definir la tabla de rutas de balanceo que le permita identificar la mejor opción para el balanceo de carga entre las celdas de la red se propone el siguiente flujo:

Inicio Tabla_Enrutamiento

Iniciar la tabla Rutas en cero

Definir la Matriz de Adyacencia M (Vecindades de balanceo por celda)

Definir el vector de Carga (Vector de 12 posiciones donde se almacena la cantidad de recursos disponibles de cada celda)

i = 1

// Se añaden las rutas de primer orden, es decir las vecindades directas por celda
mientras (i <= cantidad de celdas en la red)

j = 1

mientras (j <= cantidad de celdas en la red)

si (posición M(i,j) = 1) // La celda ubicada en la columna j es vecindad

Añade la ruta en la tabla Rutas con celda origen i

Fin

Fin

Fin

// Se añaden las rutas de segundo orden, es decir las rutas que se encuentran a 2 saltos de distancia desde la celda origen

mientras (i <= cantidad de celdas en la red)

j = 1

mientras (j <= cantidad de celdas en la red)

si (posición M(i,j) = 1) // La celda ubicada en la columna j es vecindad

Añade las rutas de la celda j en la tabla Rutas con celda origen i

Fin

Fin

Fin

Fin Tabla_Enrutamiento

El código empleado para ejecutar la función Tabla de enrutamiento se describe en el apéndice D.

8.6 Modelamiento Matemático

Para obtener un modelo matemático base establecerán las siguientes restricciones:

1. A cada celda llegarán una cierta cantidad de paquetes (voz y datos) y éstas serán las encargadas de procesarlos. Inicialmente (sin la implementación del modelo de balanceo de carga) el estado de cada celda es independiente al de las demás.
2. Los usuarios se distribuirán sobre el área del sistema de manera uniforme en el estado inicial ($t = 0$).
3. No se considerarán eventos simultáneos en el sistema, es decir que no ocurrirán llegadas de paquetes ni procesamientos de paquetes al mismo tiempo.
4. Los tiempos de llegada de paquetes al sistema se distribuirán de forma exponencial, constituyendo un proceso de Poisson.
5. Los tiempos de servicio de paquetes en el sistema se distribuirán de forma exponencial, constituyendo un proceso de Poisson.
6. Cada celda puede procesar paquetes de voz (tráfico de alta prioridad) y datos (tráfico de baja prioridad) bajo la siguiente política de servicio:
 - 6.1 Si la celda se encuentra libre y llega algún paquete éste será procesado.
 - 6.2 Si la celda está procesando un paquete de voz y llega un paquete de voz éste último será descartado del sistema, si llega un paquete de datos éste será enviado a la cola.
 - 6.3 Si la celda está procesando un paquete de datos y llega un paquete de voz, el paquete de datos que estaba siendo procesado será enviado a la cola y se procesará el paquete de voz. Si llega un paquete de datos éste será enviado a la cola.

El primer paso es encontrar que distribución de llegada de paquetes va a tener cada celda teniendo en cuenta que la llegada de paquetes al sistema constituye un proceso de Poisson (consideraciones previas 3 y 4) como se puede observar en la figura 39.

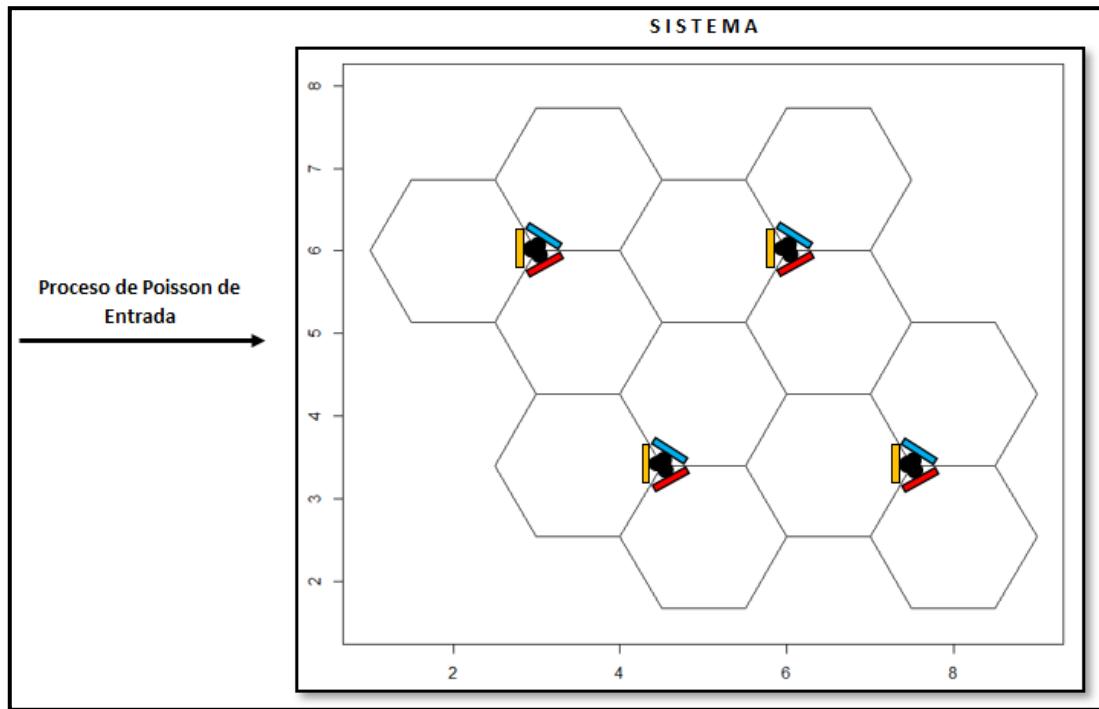


Figura 39. Proceso de llegada de paquetes a la red

8.6.1 Definición del Proceso de Poisson (Llegadas de paquetes al sistema)

Sea t el intervalo en el que se evalúa cuántos paquetes llegan a una celda, es claro notar que a medida que el intervalo t se hace más grande la probabilidad de que lleguen más paquetes también lo es, por lo cual se define que la longitud del intervalo de t es proporcional a la probabilidad de que k paquetes lleguen al sistema. [36]

Considérense n subintervalos de tiempo sobre todo el intervalo t previamente definido como se observa en la figura 40.

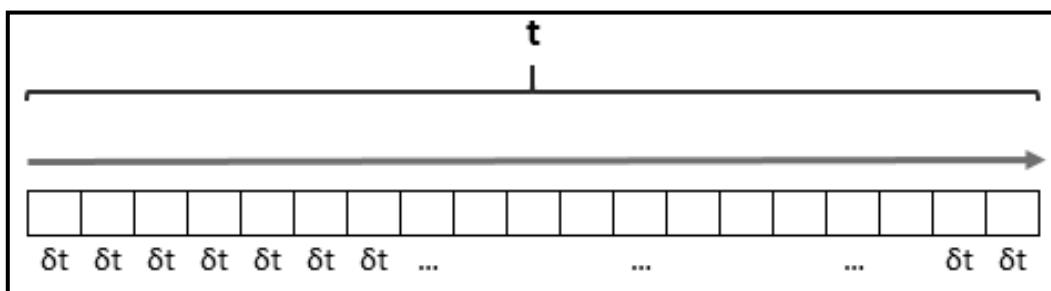


Figura 40. Intervalo t sobre el que se realiza la observación

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Tal que la suma de todos los n subintervalos de tiempo δt es igual a t.

$$t = n * \delta t \quad (6)$$

Debido a la consideración de que no hay llegadas de paquetes simultáneas, la probabilidad de que en un subintervalo δt lleguen k paquetes se reduce a 0 paquetes a lo mucho a un solo paquete, lo cual se expresa de acuerdo con la ecuación 7.

$$P[k \text{ llegadas en } \delta t] = \begin{cases} \lambda * \delta t & \text{para } k = 1 \\ 1 - \lambda * \delta t & \text{para } k = 0 \\ o(t) & \text{para } k > 1 \end{cases} \quad (7)$$

- La probabilidad de que haya una llegada en un subintervalo de tiempo δt es proporcional a la longitud de éste, cuya constante de proporcionalidad es λ .
- La probabilidad de que no hayan llegadas en dicho intervalo es la probabilidad del evento contrario ($1 - P[1 \text{ llegada}]$).
- La probabilidad de que haya más de una llegada tiende a cero y se expresa como una función $o(t)$ que tiende a cero de forma más rápida que δt .

Después de tener definida la probabilidad de k llegadas en un subintervalo δt infinitesimal, se puede hallar la probabilidad de que en el intervalo t haya m llegadas de paquetes haciendo uso de la probabilidad de una variable binomial empleada de la siguiente forma:

$$P[m \text{ llegadas en } t] = \binom{n}{m} (\lambda * \delta t)^m (1 - \lambda * \delta t)^{n-m} \quad (8.1)$$

Realizando la simplificación de la ecuación 8.1 se obtiene el resultado de la ecuación 8.2.

$$\begin{aligned} P[m \text{ llegadas en } t] &= \frac{n!}{m! * (n - m)!} * \left(\lambda * \frac{t}{n}\right)^m * \left(1 - \lambda * \frac{t}{n}\right)^{n-m} \\ P[m \text{ llegadas en } t] &= \frac{n * (n - 1) * (n - m) * (\lambda^m * t^m)}{n^m * m!} * \left(1 - \lambda * \frac{t}{n}\right)^{n-m} \\ P[m \text{ llegadas en } t] &= \frac{n * (n - 1) * (n - m) * (\lambda^m * t^m)}{n^m * m!} * \frac{\left(1 - \lambda * \frac{t}{n}\right)^n}{\left(1 - \lambda * \frac{t}{n}\right)^m} \end{aligned}$$

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Como se menciono que los intervalos de tiempo δt tienden a cero, esto implica que n tiende a infinito por la relación de la ecuación 6. De esta manera la probabilidad de la ecuación 8.1 se expresa de la siguiente forma:

$$P[m \text{ llegadas en } t] = \lim_{n \rightarrow \infty} \left[\frac{n * (n - 1) * (n - m) * (\lambda^m * t^m)}{n^m * m!} * \frac{\left(1 - \lambda * \frac{t}{n}\right)^n}{\left(1 - \lambda * \frac{t}{n}\right)^m} \right]$$

$$P[m \text{ llegadas en } t] = \lim_{n \rightarrow \infty} \left[\frac{n * (n - 1) * (n - m) * (\lambda^m * t^m)}{n^m * m!} * \frac{\left(1 - \lambda * \frac{t}{n}\right)^n}{\left(1 - \lambda * \frac{t}{n}\right)^m} \right]$$

$$P[m \text{ llegadas en } t] = \frac{(\lambda t)^m}{m!} * \lim_{n \rightarrow \infty} \left(\frac{n * (n - 1) * (n - m)}{n^m} \right) * \lim_{n \rightarrow \infty} \frac{\left(1 - \lambda * \frac{t}{n}\right)^n}{\left(1 - \lambda * \frac{t}{n}\right)^m}$$

$$P[m \text{ llegadas en } t] = \left(\frac{(\lambda t)^m}{m!} \right) * 1 * e^{-\lambda t} = \frac{(\lambda t)^m}{m!} * e^{-\lambda t} \quad (8.2)$$

Este resultado expresa la distribución de la llegada de paquetes al sistema de acuerdo con un proceso de Poisson. A modo de ilustración se mostrará el comportamiento de dicha distribución para diferentes valores de λ (figura 41).

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

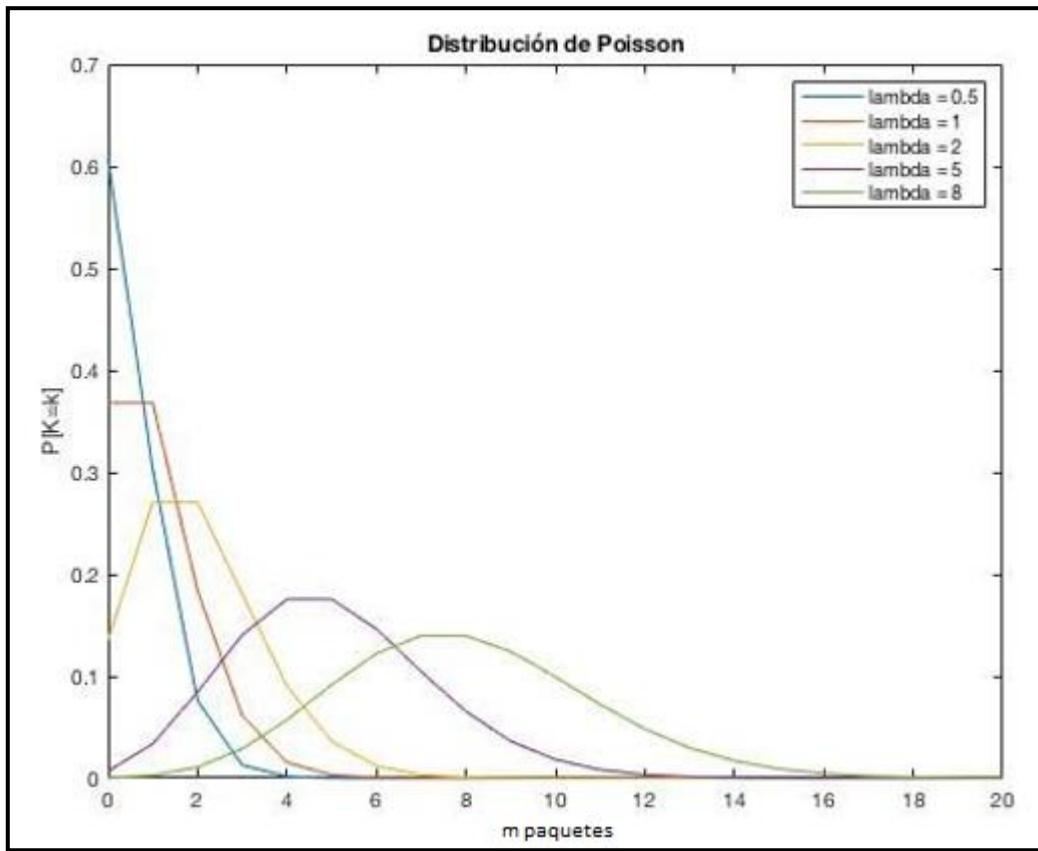


Figura 41. Distribución de Poisson para distintos valores de lambda con t=1

Vale la pena aclarar que esta distribución es discreta debido a que la cantidad de paquetes en el sistema solo puede tomar valores enteros.

Adicional al resultado obtenido en la ecuación 8.2 se puede obtener la distribución del tiempo que hay entre una llegada y otra dentro de un proceso de Poisson de la siguiente forma:

Si durante el intervalo de tiempo t no hay ninguna llegada de paquetes, la probabilidad de que esto suceda se puede obtener haciendo uso de la ecuación 8.2 y tomando $m = 0$.

$$P[0 \text{ llegadas en } t] = e^{-\lambda t} \quad (9.1)$$

Lo cual se puede expresar en término de la variable continua T (tiempo) como:

$$P[T > t] = e^{-\lambda t} \quad (9.2)$$

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

indicando que el tiempo entre llegada de paquetes es mayor al intervalo t que ha sido tomado como base del análisis. El evento contrario al de la ecuación 9.2 se puede expresar como:

$$P[T \leq t] = 1 - e^{-\lambda t} \quad (9.3)$$

que corresponde a una Función Acumulativa de Probabilidad. Para hallar la Función de Distribución de Probabilidad solo basta con derivar la ecuación 9.3 con respecto al tiempo:

$$pdf = \frac{d}{dt}(1 - e^{-\lambda t}) = \lambda * e^{-\lambda t} \quad (9.4)$$

La Función de Distribución de Probabilidad para el tiempo entre llegadas de paquetes obtenida en la ecuación 9.4 será de utilidad para deducir el resultado en el subcapítulo 8.6.2.

8.6.2 Proceso de Llegada de paquetes por Usuario

Los paquetes de voz y datos llegan a la red de acuerdo con un proceso de Poisson cuya tasa global de llegadas es λ . Así la probabilidad de que llegue un paquete al usuario 1 será λ_1/λ , para el usuario 2 será λ_2/λ y así sucesivamente como se muestra en la figura 41.

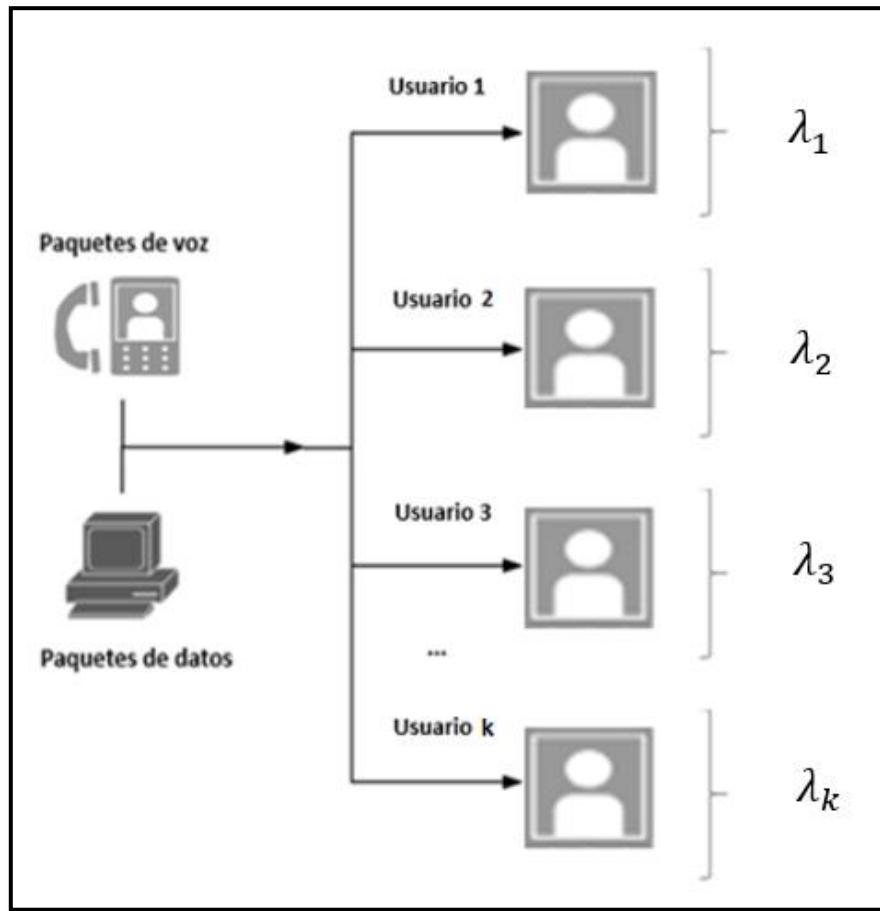


Figura 41. Asignación de los paquetes entrantes al sistema

En general para el k -ésimo usuario la probabilidad de que llegue un paquete será:

$$P [\text{llegue un paquete al } k \text{ ésmo usuario}] = \frac{\lambda_k}{\lambda} \quad (10.1)$$

Donde:

$$\lambda = \sum_{i=1}^k \lambda_i \quad (10.2)$$

Por ende la variable aleatoria $N(t)$, que corresponde al número de paquetes en el sistema en un tiempo determinado, debe ser igual a la suma de los paquetes recibidos por todos los usuarios, es decir:

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

$$N(t) = \sum_{i=1}^k N_i(t) \quad (10.3)$$

O dicho de otra forma:

$$n = \sum_{i=1}^k n_i \quad (10.4)$$

Donde n corresponde al número de paquetes.

Ahora lo que se desea hallar es la probabilidad conjunta de las variables aleatorias N_i teniendo en cuenta la condición de la ecuación 10.3. Para dar solución a ello se hará uso de la probabilidad condicional como se expone a continuación:

$$P[N_1 = n_1, N_2 = n_2, \dots, N_k = n_k] = P[N_1 = n_1, N_2 = n_2, \dots, N_k = n_k | N(t) = n] * P[N(t) = n] \quad (11.1)$$

Donde el primer término a la derecha de la igualdad corresponde a la distribución multinomial [37] debido a la condición expresada en la ecuación 10.3 y 10.4:

$$\begin{aligned} P[N_1 = n_1, N_2 = n_2, \dots, N_k = n_k | N(t) = n] &= \binom{n}{n_1, n_2, \dots, n_k} * P[N_1 = n_1]^{n_1} * P[N_2 = n_2]^{n_2} * \dots * P[N_k = n_k]^{n_k} \\ &= \frac{n!}{n_1! * n_2! * \dots * n_k!} * P[N_1 = n_1]^{n_1} * P[N_2 = n_2]^{n_2} * \dots * P[N_k = n_k]^{n_k} \end{aligned} \quad (11.2)$$

y las probabilidades pueden ser expresadas de acuerdo con la ecuación 10.1, con lo cual este término se expresa de la siguiente forma:

$$P[N_1 = n_1, N_2 = n_2, \dots, N_k = n_k | N(t) = n] = \frac{n!}{n_1! * n_2! * \dots * n_k!} * \left(\frac{\lambda_1}{\lambda}\right)^{n_1} * \left(\frac{\lambda_2}{\lambda}\right)^{n_2} * \dots * \left(\frac{\lambda_k}{\lambda}\right)^{n_k} \quad (11.3)$$

El segundo término la probabilidad de que hayan llegado n paquetes al sistema de acuerdo con un proceso de Poisson tal y como se verificó en la ecuación 8.2:

$$P[N(t) = n] = \frac{(\lambda t)^n}{n!} * e^{-\lambda t} \quad (11.4)$$

Por lo tanto la probabilidad conjunta queda expresada como:

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

$$P[N_1 = n_1, \dots, N_k = n_k] = \frac{n!}{n_1! * n_2! * \dots * n_k!} \left(\frac{\lambda_1}{\lambda}\right)^{n_1} \left(\frac{\lambda_2}{\lambda}\right)^{n_2} \dots \left(\frac{\lambda_k}{\lambda}\right)^{n_k} * \frac{(\lambda t)^n}{n!} * e^{-\lambda t}$$

De acuerdo con la ecuación 10.2, la función exponencial se puede expresar en términos de las tasas de llegada para cada usuario:

$$e^{-\lambda t} = e^{-(\lambda_1 + \lambda_2 + \dots + \lambda_k)t} = e^{-\lambda_1 t} * e^{-\lambda_2 t} * \dots * e^{-\lambda_k t}$$

y la ecuación 11.3 finalmente puede ser expresada como:

$$\begin{aligned} P[N_1 = n_1, \dots, N_k = n_k] &= \frac{n!}{n_1! * n_2! * \dots * n_k!} \left(\frac{\lambda_1}{\lambda}\right)^{n_1} \left(\frac{\lambda_2}{\lambda}\right)^{n_2} \dots \left(\frac{\lambda_k}{\lambda}\right)^{n_k} \frac{(\lambda t)^n}{n!} e^{-\lambda_1 t} * \dots * e^{-\lambda_k t} \\ P[N_1 = n_1, \dots, N_k = n_k] &= \frac{1}{n_1! * n_2! * \dots * n_k!} (\lambda_1)^{n_1} (\lambda_2)^{n_2} \dots (\lambda_k)^{n_k} \frac{\lambda^{(n_1+n_2+\dots+n_k)} t^{(n_1+n_2+\dots+n_k)}}{\lambda^{(n_1+n_2+\dots+n_k)}} e^{-\lambda_1 t} * \dots * e^{-\lambda_k t} \\ P[N_1 = n_1, \dots, N_k = n_k] &= \frac{(\lambda_1 t)^{n_1}}{n_1!} e^{-\lambda_1 t} * \frac{(\lambda_2 t)^{n_2}}{n_2!} e^{-\lambda_2 t} * \dots * \frac{(\lambda_k t)^{n_k}}{n_k!} e^{-\lambda_k t} \quad (12) \end{aligned}$$

Basados en el resultado de la ecuación 12, la función conjunta de la ecuación 11.1 se expresa como el producto de flujos de Poisson con tasas de llegada λ_i con $i = 1, 2, \dots, k$.

La conclusión de este análisis permite validar que el resultado de la descomposición de un flujo de Poisson con tasa de llegada λ son flujos de Poisson independientes con tasas de llegada λ_i con $i = 1, 2, \dots, k$ cuya suma es igual a λ . Esto quiere decir que los flujos de llegada de paquetes para cada usuario se pueden modelar como flujos de Poisson independientes.

[38] [39]

8.6.3 Definición de la Cadena de Markov para una Celda de un solo canal

El siguiente paso es modelar inicialmente el comportamiento de los paquetes en un sistema de una celda y de un solo canal que recibe y procesa paquetes de acuerdo con la política de servicio mencionada en las condiciones previas (4, 5 y 6). Para ello se realizará la representación mediante una cadena de Markov como se ilustra en la figura 42. [40] [41]

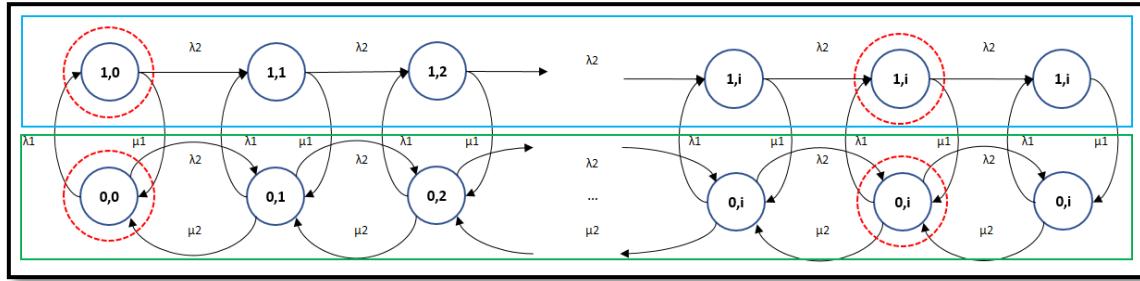


Figura 42. Representación de los paquetes en una celda mediante una cadena de Markov

Los estados de la cadena de Markov se definen de la siguiente forma:

- El estado (0,0) indica que el canal de la celda se encuentra vacío.
- El estado (1,0) indica que la celda está procesando un paquete de voz.
- El estado (0,1) indica que la celda está procesando un paquete de datos.
- El estado (1,i) indica que la celda está procesando un paquete de voz y hay i paquetes de datos en la cola.
- El estado (0,i) indica que la celda está procesando un paquete de datos y hay i-1 paquetes de datos en la cola.

Las transiciones de la cadena de Markov se definen de la siguiente forma:

- Partiendo de un estado en donde la celda no esté procesando un paquete de voz puede haber una llegada de paquete de voz con tasa λ_1 , una llegada de paquete de datos con tasa λ_2 o una salida de un paquete de datos con tasa μ_2 excepto para el estado (0,0).
- Partiendo de un estado en donde la celda esté procesando un paquete de voz puede haber una llegada de paquete de datos con tasa λ_2 o una salida de un paquete de voz con tasa μ_1 . Si llega un paquete de voz será descartado debido a que no se encolarán paquetes de voz, como se mencionó en las políticas de servicio.

8.6.4 Función Generadora de Probabilidad

El siguiente paso es encontrar la Función Generadora de Probabilidad del sistema descrito en la figura 42, con el fin de hallar de una forma sencilla el valor esperado de la cantidad de paquetes en el sistema en un determinado tiempo. [42] [41]

Sea I la variable aleatoria que representa la cantidad de paquetes en el sistema, su Función Generadora de Probabilidad se define de acuerdo con la ecuación 13.1:

$$G(z) = \sum_{k=0}^{\infty} P_k * z^k \quad (13.1)$$

Donde P_k corresponde a la probabilidad del k -ésimo evento, en cuyo caso corresponderá a uno de los estados de la Cadena de Markov estudiada en el subcapítulo anterior. Tomando como base la ecuación 13.1, hallar el valor esperado de la variable aleatoria se obtiene fácilmente derivando con respecto a z y evaluando en $z=1$ de la siguiente forma:

$$\frac{d}{dz} G(z)|_{z=1} = \frac{d}{dz} \left(\sum_{k=0}^{\infty} P_k * z^k \right) |_{z=1} = \sum_{k=0}^{\infty} P_k * \frac{d}{dz} (z^k) |_{z=1} = \sum_{k=0}^{\infty} P_k * k(z^{k-1}) |_{z=1} = \sum_{k=0}^{\infty} P_k * k$$

Por lo cual el valor esperado de la variable aleatoria se expresa como:

$$E[I] = \frac{d}{dz} G(z) |_{z=1} = \sum_{k=0}^{\infty} P_k * k \quad (13.2)$$

Este resultado se empleará más adelante para obtener calcular de forma teórica la cantidad de paquetes en el sistema.

Ahora bien, retomando el análisis del sistema estudiado se expresará la Función Generadora Total del sistema de la siguiente manera:

$$G(z) = G_0(z) + G_1(z) \quad (13.3)$$

Donde $G_0(z)$ es la función parcial que se recorre sobre los estados de la forma $(0,i)$ para $i \geq 1$ y $G_1(z)$ es la función parcial que se recorre sobre los estados de la forma $(1,i)$ para $i \geq 1$. A modo de aclaración los estados de la forma $(0,i)$ se observan en color azul y los estados de la forma $(1,i)$ se observan en color verde sobre la figura 42.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Estas dos funciones se expresan de acuerdo con las ecuaciones 14 y 15.

$$G_0(z) = \sum_{i=0}^{\infty} P_{0,i} * z^i \quad (14)$$

$$G_1(z) = \sum_{i=0}^{\infty} P_{1,i} * z^i \quad (15)$$

Partiendo de la Cadena de Markov descrita se obtiene un sistema de ecuaciones que permite poder relacionar la probabilidad de un estado $(0,i)$ o $(1,i)$ en términos del estado inicial $(0,0)$ y las constantes $\lambda_1, \lambda_2, \mu_1, \mu_2$ que definen la tasa de llegada y servicio de los procesos de Poisson definido en las condiciones para el tráfico de voz y datos.

De la figura 43 se pueden obtener las ecuaciones 16 y 17 de Balance detallado para los estados $(0,0)$ y $(1,0)$.

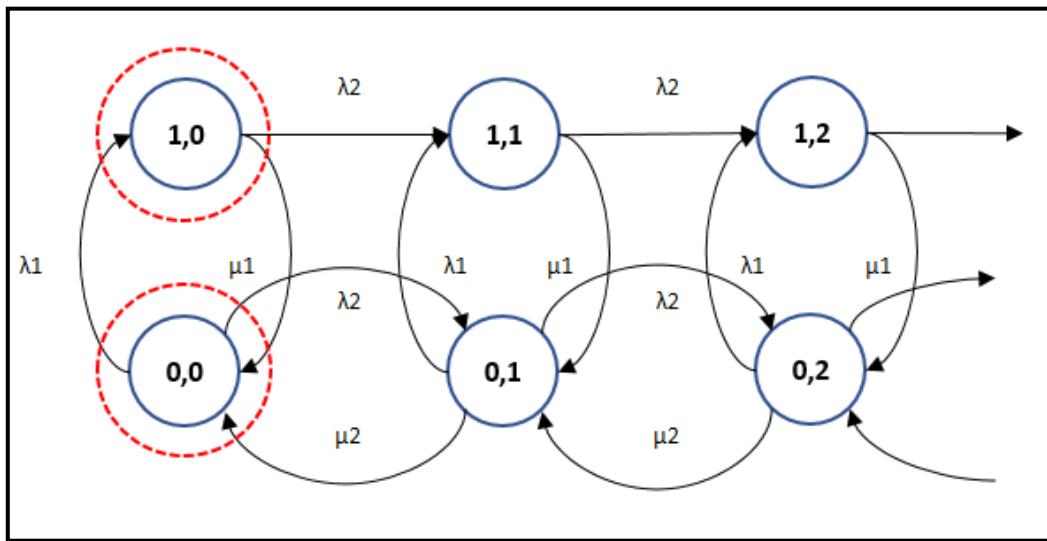


Figura 43. Imagen ampliada Cadena de Markov estados $(0,0)$ y $(1,0)$

$$(\lambda_{-1} + \lambda_2) * P_{0,0} = \mu_1 * p_{1,0} + \mu_{-2} * P_{0,1} \quad (16)$$

$$(\lambda_2 + \mu_1) * P_{1,0} = \lambda_1 * P_{0,0} \quad (17)$$

De la figura 44 se pueden obtener las ecuaciones (18 y 19) de Balance detallado para los estados $(0,i)$ y $(1,i)$.

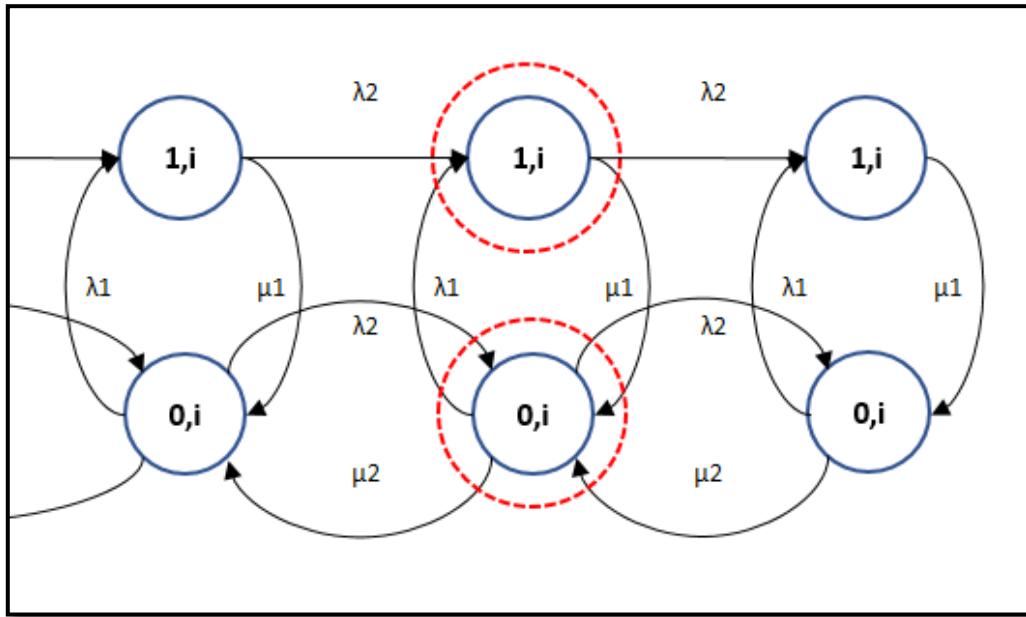


Figura 44. Imagen ampliada Cadena de Markov estados (0,i) y (1,i)

$$(\lambda_1 + \lambda_2 + \mu_2) * P_{0,i} = \lambda_2 * P_{0,i-1} + \mu_1 * P_{1,i} + \mu_2 * P_{0,i+1} \quad (18)$$

$$(\lambda_2 + \mu_1) * P_{1,i} = \lambda_2 * P_{1,i-1} + \lambda_1 * P_{0,i} \quad (19)$$

Después de realizar el desarrollo matemático descrito en el apéndice A, se obtiene la función $G(z)$ de la ecuación 20:

$$G(z) = \frac{\rho_1 + \beta * \rho_2 - \beta * \rho_2 * z + 1}{1 - \rho_2 * z + \beta * \rho_2 - \beta * \rho_2^2 * z + \beta * \rho_2^2 * z^2 - \beta * \rho_2 * z - \rho_1 * \rho_2 * z} * \frac{1 - \rho_2 - \rho_1 * \rho_2}{1 + \rho_1} \quad (20)$$

Para hallar el valor esperado de la variable aleatoria I se hará uso del resultado obtenido en la ecuación 13.2 y de la ecuación 20 como se muestra a continuación:

$$E[I] = \frac{d}{dz} G(z) \text{ con } z = 1$$

$$E[I] = - \left(\frac{\rho_2 + \beta * \rho_2 + \rho_1 * \rho_2 - \beta * \rho_2^2}{\rho_2 + \rho_1 * \rho_2 - 1} + \frac{\beta * \rho_2}{1 + \rho_1} \right) \quad (21)$$

Para poder corroborar el resultado de la ecuación 21 se ejecutará una simulación del sistema, cuyo desarrollo se encuentra descrito en el siguiente capítulo.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Como análisis adicional se puede hallar la probabilidad de bloqueo para paquetes de voz (teniendo en cuenta que los paquetes de datos nunca se rechazan del sistema) de la siguiente forma. Con base en la ecuación 13.1, si la Función se evalua en z=1 se obtiene:

$$G(1) = \sum_{k=0}^{\infty} P_k * z^k |_{z=1} = \sum_{k=0}^{\infty} P_k = 1$$

Debido a que P_k corresponde a la probabilidad de todos los posibles eventos del sistema. A partir de este resultado se evalúa z = 1 en la ecuación 13.3 y en la ecuación 57 del apéndice A, con el fin de hallar la probabilidad de que haya un paquete de voz en el sistema o lo que es lo mismo que haya bloqueo de paquetes de voz debido a que el sistema se encuentra ocupado.

Al evaluar z = 1 en la ecuación 13.3 se obtiene la ecuación 22.

$$G(1) = G_0(1) + G_1(1) = 1 \quad (22)$$

Al evaluar z = 1 en la ecuación 57 se obtiene la ecuación 23.1.

$$G_0(1) = \frac{1 - \beta * \rho_2 * (1 - 1)}{1 + \rho_2 * (\beta - 1 * (1 + \beta * (1 - \rho_2 * (1 - 1)) + \rho_1))} * \frac{1 - \rho_2 - \rho_1 * \rho_2}{1 + \rho_1}$$

$$G_0(1) = \frac{1 - \rho_2 - \rho_1 * \rho_2}{1 - \rho_2 - \rho_1 * \rho_2} * \frac{1}{1 + \rho_1} = \frac{1}{1 + \rho_1} \quad (23.1)$$

Reemplazando $G_0(1)$ en la ecuación 22 se obtiene la ecuación 23.2 para obtener el valor de $G_1(1)$.

$$G_1(1) = 1 - G_0(1)$$

$$G_1(1) = 1 - \frac{1}{1 + \rho_1} = \frac{\rho_1}{1 + \rho_1} \quad (23.2)$$

A modo de aclaración la función $G_1(z)$ es la que recorre los estados de la forma (1,i), es decir todos los estados en los que hay al menos 1 paquete de voz en el sistema, por lo tanto $G_1(1)$ es la probabilidad de que haya 1 paquete de voz en el sistema.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Al existir un paquete de voz en el sistema, la celda no podrá atender ningún otro paquete hasta que el paquete actual sea procesado, por ende, mientras haya un paquete de voz en la celda se bloquearán los paquetes de voz entrantes de acuerdo con la ecuación 24:

$$P_{Bloqueo} = \frac{\rho_1}{1 + \rho_1} \quad (24)$$

Finalmente, para garantizar la estabilidad del sistema es necesario que la probabilidad de que no haya ningún paquete (ni voz ni datos) en la celda sea mayor a cero de acuerdo con la ecuación 60 del apéndice A:

$$P_{0,0} = \frac{1 - \rho_2 - \rho_1 * \rho_2}{1 + \rho_1} > 0$$

$$1 - \rho_2 - \rho_1 * \rho_2 > 0$$

$$\rho_2 * (1 + \rho_1) < 1$$

$$\rho_2 < \frac{1}{1 + \rho_1}$$

Donde finalmente se obtiene la condición de estabilidad descrita en la ecuación 25.

$$\rho_2 < 1 - P_{Bloqueo} \quad (25)$$

8.6.5 Balanceo de Carga entre 2 Celdas

Ya realizado el análisis para una celda individual, el siguiente paso es estudiar el comportamiento de 2 celdas (A y B) que pueden balancear carga (paquetes) entre sí.

Las condiciones para el sistema de 2 celdas serán las siguientes:

- Solo habrá balanceo de paquetes de voz entre las celdas debido a que los paquetes de datos nunca son rechazados del sistema, sino en su lugar se envían a la cola, a la espera de ser procesados.
- Las políticas de servicio de cada celda serán las mismas que en el caso anterior (consideraciones previas número 6).

- Si la celda A está procesando un paquete de voz y llega un paquete de voz antes de descartarlo se validan los siguientes estados en la celda B:
1. Si la celda B no está procesando ningún paquete, ésta procesará el paquete de voz originario de la celda A.
 2. Si la celda B está procesando un paquete de datos, ésta procesará el paquete de voz originario de la celda A y el paquete de datos será enviado a la cola de la celda B.
 3. Si la celda B está procesando un paquete de voz el paquete de voz originario de la celda A se descartará del sistema.

Este mismo razonamiento aplica para la celda B.

De nuevo se hará uso del modelamiento de este sistema mediante una cadena de Markov como se observa en la figura 45.

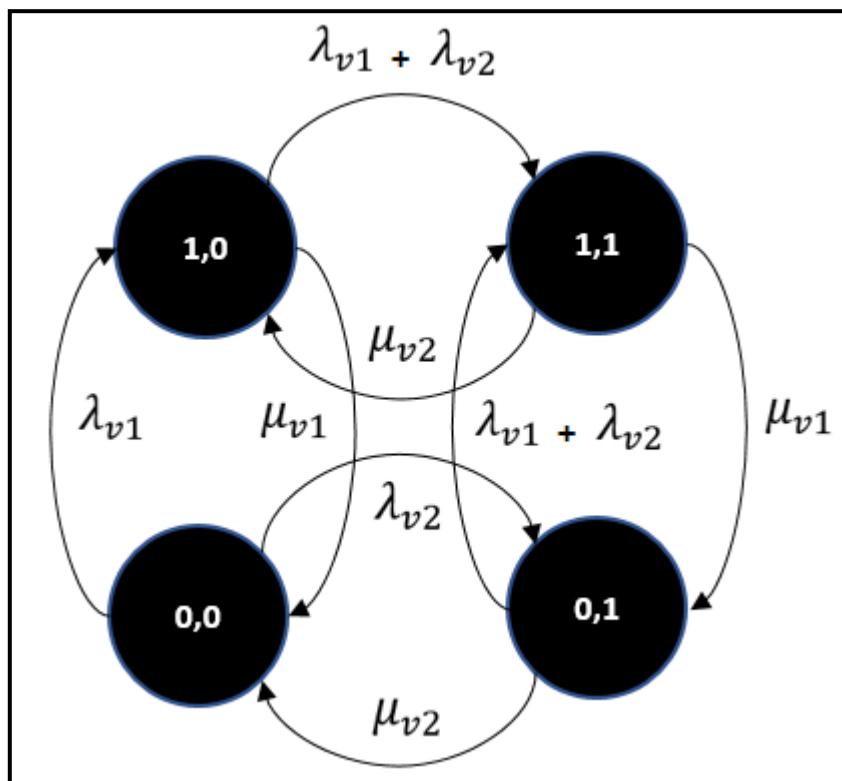


Figura 45. Cadena de Markov para modelar el balanceo de paquetes de voz

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Los estados de la cadena de Markov se definen de la siguiente forma:

- El estado (0,0) indica que ninguna celda está procesando un paquete de voz.
- El estado (1,0) indica que la celda A está procesando un paquete de voz y la celda B no está procesando un paquete de voz.
- El estado (0,1) indica que la celda B está procesando un paquete de voz y la celda A no está procesando un paquete de voz.
- El estado (1,1) indica que las dos celdas están procesando un paquete de voz (un paquete por celda).
- En cualquier otra situación un paquete de voz entrante al sistema es descartado.

Se debe tener en cuenta que los parámetros $\lambda_1, \lambda_2, \mu_1, \mu_2$ son diferentes al del caso analizado en la sección de balanceo. Las transiciones de la cadena de Markov se definen de la siguiente forma:

- λ_{v1} es la tasa de llegada de paquetes de voz para la celda A.
- λ_{v2} es la tasa de llegada de paquetes de voz para la celda B.
- μ_{v1} es la tasa de servicio (salida) para los paquetes de voz originarios de la celda A.
- μ_{v2} es la tasa de servicio (salida) para los paquetes de voz originarios de la celda B.
- Para el caso del balanceo se han agregado las transiciones $\lambda_{v1} + \lambda_{v2}$ desde los estados $P_{0,1}$ Y $P_{1,0}$ hacia el estado $P_{1,1}$. Esto quiere decir que si el sistema se encuentra en el estado (1,0) es posible que la celda B pueda procesar un paquete entrante ya sea de la celda A o de si misma, de igual forma para el estado (0,1).

Ahora bien, el objetivo de este análisis es encontrar la probabilidad de bloqueo para paquetes de voz en el sistema cuando se emplea balanceo de carga, en este caso la probabilidad del estado (1,1).

Basados en la figura 45 las ecuaciones de balance detallado de la cadena de Markov son las siguientes:

$$(\lambda_{v1} + \lambda_{v2}) * P_{0,0} = \mu_{v1} * P_{1,0} + \mu_{v2} * P_{0,1} \quad (26)$$

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

$$(\lambda_{v1} + \lambda_{v2} + \mu_{v1}) * P_{1,0} = \mu_{v2} * P_{1,1} + \lambda_{v1} * P_{0,0} \quad (27)$$

$$(\lambda_{v1} + \lambda_{v2} + \mu_{v2}) * P_{0,1} = \mu_{v1} * P_{1,1} + \lambda_{v2} * P_{0,0} \quad (28)$$

$$(\mu_{v1} + \mu_{v2}) * P_{1,1} = (\lambda_{v1} + \lambda_{v2}) * P_{1,0} + (\lambda_{v1} + \lambda_{v2}) * P_{0,1} \quad (29)$$

$$P_{0,0} + P_{0,1} + P_{1,0} + P_{1,1} = 1 \quad (30)$$

De este sistema de ecuaciones como se describe en el apéndice B, se obtienen los valores para las probabilidades de los 4 estados de la cadena de Markov de la figura 45, siendo inmediato encontrar la probabilidad de Bloqueo en el sistema ($P_{1,1}$):

$$P_{Bloqueo} = \frac{3 * \lambda_{v1}^2 * \lambda_{v2} + 3 * \lambda_{v2}^2 * \lambda_{v1} + \lambda_{v1}^2 * \mu_{v2} + \lambda_{v2}^2 * \mu_{v1} + \lambda_{v1}^3 + \lambda_{v2}^3 + \lambda_{v1} * \lambda_{v2} * \mu_{v1} + \lambda_{v1} * \lambda_{v2} * \mu_{v2}}{\gamma} \quad (31)$$

Donde γ es igual a:

$$\begin{aligned} \gamma = & \lambda_{v1}^3 + 3\lambda_{v1}^2\lambda_{v2} + \lambda_{v1}^2\mu_{v1} + 2\lambda_{v1}^2\mu_{v2} + 3\lambda_{v2}^2\lambda_{v1} + 3\lambda_{v1}\lambda_{v2}\mu_{v1} + 3\lambda_{v1}\lambda_{v2}\mu_{v2} + 3\lambda_{v1}\mu_{v1}\mu_{v2} + \lambda_{v1}\mu_{v2}^2 + \lambda_{v2}^3 + 2\lambda_{v2}^2\mu_{v1} \\ & + \lambda_{v2}^2\mu_{v2} + \lambda_{v2}\mu_{v1}^2 + 3\lambda_{v2}\mu_{v1}\mu_{v2} + \mu_{v1}^2\mu_{v2} + \mu_{v2}^2\mu_{v1} \end{aligned} \quad (32)$$

Tener en cuenta que el valor de λ_1 corresponde a la tasa de llegada de paquete de voz que en este caso será el de λ_{v1} y λ_{v2} y el valor de μ_1 corresponde a μ_{v1} y μ_{v2} .

9. Simulación y resultados

En el presente capítulo se expondrá el diseño del simulador, se contrastará su validez mediante una comparación con los resultados analíticos obtenidos en el capítulo anterior, se ejecutarán simulaciones de escenarios más complejos y se obtendrá un análisis a partir de ello.

9.1 Definición del modelo de simulación

Se definen los 4 puntos clave dentro del sistema de simulación:

Componentes:

- Celdas (se modelarán como servidores que procesan paquetes).
- Usuarios del sistema.
- Paquetes de voz y datos generados por los usuarios y procesados por el sistema.
- Buffer de las celdas para los paquetes de datos: Este componente permitirá manejar los tamaños de las colas. En la simulación dicho buffer se maneja mediante la variable **buffer_size** (apéndice D) y como se analizará más adelante dicha variable permitirá variar los tamaños de la cola con el fin de evaluar el comportamiento del sistema para cada valor.

Variables descriptivas:

- **Tasa de pérdidas:** Esta variable se refiere a la cantidad porcentual de paquetes que no pueden ser transmitidos por cada celda durante el tiempo de simulación. Dicha variable se define mediante la siguiente relación:

$$Tasa\ de\ pérdida\ (%) = 1 - \frac{(\# \ de \ paquetes \ transmitidos)}{(\# \ total \ de \ paquetes \ entrantes \ al \ sistema)} \quad (33)$$

En el apéndice F se describe como se implementa este cálculo dentro de la simulación realizada.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

- **Throughput (Caudal de tráfico):** Esta variable permitirá calcular el caudal de información enviada por unidad de tiempo (bits/s) entre los usuarios del sistema y las celdas. En el apéndice G se describe la forma en la que se realizó este cálculo dentro de la simulación.
- **Handovers en el sistema:** Esta variable permitirá medir la cantidad de handovers realizados en el periodo de simulación para escenarios de balanceo de carga (primer y segundo orden). En el apéndice H se describe la forma en la que se realizó este cálculo dentro de la simulación.

Reglas de Interacción:

- Los usuarios en el sistema se distribuyen inicialmente de forma uniforme sobre el espacio bidimensional, es decir sobre las celdas del sistema.
- Se definirá una zona de cerca-media celda y borde celda de acuerdo con lo mencionado en el capítulo anterior (subcapítulo 8.1). Los usuarios ubicados en la zona de borde de celda serán los únicos que podrán ser transferidos a celdas vecinas en el momento de ejecutar el balanceo de carga.
- Cada celda contará con un número máximo de canales para el procesamiento de los paquetes de voz y datos. Cuando la demanda de servicio supera la capacidad del sistema (tasa de llegada de paquetes supera la tasa de procesamiento de paquetes) la celda actual intentará ejecutar balanceo de primer orden con celdas vecinas, de no poderse intentará ejecutar balanceo de segundo orden, de no ser posible los paquetes de voz entrantes y los paquetes de datos que se encuentren por fuera del tamaño del buffer establecido se descartarán.
- La forma en que se ejecuta el balanceo de usuarios de borde de celda es por medio de handovers inducidos desde la celda origen a la celda destino mediante un ajuste dinámico en los parámetros del proceso de handover. Esto quiere decir que cuando la celda presenta saturación de recursos aumentará dinámicamente el umbral de RSRP para iniciar el proceso de handover con alguna de las celdas adyacentes tal y como se mencionó en el subcapítulo 6.8 con el fin de liberar recursos de radio (Resources Block). Como se explicará más adelante el umbral de RSRP que identificará los usuarios de borde de celda será -83.5 dBm, es decir que los usuarios con un RSRP inferior a dicho valor serán usuarios potenciales para el balanceo entre celdas.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Parámetros Descriptivos:

- Tamaño del buffer de las celdas.
- Tasa de llegada de paquetes al sistema.
- Tasa de procesamiento de paquetes por celda.
- Cantidad de vecindades de balanceo por celda.

En el apéndice C, se encuentra detallado el programa de simulación que representa los componentes del modelo de simulación enunciados anteriormente.

9.2 Verificación del modelo de simulación

[43] [44] Con el fin de verificar que el modelo de simulación es adecuado, se debe contrastar los resultados de la simulación con el resultado analítico descrito en el capítulo anterior. Las bondades de la simulación permiten acotar el escenario, tal y como se modeló en el subcapítulo 8.6.4 y 8.6.5 con el fin de poder hacer la verificación mencionada.

Es importante realizar algunas consideraciones con respecto a la estabilización del sistema previo a la exposición de los resultados.

Bastaría con realizar un solo experimento para poder hallar el valor promedio de paquetes en el sistema, de acuerdo con la definición del valor esperado de una variable aleatoria como se muestra a continuación [45] [41]:

$$E[N] = \lim_{T \rightarrow \infty} \int_0^T N(t) \quad (34)$$

Donde N es la cantidad de paquetes y t es el tiempo del experimento.

Sin embargo, como se evidencia en la ecuación 34 para poder hallar dicho promedio sería necesario realizar la simulación durante un tiempo infinito o en su defecto un tiempo lo suficientemente grande (desconocido) para que el efecto del periodo transitorio no afecte la estadística objeto de estudio.

En contraparte se pueden realizar varios experimentos del proceso estocástico descrito bajo las mismas condiciones con el fin de obtener un estimador del número promedio de

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

paquetes, en donde solo hace falta escoger apropiadamente el tiempo de simulación para cada experimento.

Como se puede evidenciar en las figuras 46, 47 y 48 el estimador de paquetes promedio (línea negra) en el sistema se estabiliza bajo ciertos parámetros de entrada (tasa de llegadas y de servicio) alrededor de las 20000 unidades de tiempo, es decir que se obtiene un criterio para escoger el tiempo de simulación de cada experimento, por ello para el caso de estudio se escogerá 20000 unidades de tiempo como tiempo de simulación.

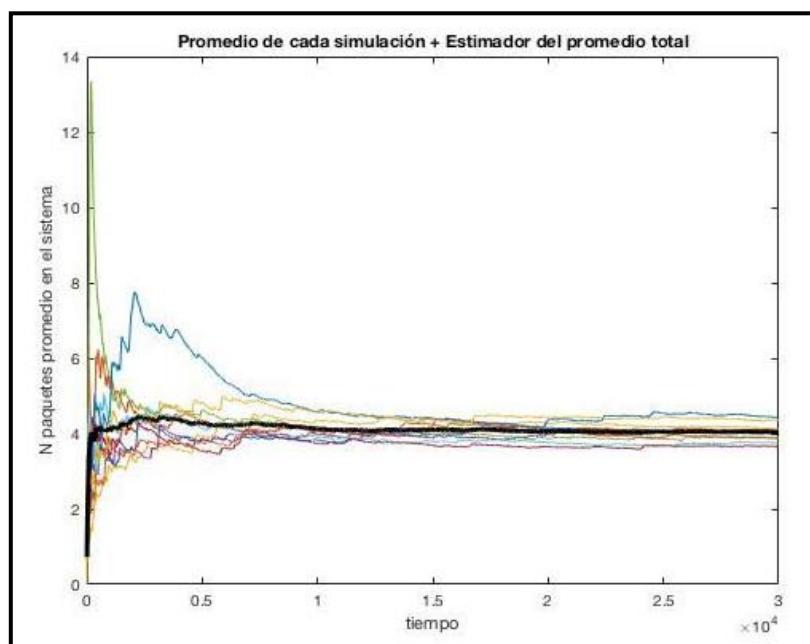


Figura 46. Promedio de cada simulación + Estimador del promedio total. $\lambda_1 = 1, \lambda_2 = 1, \mu_1 = 2, \mu_2 = 2$

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

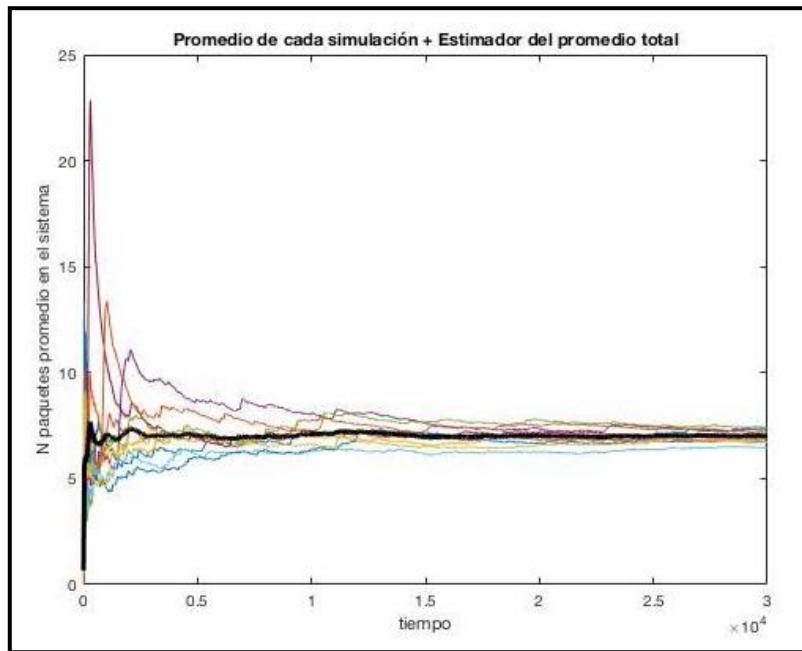


Figura 47. Promedio de cada simulación + Estimador del promedio total. $\lambda_1 = 2, \lambda_2 = 2, \mu_1 = 3, \mu_2 = 4$

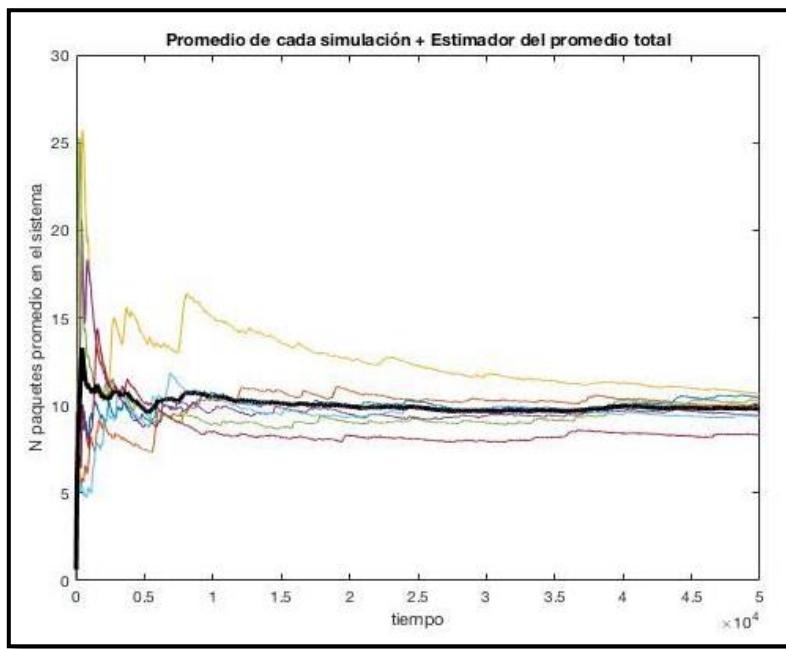


Figura 48. Promedio de cada simulación + Estimador del promedio total. $\lambda_1 = 1, \lambda_2 = 2, \mu_1 = 3, \mu_2 = 3$

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Teniendo en cuenta que el número promedio de paquetes en el sistema toma una distribución normal con media μ y varianza $\frac{\sigma^2}{n}$ de acuerdo con lo establecido en el Teorema del Límite Central [46] , sería adecuado estimar un intervalo de confianza para dicha estadística y confrontarla con el valor teórico de la ecuación 21.

Sea el estimador de paquetes promedio en el sistema:

$$\hat{N} = \frac{1}{n} * \sum_{i=1}^n N_i \quad (35.1)$$

Donde N_i corresponde al número promedio de paquetes de la i-ésima simulación después de transcurrido un periodo de tiempo lo suficientemente grande.

El primer paso será normalizar la variable aleatoria N de tal forma que la media sea 0 y la varianza sea 1 de la siguiente forma:

$$Z = \frac{\hat{N} - \mu}{\frac{\sigma}{\sqrt{n}}} \quad (35.2)$$

Sin embargo, esta normalización implica conocer la varianza de la distribución (información que no se tiene). En lugar de ello se deberá hallar un estimador S para la varianza con la información de las muestras obtenidas en las simulaciones, y a partir de ello proponer una nueva variable normalizada T [41].

El estimador S propuesto para la varianza se calculará a partir de los valores observados en cada experimento (N_i) [47] [48] . La forma en que se calculará el estimador S es la siguiente:

$$S = \frac{1}{n-1} * \sum_{i=1}^n (N_i - \hat{N})^2 \quad (36)$$

Así la nueva variable aleatoria normalizada T en términos del estimador S para la varianza queda expresada:

$$T = \frac{\hat{N} - \mu}{\frac{S}{\sqrt{n}}} \quad (37)$$

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Es importante mencionar que la variable T tiene una distribución T-Student debido a la inclusión del estimador S en lugar de σ . A medida que la cantidad de experimentos tiende a infinito, los grados de libertad también aumentan y la distribución de la variable T tenderá en distribución a una variable normal. Sin embargo para el desarrollo del presente trabajo la cantidad de experimentos n será finita y por ende se trabajará con la variable T en lugar de Z. [47]

Como T posee una distribución simétrica con respecto a la media, es apropiado definir un intervalo de confianza donde con cierta probabilidad p se pueda encontrar el valor de la media del estimador N . El intervalo de confianza se puede expresar en términos del α . 100% percentil [41] de la siguiente manera:

$$P \left[|T| \leq T_{\frac{\alpha}{2}} \right] = 1 - \alpha \quad (38)$$

$$P \left[-T_{\frac{\alpha}{2}} \leq T \leq T_{\frac{\alpha}{2}} \right] = 1 - \alpha$$

$$P \left[-T_{\frac{\alpha}{2}} \leq \frac{\hat{N} - \mu}{\frac{S}{\sqrt{n}}} \leq T_{\frac{\alpha}{2}} \right] = 1 - \alpha$$

$$P \left[\hat{N} - \frac{T_{\frac{\alpha}{2}} * S}{\sqrt{n}} \leq \mu \leq \hat{N} + \frac{T_{\frac{\alpha}{2}} * S}{\sqrt{n}} \right] = 1 - \alpha \quad (39)$$

Es decir que asignando un valor a α es posible hallar la probabilidad con la que se puede encontrar el valor promedio de paquetes en el intervalo de confianza definido. En la tabla 11 se puede evidenciar para distintos valores de $\lambda_1, \lambda_2, \mu_1, \mu_2$ y tiempo de simulación de 20000 unidades, la estimación del intervalo de confianza en contraste con el valor teórico del número promedio de paquetes. El código empleado para calcular el intervalo de confianza se encuentra en el apéndice C.

Parámetros de Entrada				Intervalo	Nivel Confianza	Número simulaciones	Promedio Estimado	Promedio Teórico	Error
μ_1	μ_2	λ_1	λ_2						
1	1	2	2	95%	+/- 4.952%	7	3.8748	3.67	5.5%
2	2	3	4	95%	+/- 3.062%	5	6.8302	6.6	3.48%
1	2	3	3	95%	+/- 4.851%	26	9.6105	9.5	1.16%

Tabla 11. Validación resultados teóricos y por simulación del número promedio de paquetes

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

En la tabla 12 se puede evidenciar el bloqueo de paquetes de voz teórico en contraste con el valor obtenido mediante simulación.

Parámetros de Entrada				Bloqueo por simulación	Bloqueo Teórico	Error
λ_1	λ_2	μ_1	μ_2			
1	1	2	2	0.338	0.333	1.5%
2	2	3	4	0.396	0.4	1%
1	2	3	3	0.261	0.25	2.7%

Tabla 12. Validación resultados teóricos y por simulación del bloqueo de paquetes de voz.

Para corroborar el caso de la inestabilidad del sistema se considerará la condición descrita en la ecuación 25 y se elegirá un valor de ρ_2 menor a $P_{No\ Bloqueo}$ con el fin de validar el comportamiento de la cantidad promedio de paquetes en el sistema.

Supóngase $P_{Bloqueo} = 0.4$ y por ende $\rho_1 = 0.333$ se escoge un valor de $\rho_2 = 0.75$ y se realiza la simulación para verificar que el estado del sistema no converge cuando $t \rightarrow \infty$ como se observa en la figura 49.

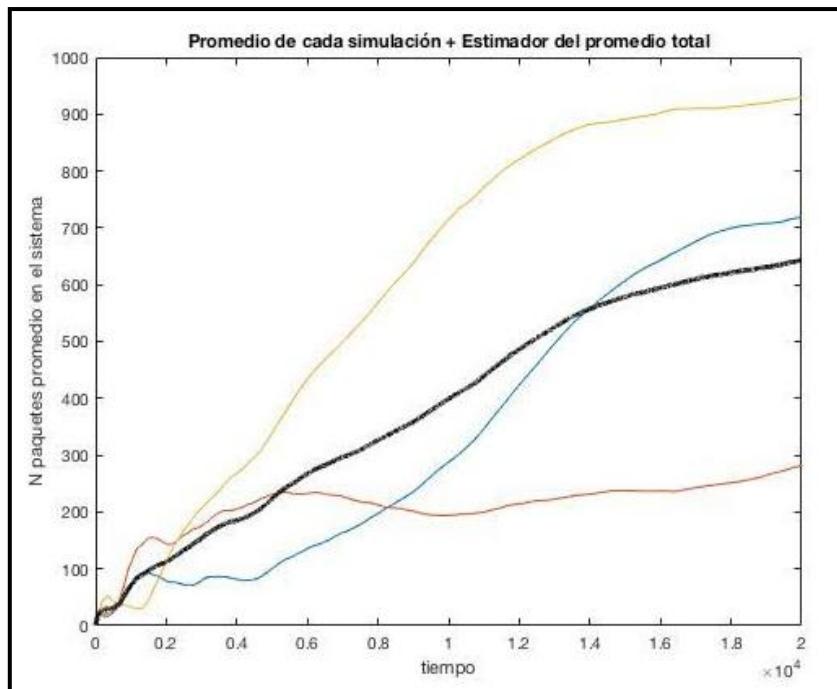


Figura 49. Caso de inestabilidad en el sistema un valor de ρ_2 menor a $P_{No\ Bloqueo}$

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Por último, En la tabla 13 se puede evidenciar el bloqueo de paquetes de voz teórico bajo el escenario de balanceo de carga en contraste con el valor obtenido mediante simulación.

Parámetros de Entrada				Bloqueo por simulación	Bloqueo Teórico	Tasa de Error porcentual
T	λ_2	μ_1	μ_2			
1	1	2	2	0.21003	0.2	5 %
2	2	3	3	0.2758	0.2759	0.3 %
3	3	4	4	0.3123	0.3101	0.7 %
4	4	5	5	0.3299	0.3296	0.9 %
2	2	5	5	0.1747	0.1509	10 %

Tabla 13. Validación resultados teóricos y por simulación del bloqueo de paquetes de voz utilizando balanceo de carga

9.3 Condiciones previas del primer escenario a simular

Validando el funcionamiento del simulador con respecto al modelo analítico obtenido en el capítulo anterior, en este subcapítulo se expondrán las condiciones del sistema bajo las cuales se realizarán las simulaciones de escenarios más complejos.

Para el primer escenario de simulación las condiciones de simulación se describen a continuación:

- El sistema consta de 4 eNodeB, cada uno con 3 celdas para un total global en el sistema de 12 celdas.
- Cada eNodeB cuenta con un sistema radiante de 3 sectores (uno por celda). Las características de las antenas se pueden observar en la tabla 14.

Electrical Properties	
Frequency Range (MHz)	1920 - 2170
Electrical Tilt (°)	0
Mechanical Tilt (°)	0
Gain (dBi)	10
Horizontal 3dB beam width (°)	120
Vertical 3dB beam width (°)	+/- 5
Front to Back Ratio +/- 30° (dB)	> 27
Impedance (Ohm)	50
Antenna Height (m)	5

Tabla 14. Propiedades Eléctricas y físicas de los sistemas radiantes

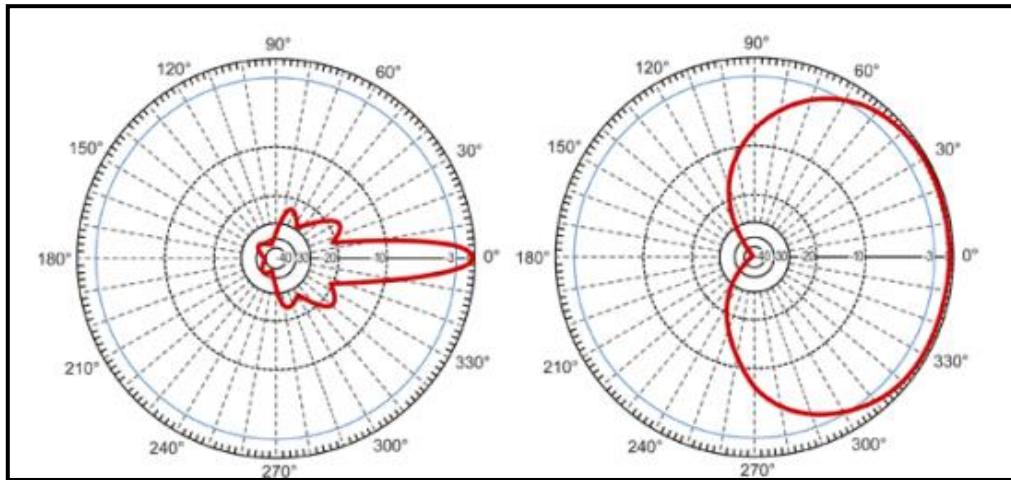


Figura 50a. Aproximación del patrón de radiación [49] de acuerdo con las características eléctricas y físicas de los sistemas radiantes expuestos en la tabla 14

Mediante una linealización del patrón horizontal de radiación de la figura 50a, se logra obtener el mapa de cobertura de la figura 50b obtenido mediante la medición del RSRP. La implementación de este código se encuentra descrita en el apéndice E.

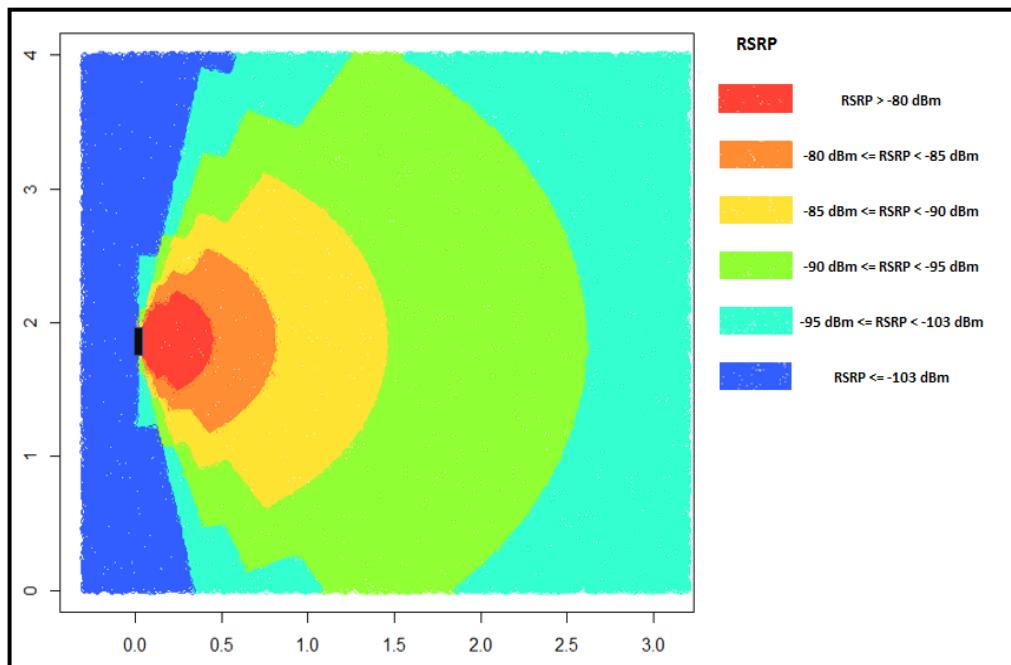


Figura 50b. Mapa de cobertura de un sistema radiante con un ángulo de abertura horizontal de 120° a través de la medición del RSRP, bajo un escenario que solo presenta pérdidas de espacio libre a partir de una linealización del patrón de radiación de la figura 50a. y la potencia del transmisor es 20 dBm. Las unidades de la gráfica se dan en Km.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Es importante resaltar que durante la simulación se grafican las celdas hexagonales (figura 52) para representar el área de cobertura por celda, sin embargo, el patrón de radiación se aproxima al de la figura 50 y no a una celda hexagonal.

- El sistema contará con 2000 usuarios.
- La frecuencia de la portadora será 1900 MHz y el ancho de banda será 1.4 MHz (6 Resource Block).
- Las vecindades de balanceo de carga dentro de la red obedecen la topología de la figura 38.
- Antes de iniciar la simulación, la ubicación bidimensional de los usuarios en el sistema obedece una distribución uniforme como se observa en la figura 51, es decir que inicialmente el valor promedio de usuarios por celda es el mismo (aproximadamente 167 usuarios por celda).

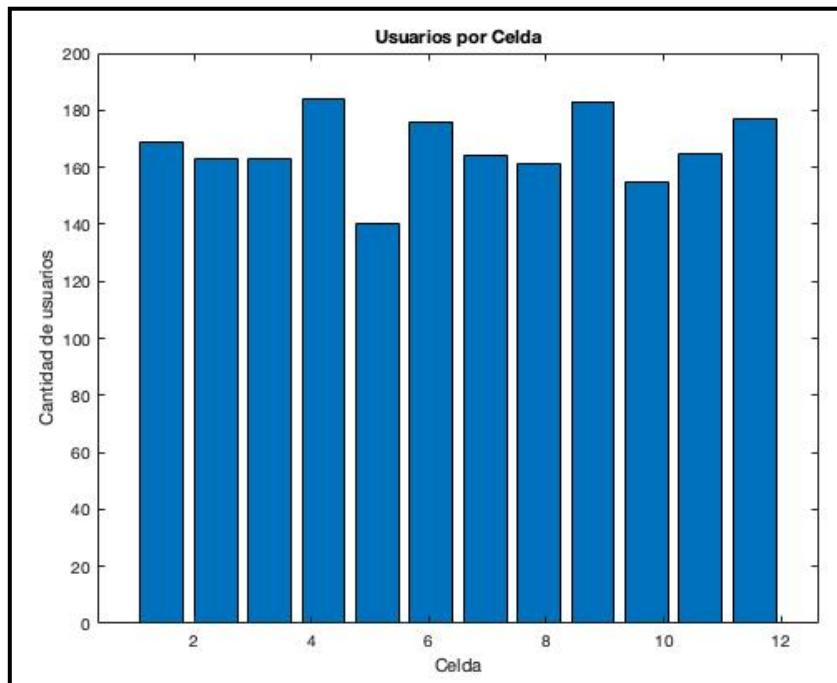


Figura 51. Distribución Uniforme de la asignación inicial de usuarios por celda

- Se considera que las pérdidas solo se presentan por pérdidas del espacio libre, por tal motivo la potencia de recepción en los UE viene dada por la siguiente fórmula:

$$\text{Received Power (dBm)} = \text{Transmitter Power (dBm)} + \text{Antenna Gain (dBi)} - \text{Free Path Loss(dB)} \quad (40)$$

Para los escenarios a simular, la potencia del transmisor será 20dBm y la ganancia de la antena será 10dBi de acuerdo con las especificaciones de la tabla 14. Las pérdidas de espacio libre se calculan de la siguiente forma:

$$\text{Free Path Loss}(dB) = 92.45 + 20 * \log(f[GHz]) + 20 * \log(d[Km]) \quad (41) \quad [50]$$

- Solamente los usuarios de borde de celda serán usuarios potenciales para ejecutar el balanceo de carga entre celdas adyacentes. Para ello se definirá que los usuarios de borde de celda serán los que cuenta con un nivel de RSRP inferior a -83.5 dBm. En la figura 52 se evidencian en rojo los usuarios que podrán ser trasladados a celdas vecinas cuando se presenta saturación.

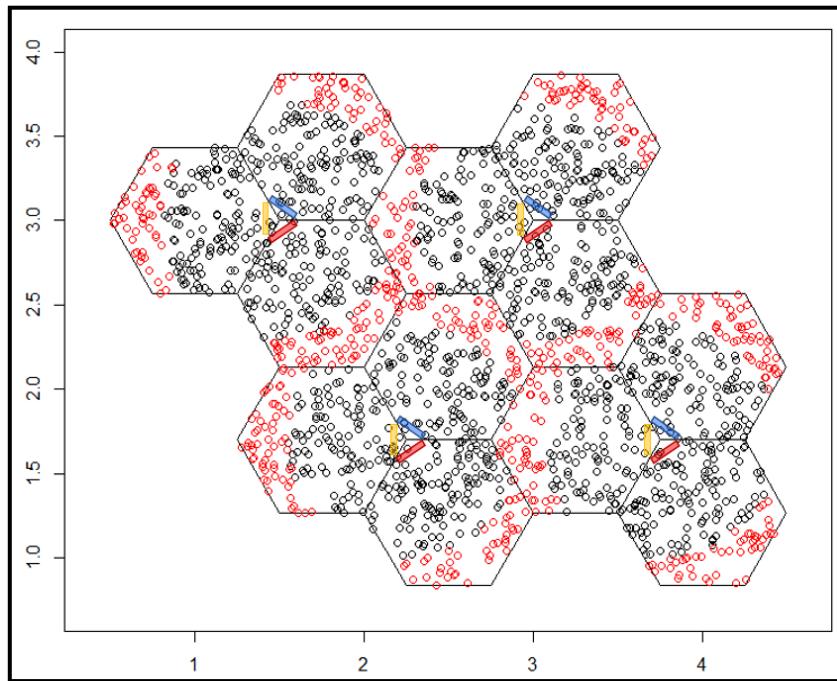


Figura 52. Ubicación inicial de los usuarios dentro del sistema con discriminación entre los usuarios de media celda (color negro) y usuarios de borde de celda (color rojo).

- Los tiempos de servicio se mantendrán constantes tanto para los paquetes de voz como para los paquetes de datos ($\mu_1 = \mu_2 = 1$). Se escogió este valor de forma que la relación con las tasas de llegadas quede normalizada.

Si en promedio cada unidad de simulación para los tiempos de servicio correspondiese a 50ms se esperaría que cada paquete IP (más los encabezados propios del estándar LTE **subcapítulo 6.7.1**) tuviese un tamaño promedio aproximado de 50000 bits (6.2KB). Es importante mencionar que si el paquete IP en

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

formato IPv4 presenta una longitud mayor 65535 Bytes éste deberá ser fragmentado en paquetes más pequeños. [51]

- Los tiempos de llegada de paquetes tanto para el tráfico de voz como el de datos será variable.
- Las celdas configuran una modulación de 64QAM para usuarios de media celda y de 16QAM para usuarios de borde de celda.
- Como método para evitar la interferencia entre celdas (para usuarios de borde celda) se implementará el mecanismo de ICIC (Inter-Cell Interference Coordination). [52]

9.4 Estabilidad del sistema

Al momento de ejecutar las simulaciones es de interés conocer en qué condiciones el sistema será estable, donde se debe considerar un escenario con cola de tamaño infinito y validar si el sistema converge o no converge, es decir que la cantidad de paquetes en el sistema (paquetes procesados más los paquetes en la cola de datos) no crezca indefinidamente. Esto debido a que las simulaciones se deben ejecutar sobre escenarios convergentes con el fin de que los resultados permitan establecer una comparación objetiva a medida que varían las tasas de llegada de paquetes.

Para ello se simularán los escenarios de la tabla 15 cuatro veces por un tiempo de 4000 unidades.

Lambda Voice	Lambda Data	Convergencia
40	30	Si
40	40	Si
40	50	Si
40	60	No

Tabla 15. Escenarios para estimar condiciones de estabilidad a través de simulación

Cuyos resultados se pueden observar en las figuras 53, 54, 55, 56 y en la tabla 14.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

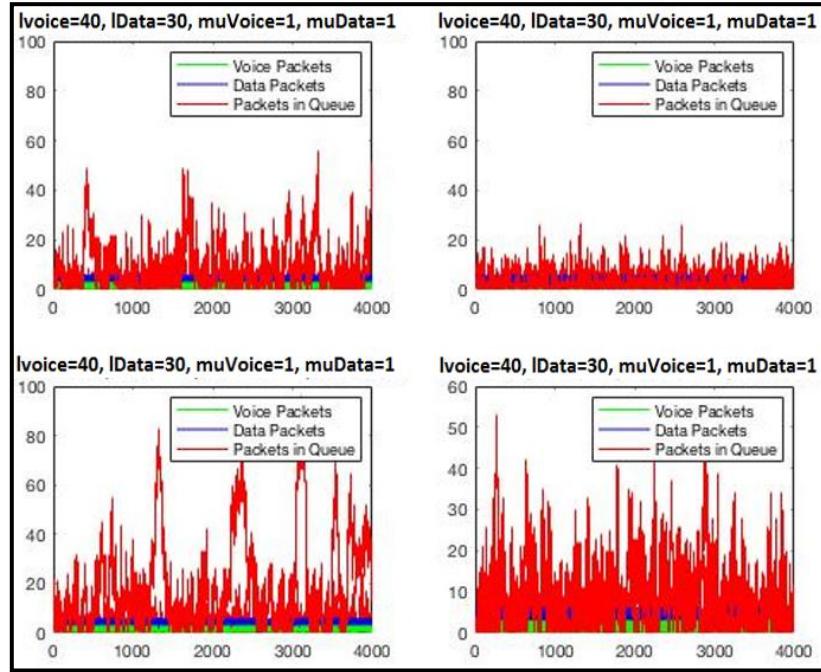


Figura 53. Escenario con cola infinita, lambda Voice = 40, lambda Data = 30

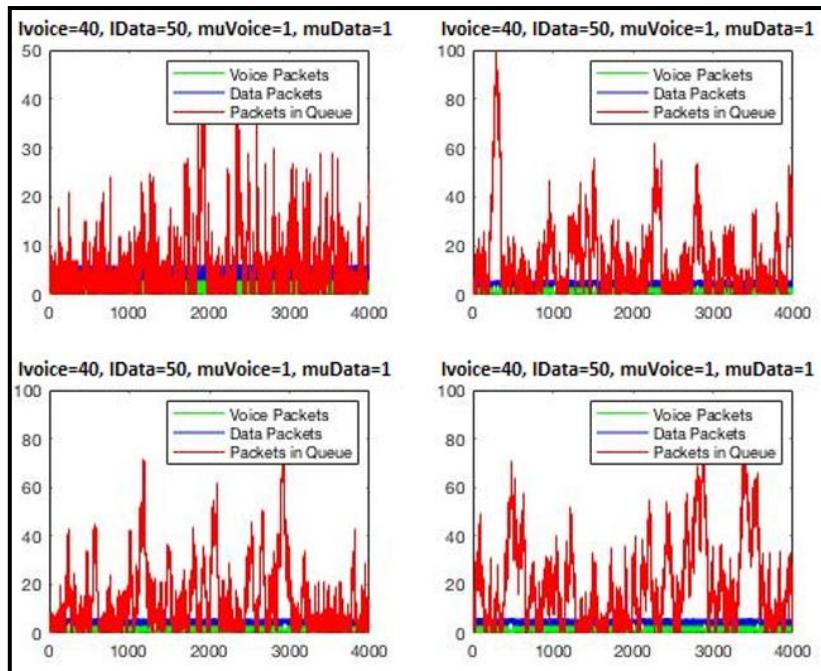


Figura 54. Escenario con cola infinita, lambda Voice = 40, lambda Data = 40

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

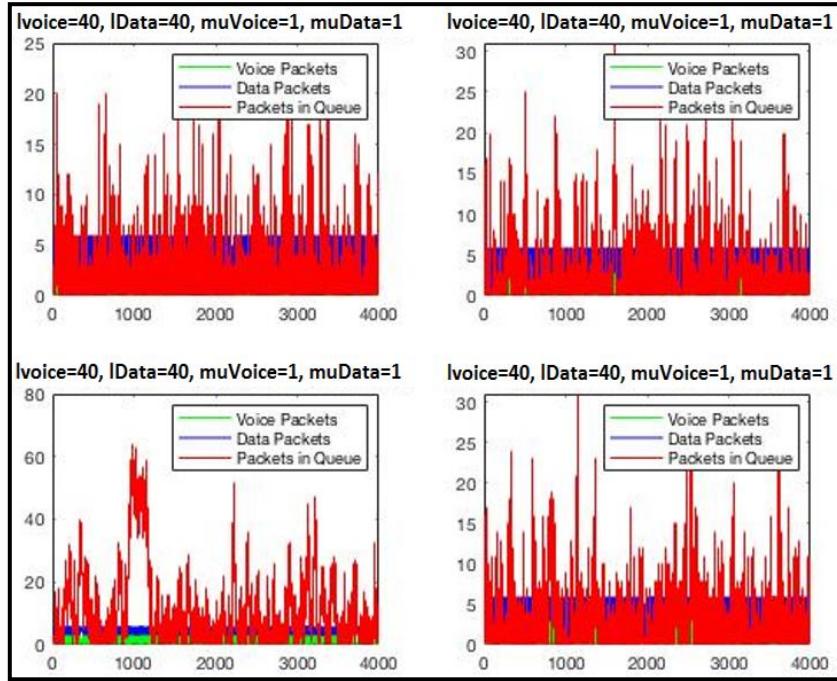


Figura 55. Escenario con cola infinita, lambda Voice = 40, lambda Data = 50

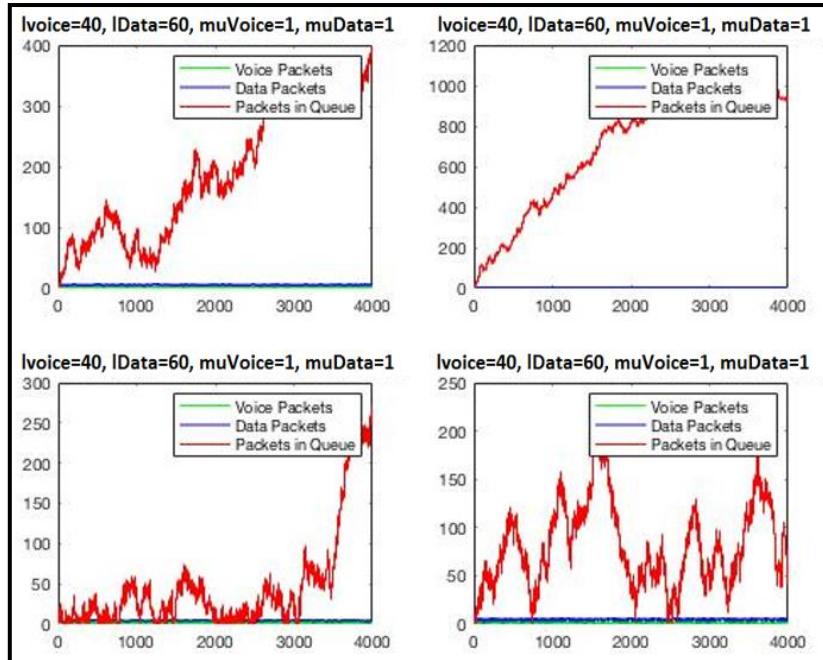


Figura 56. Escenario con cola infinita, lambda Voice = 40, lambda Data = 60

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Basados en los resultados expuestos es sencillo notar que en el escenario de lambda Voice = 40 y lambda Data = 60 la cola del sistema empieza a crecer indefinidamente. Se podría considerar que las condiciones límites de convergencia se obtienen con lambda Voice = 40 y lambda Data = 50 sin embargo, esto no se puede considerar de forma certera ya que para ello habría que ejecutar un número infinito de simulaciones y validar para cada uno si la cantidad de paquetes en el sistema converge.

Con el fin de no incluir un escenario no convergente se considerarán los siguientes escenarios para realizar el análisis (Tabla 16):

Lambda Voice	Lambda Data	10	20	30	40
10		✓	✓	✓	✓
20		✓	✓	✓	✓
30		✓	✓	✓	✓
40		✓	✓	✓	✓

Tabla 16. Escenarios seleccionados para ser simulados y analizados

9.5 Tipo de Distribución para la Tasa de pérdidas y el Throughput promedio

Con el fin de observar la distribución de la tasa de pérdidas y del throughput promedio por celda, se realizarán 300 simulaciones con parámetros $\lambda_1 = 10, \lambda_2 = 10, \mu_1 = 1, \mu_2 = 1$ y se analizará el comportamiento de las dos variables mencionadas.

En las figuras 57 y 58 se observan las distribuciones de pérdida de paquetes (tasa porcentual %) para los usuarios de media celda y de borde en cada una de las celdas del sistema, mientras que en la figura 59 se observa la distribución del throughput promedio por celda.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

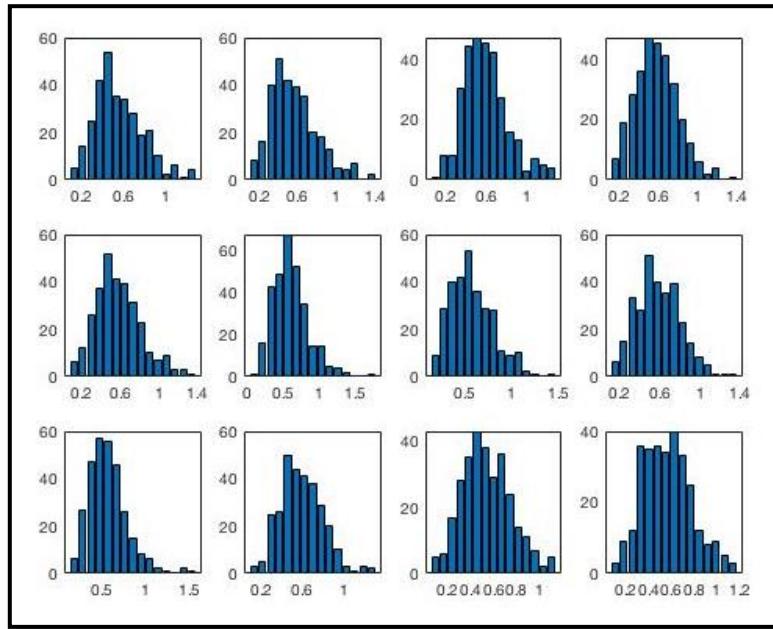


Figura 57. Distribución de la pérdida de paquetes por celda, para usuarios de media celda

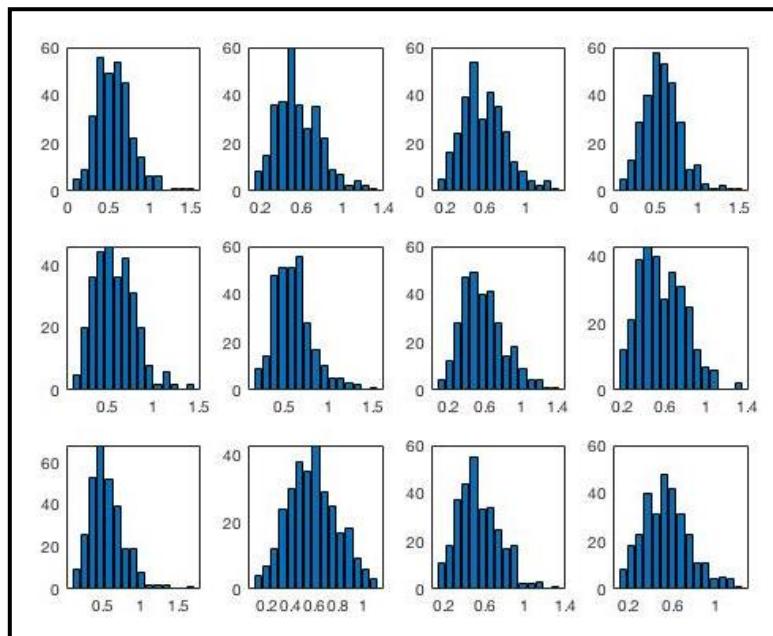


Figura 58. Distribución de la pérdida de paquetes por celda, para usuarios de borde de celda

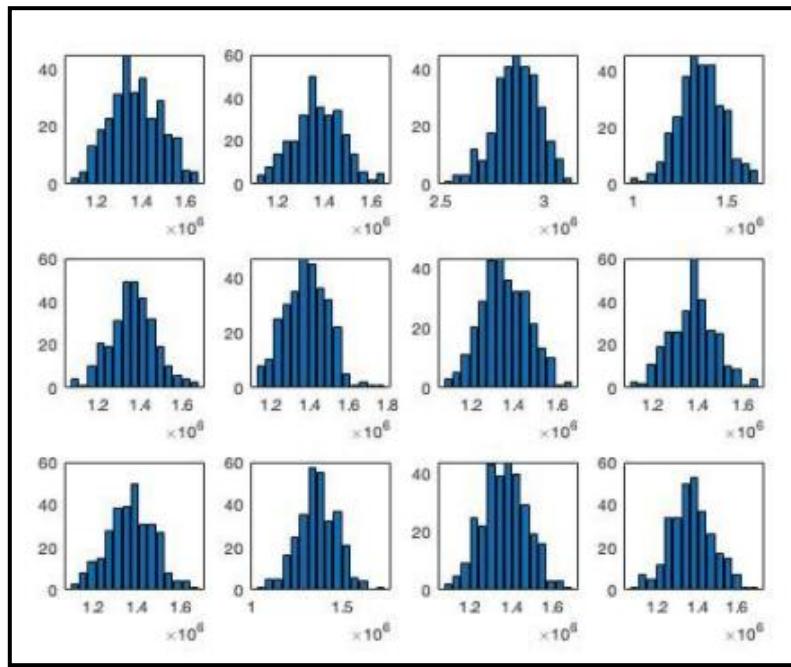


Figura 59. Distribución del throughput promedio por celda

Se puede evidenciar que la distribución de la pérdida de paquetes y del throughput promedio tiende a ser normal, lo cual concuerda con el teorema del límite central que establece que cuando el tamaño de la muestra es lo suficientemente grande, la distribución de las medias sigue aproximadamente una distribución normal. Para el caso práctico, en el siguiente capítulo se calculará la estimación de la media a partir del promedio de 7 experimentos por escenario de acuerdo con la ecuación 34.

9.6 Simulaciones y Análisis

En este subcapítulo se ejecutarán 1344 simulaciones divididas en 192 escenarios descritos en la tabla 15 variando los tamaños de la cola (cero, cinco, diez y sin límite) y los 3 escenarios de balanceo (sin balanceo, con balanceo de primer orden y balanceo de segundo orden). Las variables de evaluación obtenidas en las simulaciones serán: Cantidad de Handovers para balanceo de primer y segundo orden, Tasa de pérdidas global en el sistema y Throughput promedio por celda.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

De la figura 60 a la figura 68 se puede evidenciar como varía el número de Handovers de primer y segundo orden con respecto a la variación de la tasa de llegada de paquetes de voz y datos (λ_1, λ_2). Las gráficas ubicadas a la izquierda corresponden a la cantidad de handovers empleados en el balanceo de carga de primer orden y las gráficas ubicadas a la derecha corresponden a la cantidad de handovers empleados en el balanceo de carga de segundo orden.

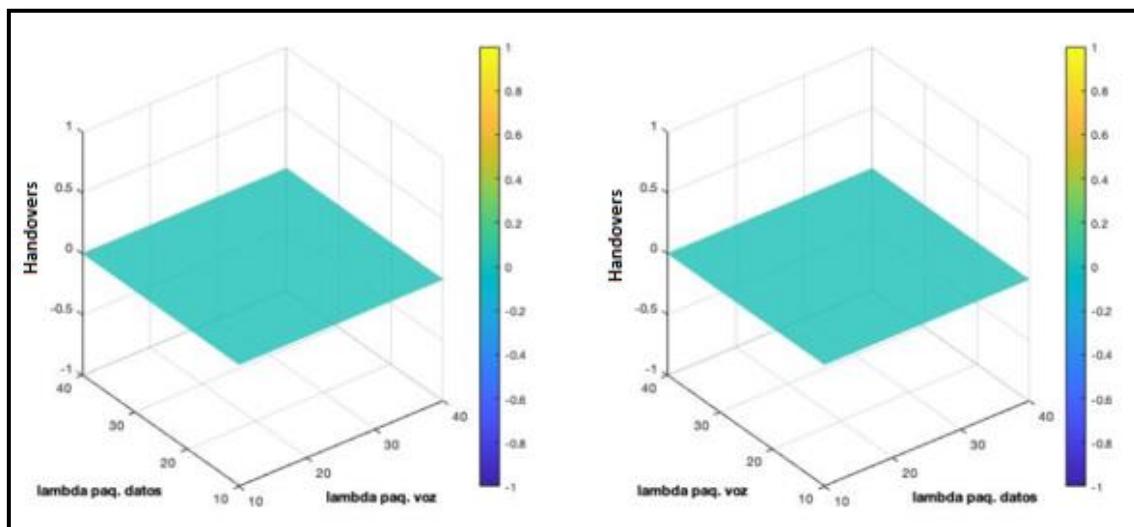


Figura 60. Número de Handovers sin la implementación del modelo de balanceo de carga

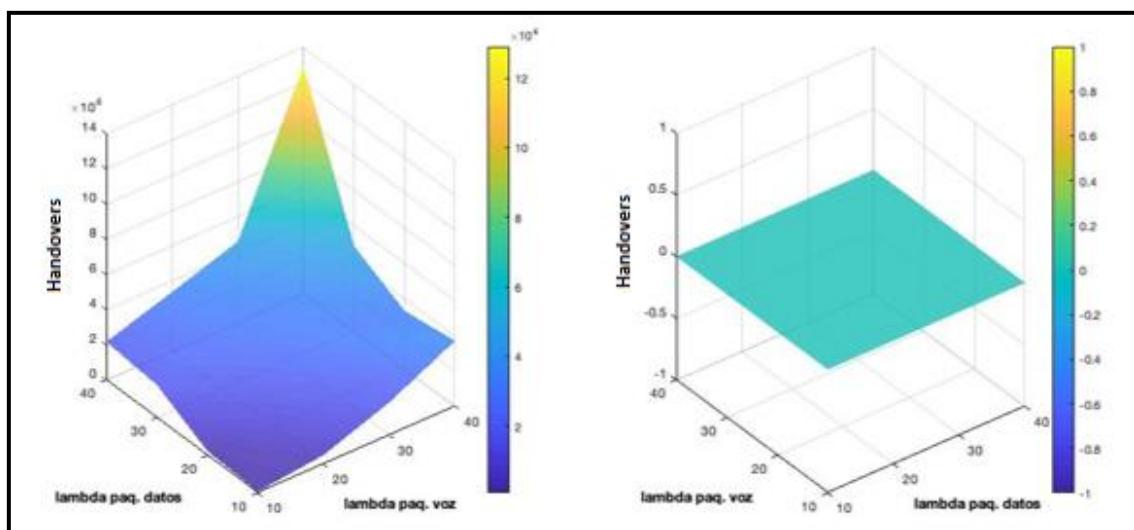


Figura 61. Número de Handovers de primer y segundo orden respectivamente, bajo un escenario de balanceo de primer orden con tamaño de la cola de paquetes de datos igual a cero.

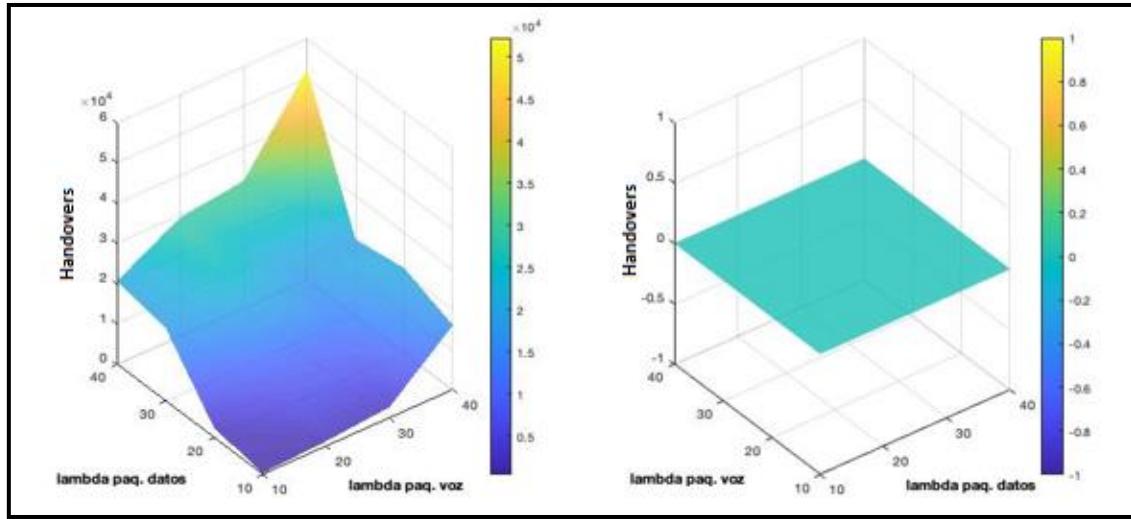


Figura 62. Número de Handovers de primer y segundo orden respectivamente, bajo un escenario de balanceo de primer orden con tamaño de la cola de paquetes de datos igual a cinco.

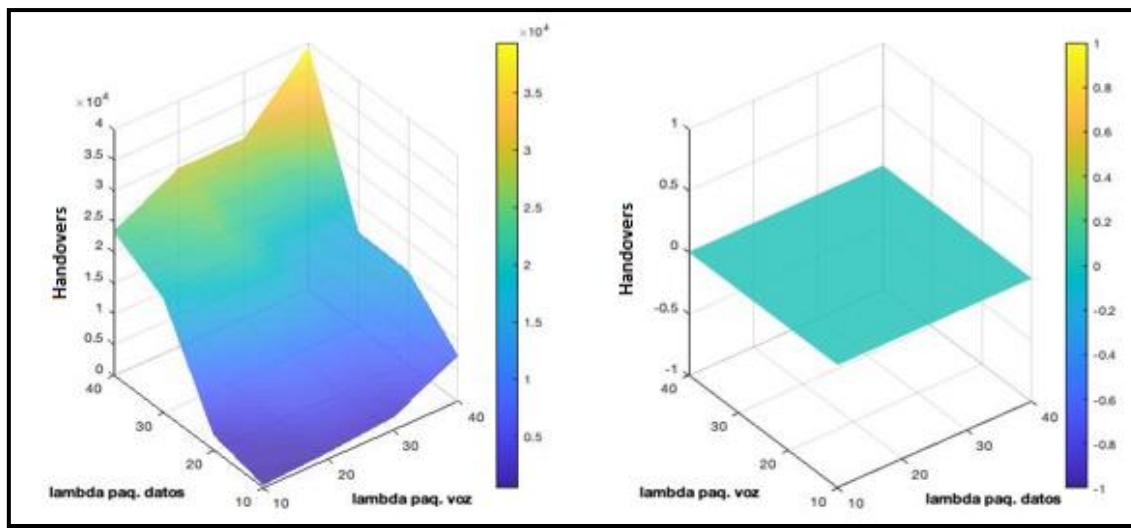


Figura 63. Número de Handovers de primer y segundo orden respectivamente, bajo un escenario de balanceo de primer orden con tamaño de la cola de paquetes de datos igual a diez.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

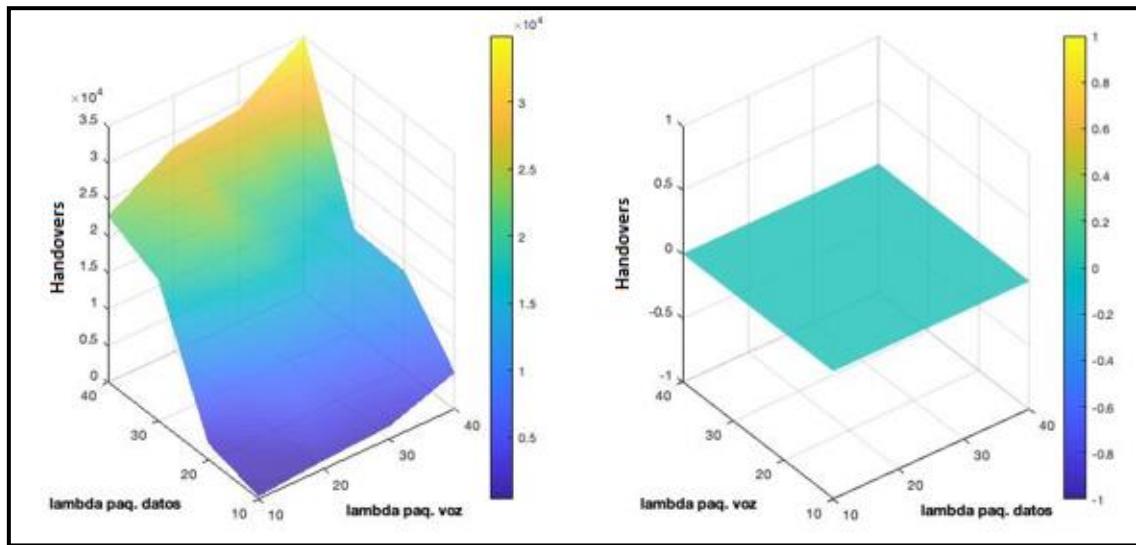


Figura 64. Número de Handovers de primer y segundo orden respectivamente, bajo un escenario de balanceo de primer orden con tamaño de la cola de paquetes de datos sin límite.

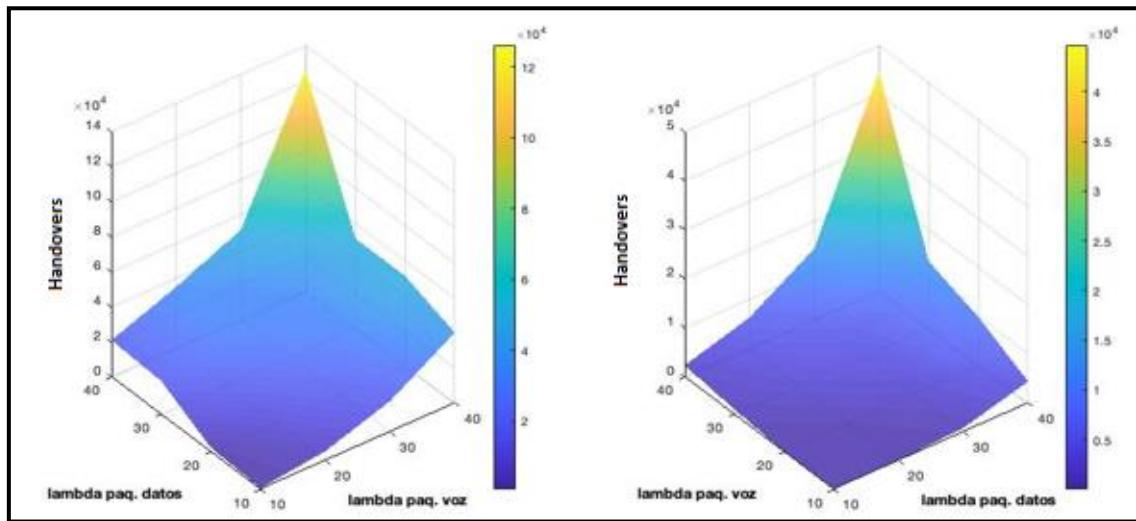


Figura 65. Número de Handovers de primer y segundo orden respectivamente, bajo un escenario de balanceo de segundo orden con tamaño de la cola de paquetes de datos igual a cero.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

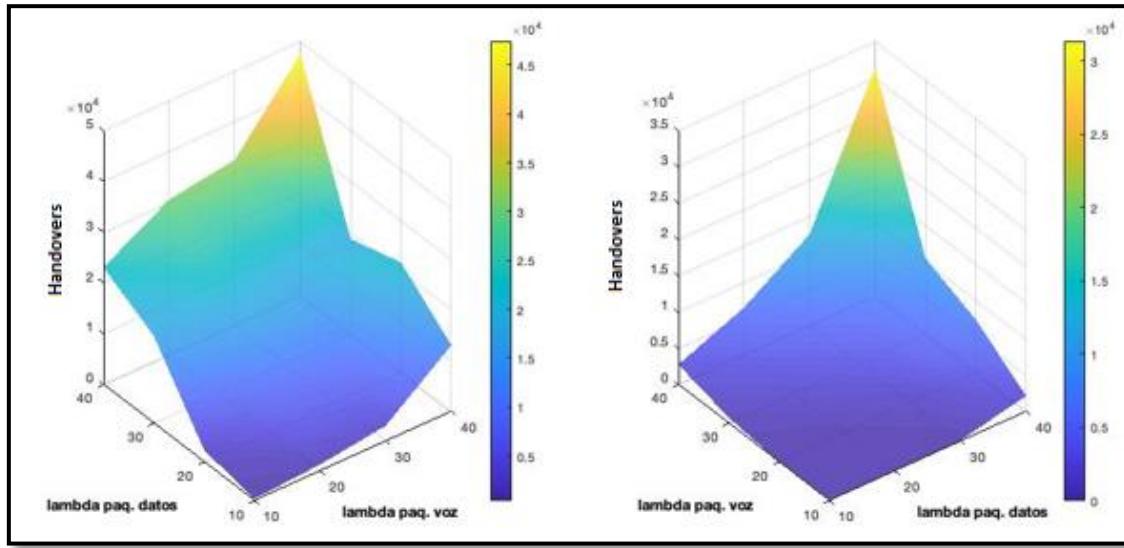


Figura 66. Número de Handovers de primer y segundo orden respectivamente, bajo un escenario de balanceo de segundo orden con tamaño de la cola de paquetes de datos igual a cinco.

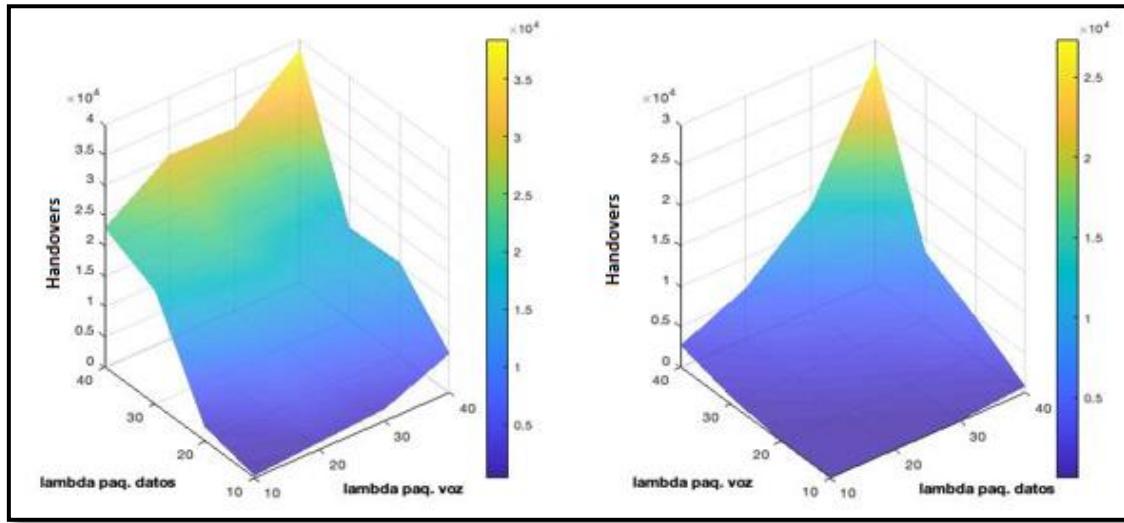


Figura 67. Número de Handovers de primer y segundo orden respectivamente, bajo un escenario de balanceo de segundo orden con tamaño de la cola de paquetes de datos igual a diez.

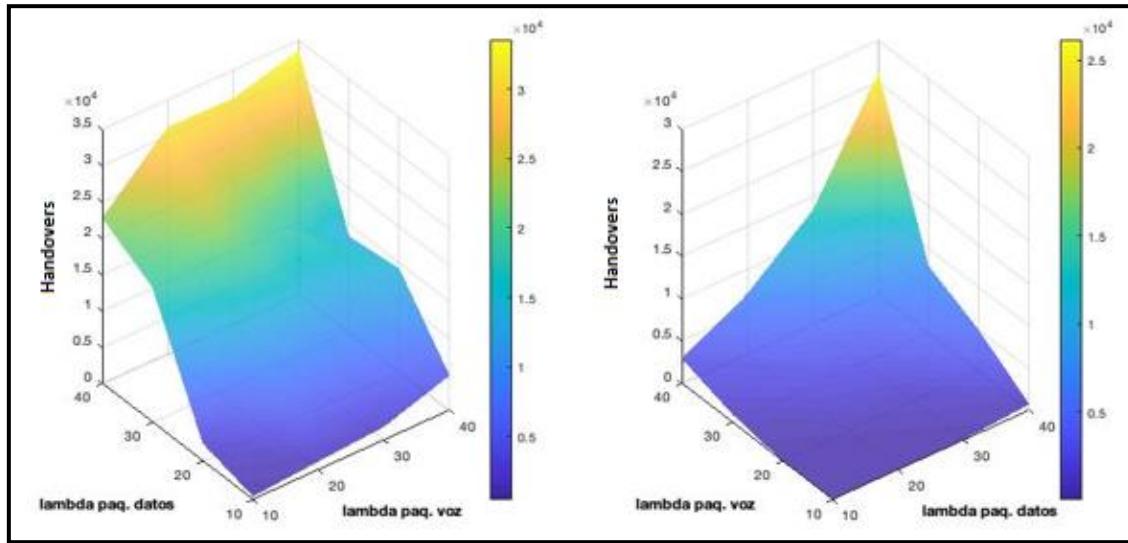


Figura 68. Número de Handovers de primer y segundo orden respectivamente, bajo un escenario de balanceo de segundo orden con tamaño de la cola de paquetes de datos sin límite.

De estos primeros resultados es claro evidenciar que la cantidad de handovers empleados para ejecutar el balanceo de carga se deben a 3 factores principales:

- Tasa de llegada de paquetes para el tráfico de voz y datos. A medida que la tasa es mayor la saturación dentro de la celda también lo es, por lo cual los umbrales para la activación del mecanismo de balanceo se alcanzan con mayor frecuencia y por ende la cantidad de handovers.
- El tamaño de la cola (buffer). Cuando el tamaño se hace más grande la cantidad de handover correspondientes a balanceo de carga se hace menor, debido a que las pérdidas en el sistema disminuyen y la saturación (carga) en cada celda se deberá solamente al tráfico de voz por lo cual la activación del procedimiento de balanceo es menor en comparación a escenarios con tamaño de colas más pequeños.
- Tipo de balanceo. Dependiendo del tipo de balanceo variará la cantidad y el tipo de handover. En el escenario donde no hay balanceo la cantidad de handovers será cero, en el escenario donde hay balanceo de primer orden solo habrán handovers de primer orden y en el escenario donde hay balanceo de segundo orden habrán handovers de primer orden y de segundo orden.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

De la figura 69 a la figura 80 se puede evidenciar como varía la tasa de pérdidas con respecto a la variación de la tasa de llegada de paquetes de voz y datos (λ_1, λ_2). Las gráficas ubicadas a la izquierda corresponden a la tasa de pérdidas del tráfico de voz y las gráficas ubicadas a la derecha corresponden a la tasa de pérdidas del tráfico de datos.

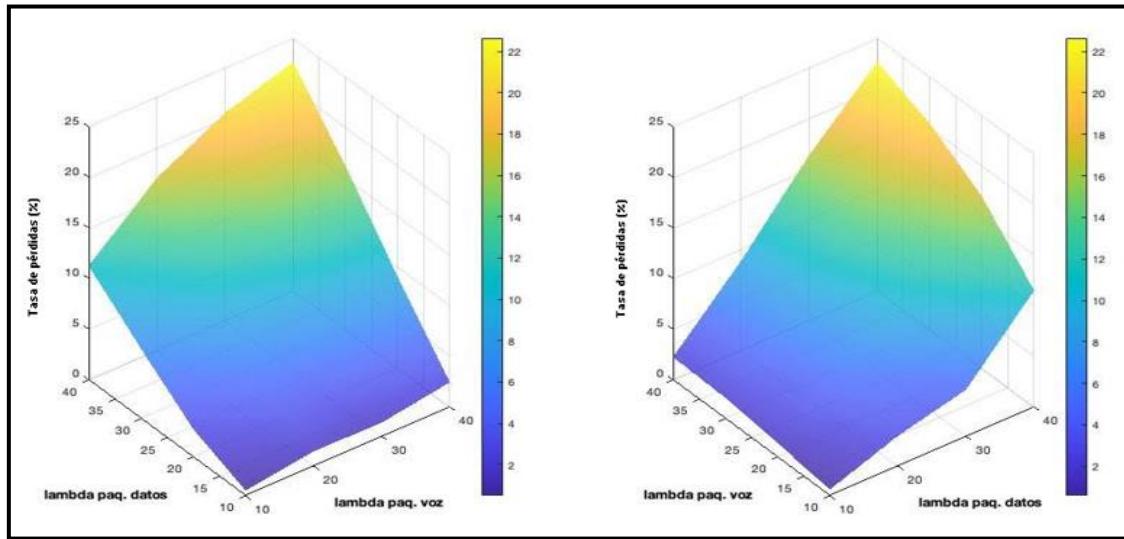


Figura 69. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario sin balanceo de carga con tamaño de la cola de paquetes de datos igual a cero.

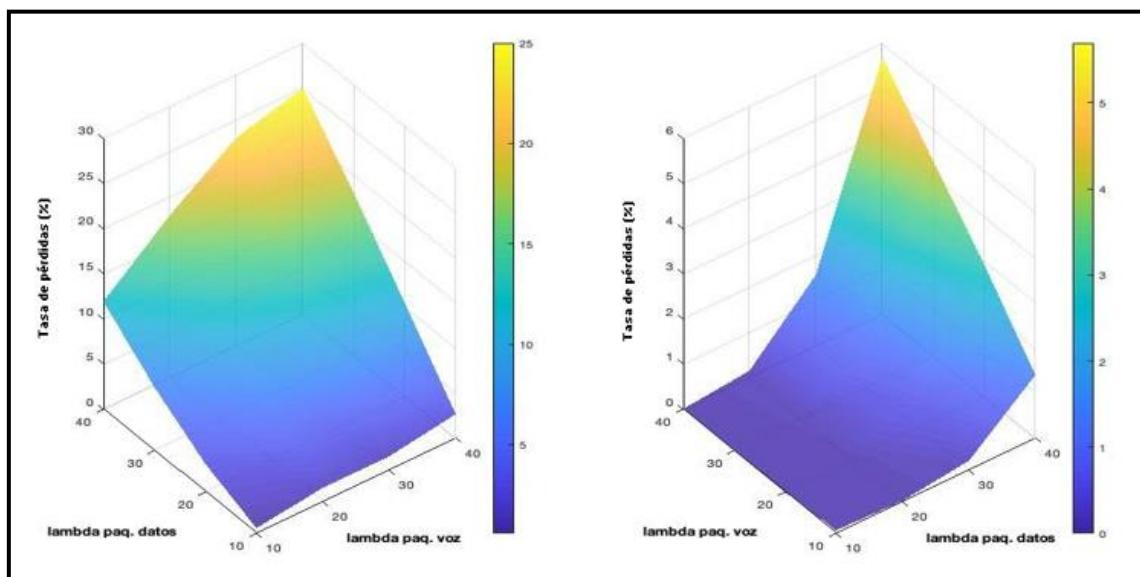


Figura 70. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario sin balanceo de carga con tamaño de la cola de paquetes de datos igual a cinco.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

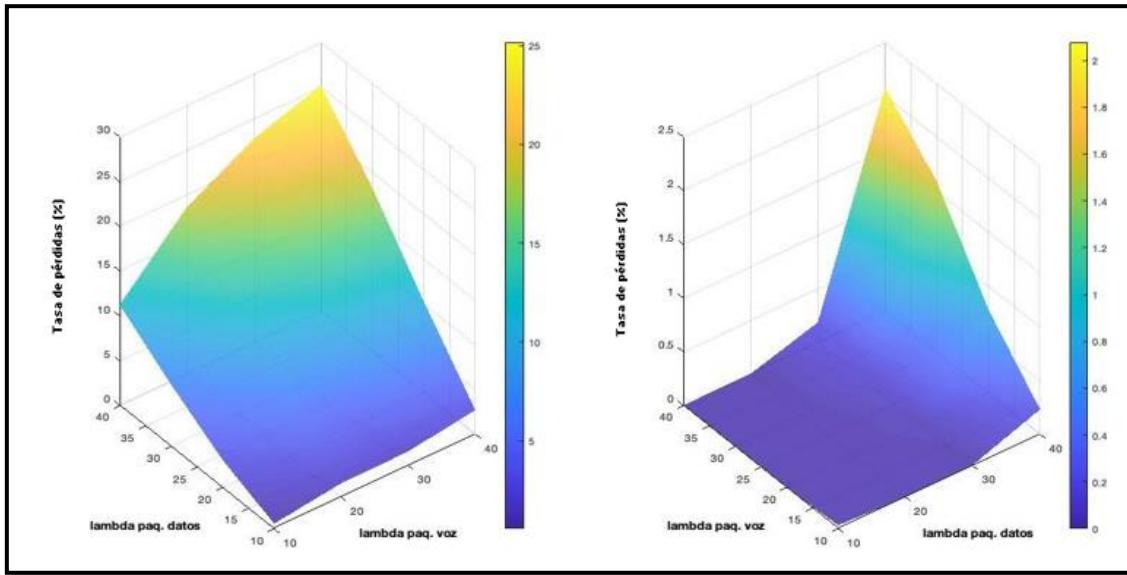


Figura 71. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario sin balanceo de carga con tamaño de la cola de paquetes de datos igual a diez.

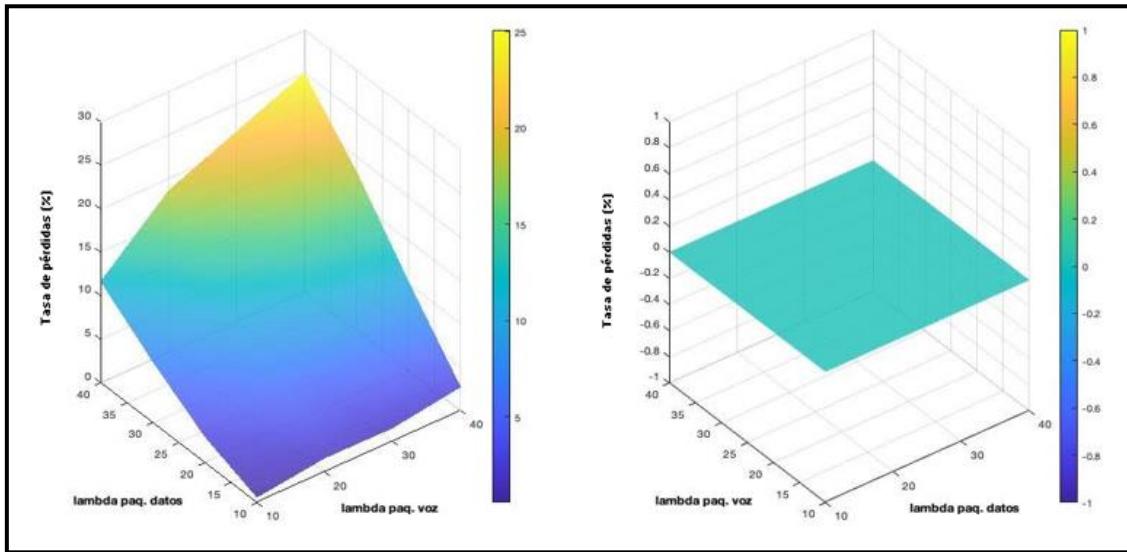


Figura 72. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario sin balanceo de carga con tamaño de la cola de paquetes de datos sin límite.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

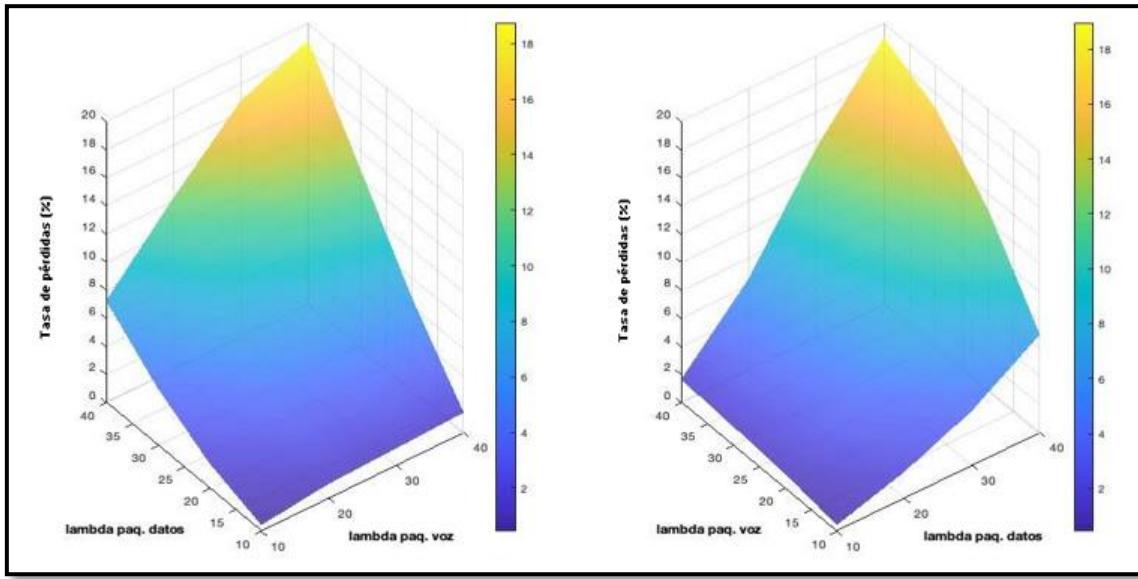


Figura 73. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario de balanceo de primer orden con tamaño de la cola de paquetes de datos igual a cero.

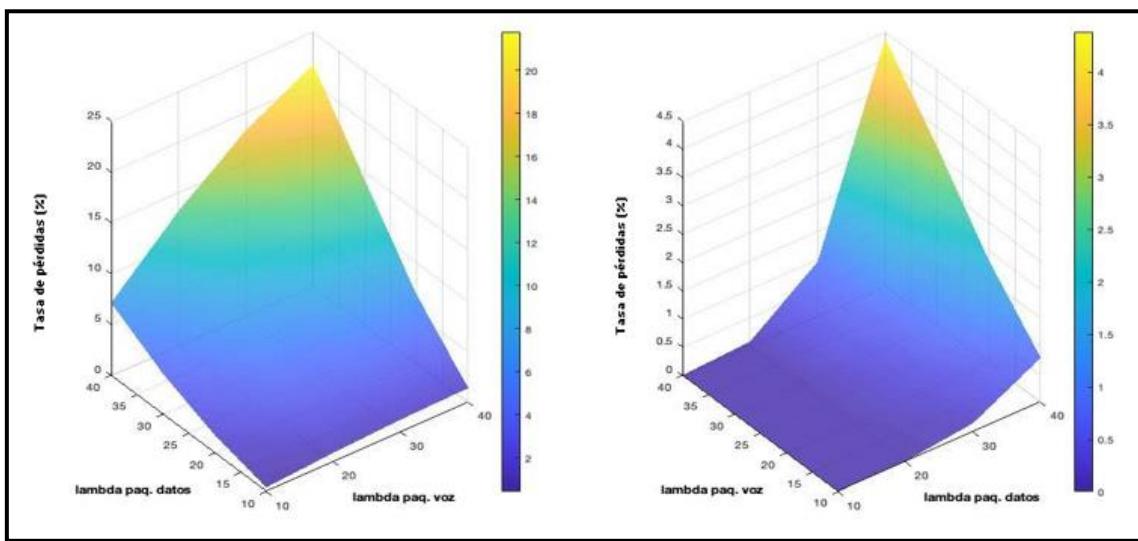


Figura 74. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario de balanceo de primer orden con tamaño de la cola de paquetes de datos igual a cinco.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

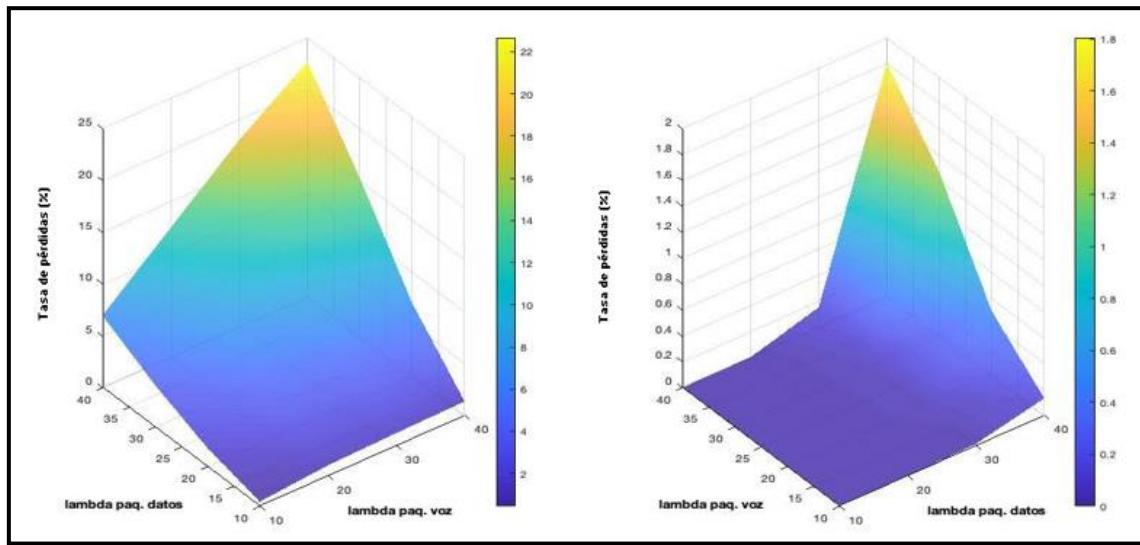


Figura 75. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario de balanceo de primer orden con tamaño de la cola de paquetes de datos igual a diez.

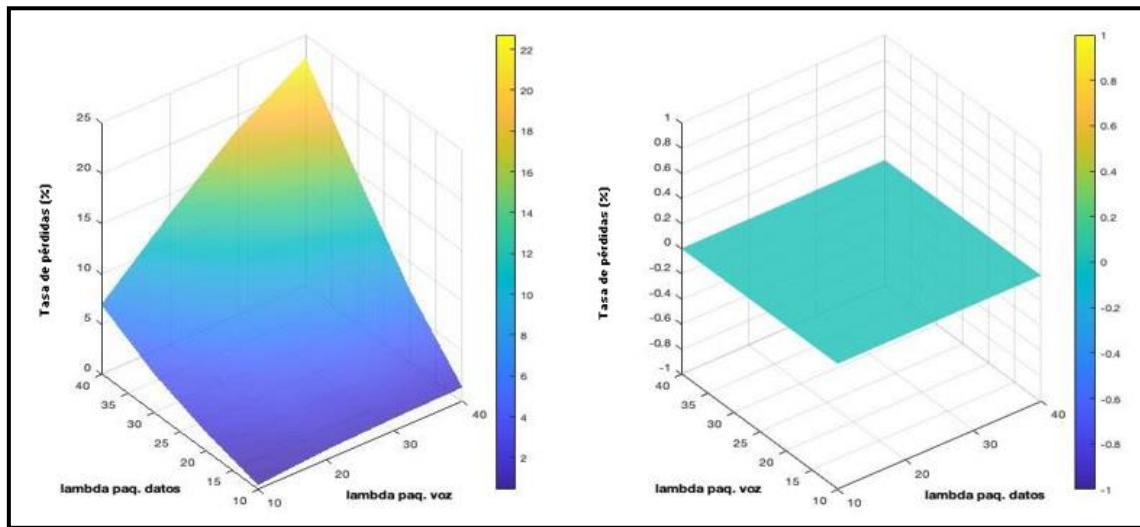


Figura 76. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario de balanceo de primer orden con tamaño de la cola de paquetes de datos sin límite.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

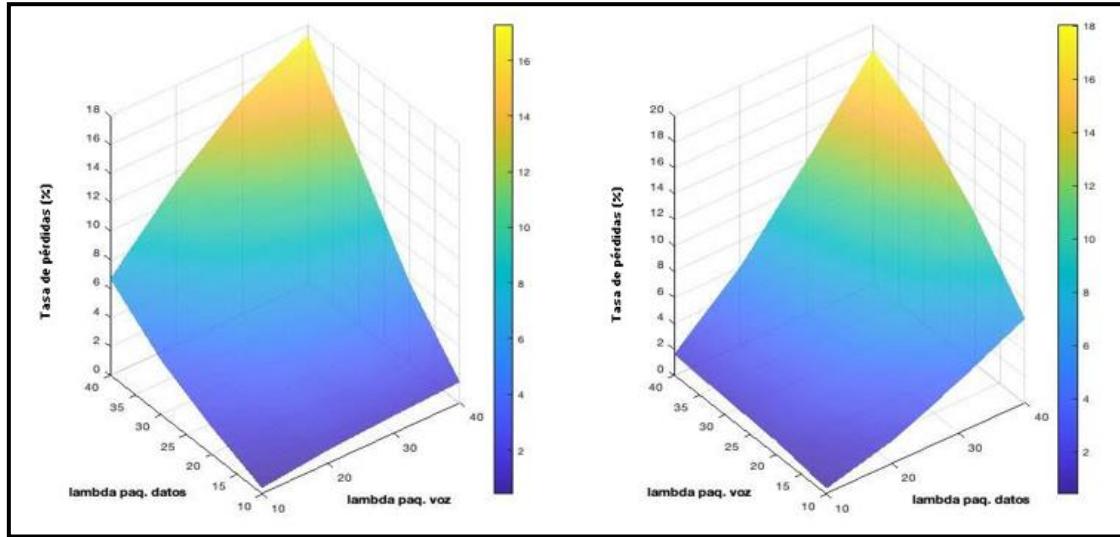


Figura 77. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario de balanceo de segundo orden con tamaño de la cola de paquetes de datos igual a cero.

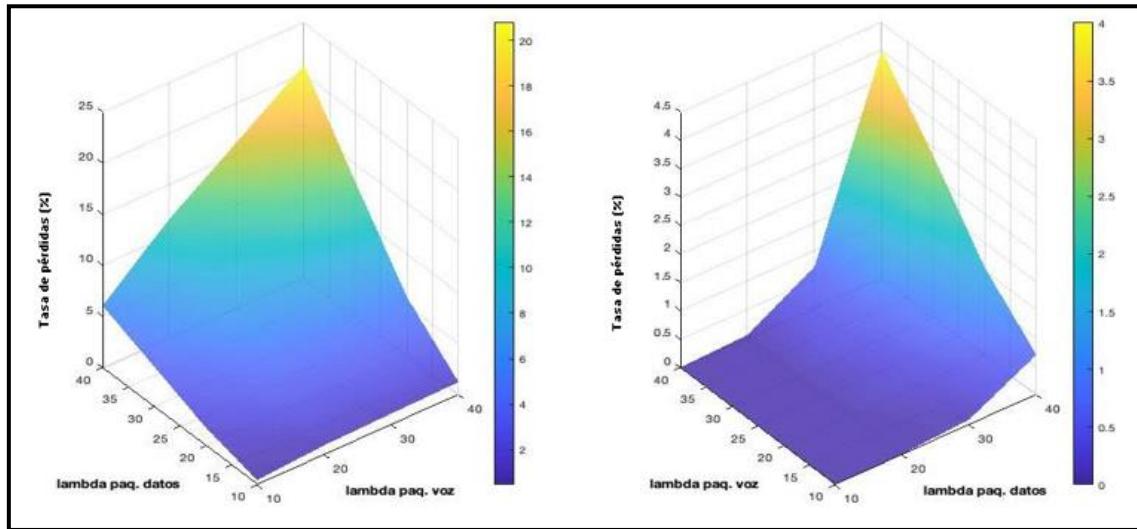


Figura 78. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario de balanceo de segundo orden con tamaño de la cola de paquetes de datos igual a cinco.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

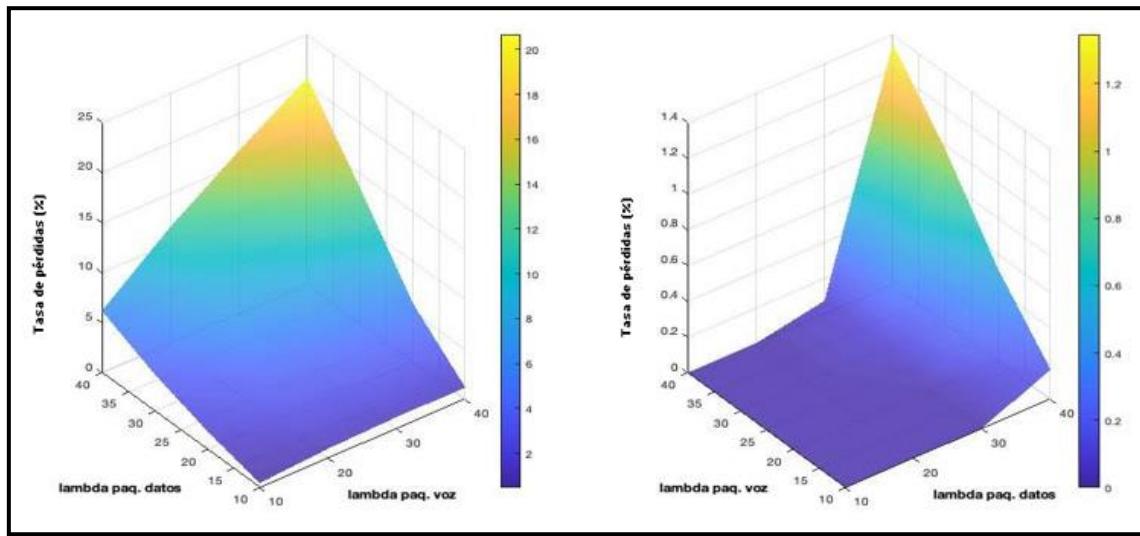


Figura 79. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario de balanceo de segundo orden con tamaño de la cola de paquetes de datos igual a diez.

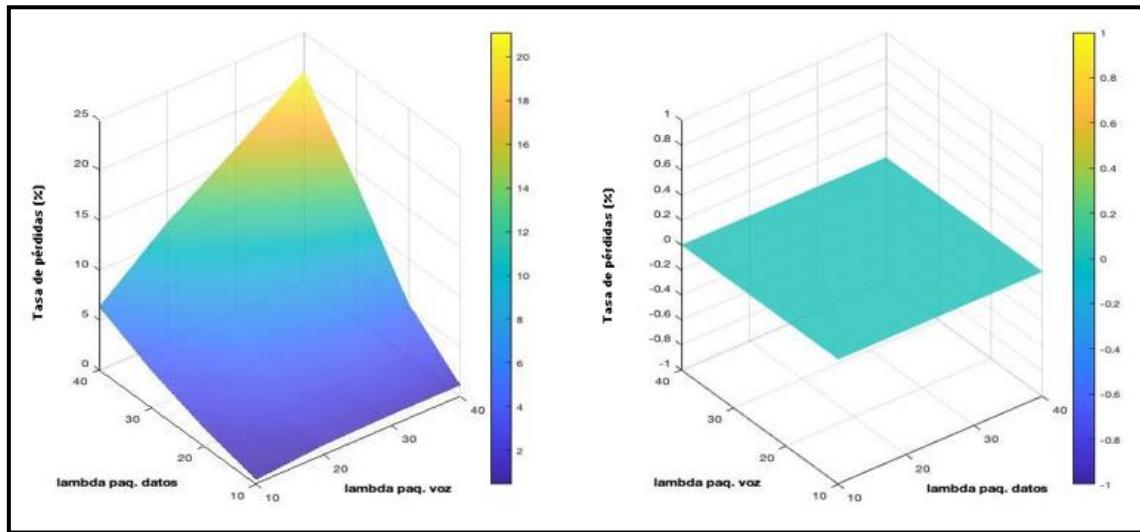


Figura 80. Tasa de pérdidas para paquetes de voz y datos respectivamente, bajo un escenario de balanceo de segundo orden con tamaño de la cola de paquetes de datos sin límite.

Al igual que en el caso de análisis de la cantidad de handovers, la tasa de pérdidas de paquetes también depende esencialmente de la carga de las celdas, el tamaño de la cola para tráfico de datos y el tipo de balanceo empleado. Es sencillo notar de las anteriores gráficas lo siguiente:

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

- A medida que la cola de datos aumenta su tamaño, la tasa de pérdidas del tráfico de datos disminuye hasta el punto donde no se presentan pérdidas (cola infinita).
- En general para los 2 tipos de tráfico se evidencia que la pérdida de paquetes es menor cuando se aplica balanceo de carga de segundo orden, lo cual es coherente con el desarrollo de la propuesta ya que incluye la funcionalidad de balanceo de primer orden más la funcionalidad del balanceo de segundo orden.

De la figura 81 a la figura 83 se puede evidenciar como varía el throughput promedio por celda con respecto a la variación de la tasa de llegada de paquetes de voz y datos (λ_1, λ_2).

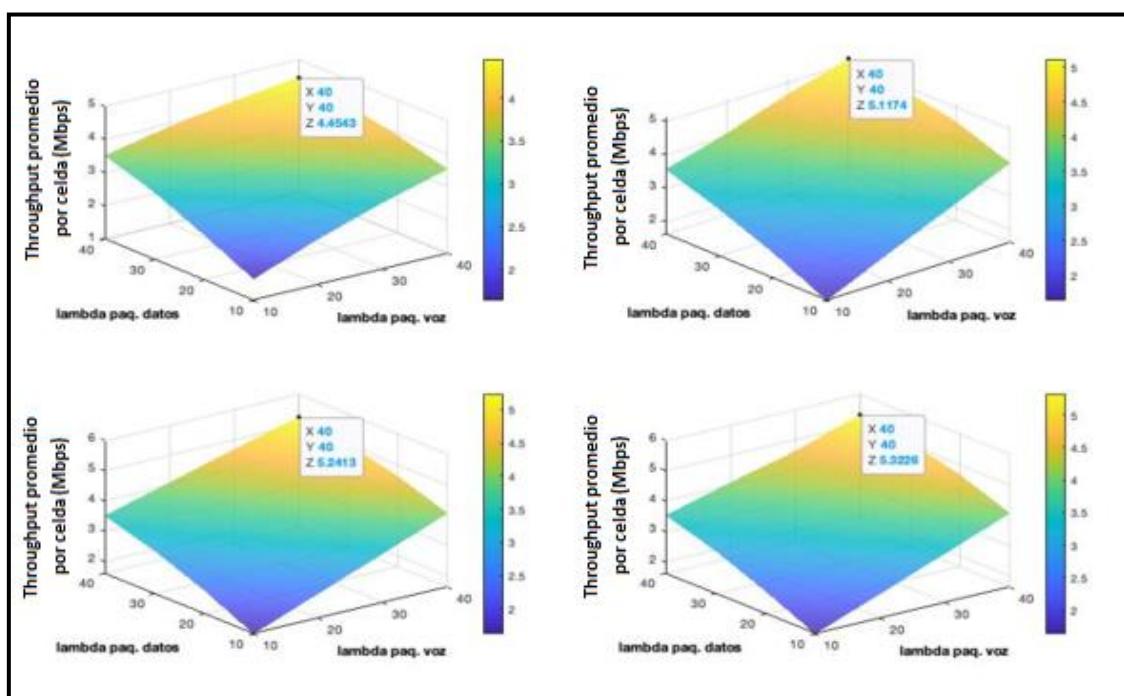


Figura 81. Throughput promedio por celda bajo un escenario sin balanceo de carga con tamaño de la cola de paquetes de datos igual a cero, cinco, diez y sin límite.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

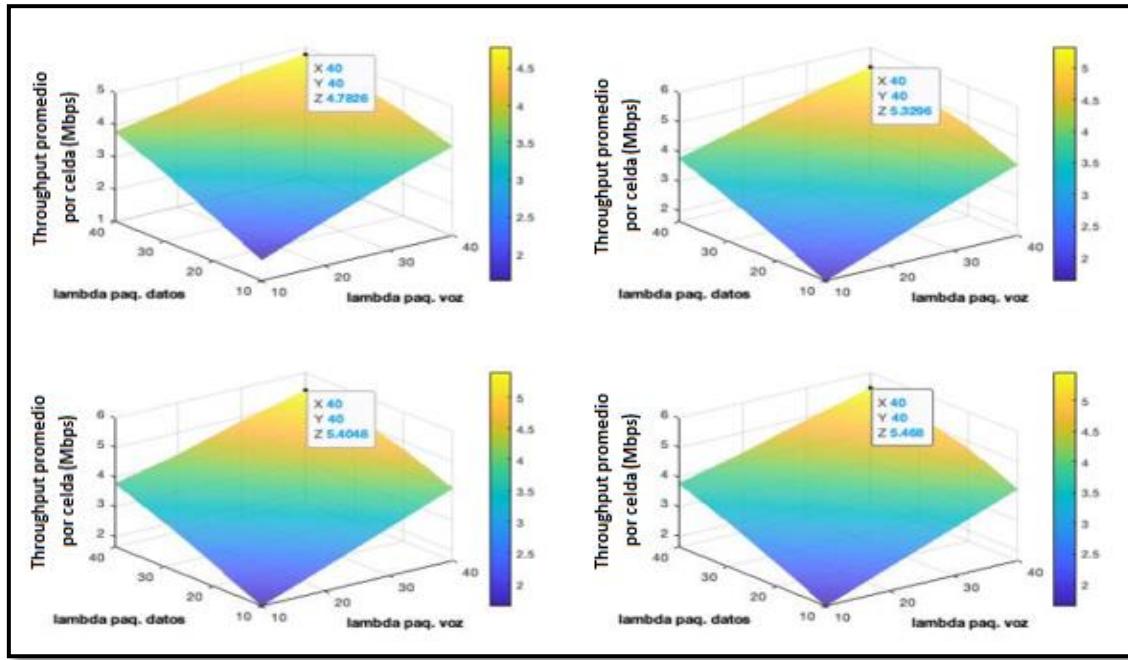


Figura 82. Throughput promedio por celda bajo un escenario con balanceo de carga de primer orden con tamaño de la cola de paquetes de datos igual a cero, cinco, diez y sin límite.

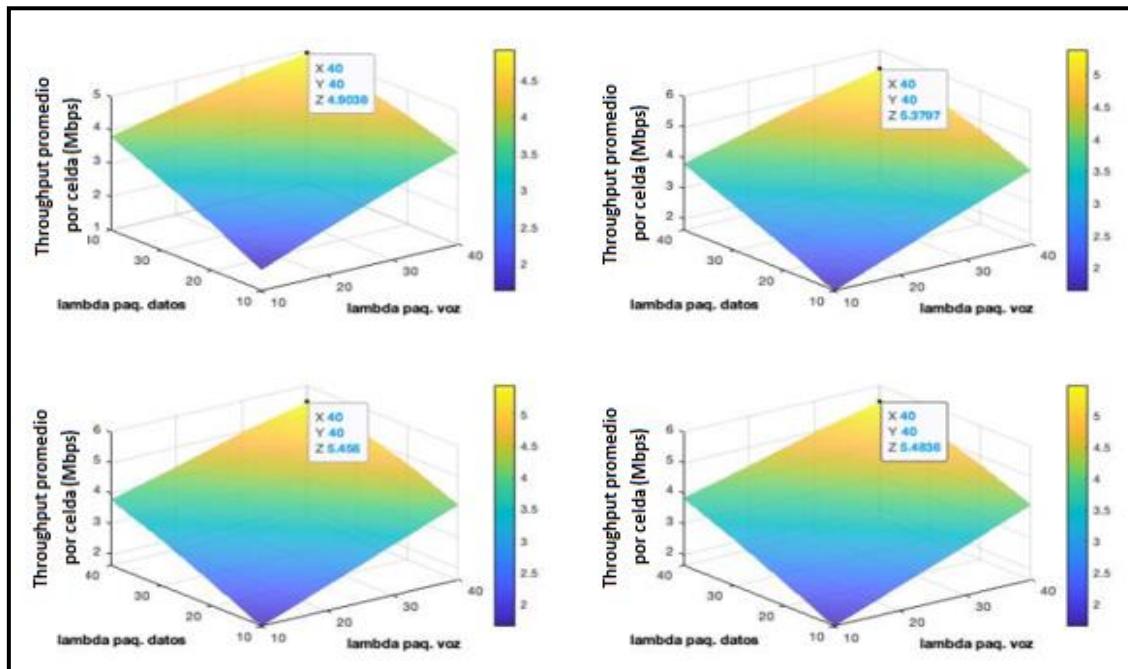


Figura 83. Throughput promedio por celda bajo un escenario con balanceo de carga de segundo orden con tamaño de la cola de paquetes de datos igual a cero, cinco, diez y sin límite.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Finalmente, con respecto a las gráficas relacionadas con el throughput promedio por celda se puede notar lo siguiente:

- A medida que el tráfico aumenta, el throughput promedio por celda aumenta para los 3 escenarios de balanceo de carga, sin embargo, el throughput promedio es mayor para el escenario del balanceo de segundo orden lo cual es coherente con el desarrollo de la propuesta evidenciado en el capítulo 8 y 9.

Es importante recalcar que el throughput máximo de la simulación (5.4836 Mbps) se aproxima al máximo teórico de un escenario con un ancho de banda de 1.4 MHz (6 Resource Block) cuyo valor es aproximadamente 6Mbps (incluyendo tanto información de usuario como de control).

9.7 Condiciones previas del segundo escenario a simular

Las condiciones serán las mismas que las mencionadas en el subcapítulo 9.3 excepto que los usuarios no se distribuirán uniformemente previo a la simulación, en vez de ello se concentrará mayor cantidad de usuarios en la celda 3 como se puede apreciar en la figura 84, con el fin de causar saturación de recursos y poder evidenciar con mayor claridad la diferencia entre escenarios sin balanceo, con balanceo de primer orden y balanceo de segundo orden.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

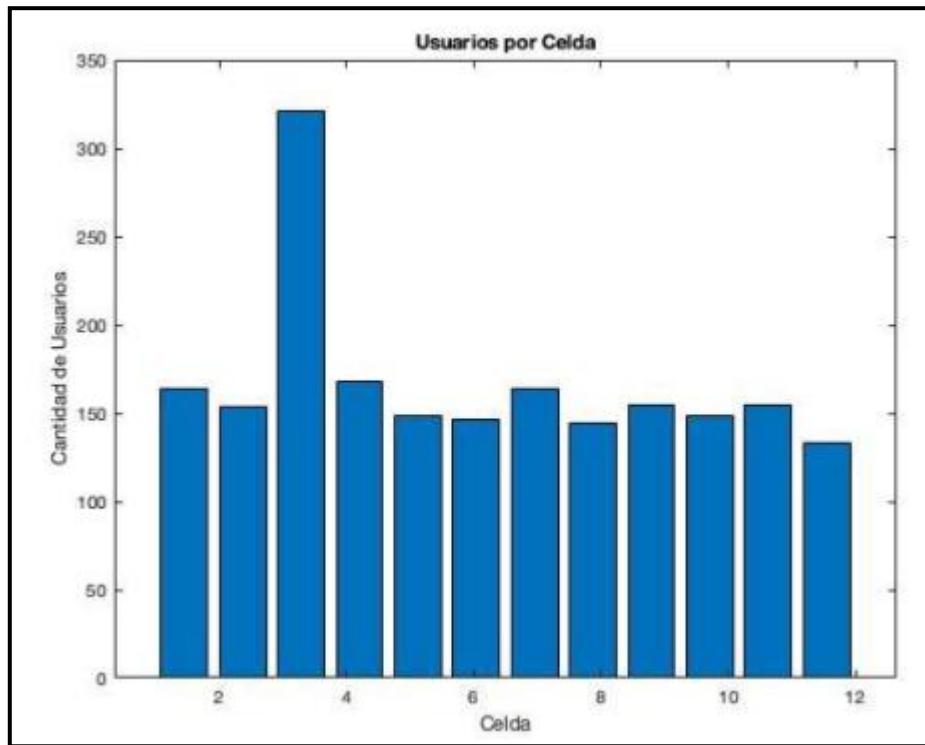


Figura 84. Distribución inicial de usuarios por celda con una sobrecarga de usuarios en la celda 3 de aproximadamente un 100% con respecto al promedio en las demás celdas del sistema.

Para esta simulación se tomarán los siguientes 3 escenarios (Tabla 17):

	λ_1	λ_2	μ_1	μ_2
Escenario 1	25	25	1	1
Escenario 2	35	35	1	1
Escenario 3	45	45	1	1

Tabla 17. Escenarios de simulación con saturación de la celda 3

En las figuras 85, 86 y 87 se observa como varía el throughput total promedio en función de las tasas de llegada para paquetes de voz y datos y del tipo de balanceo empleado.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

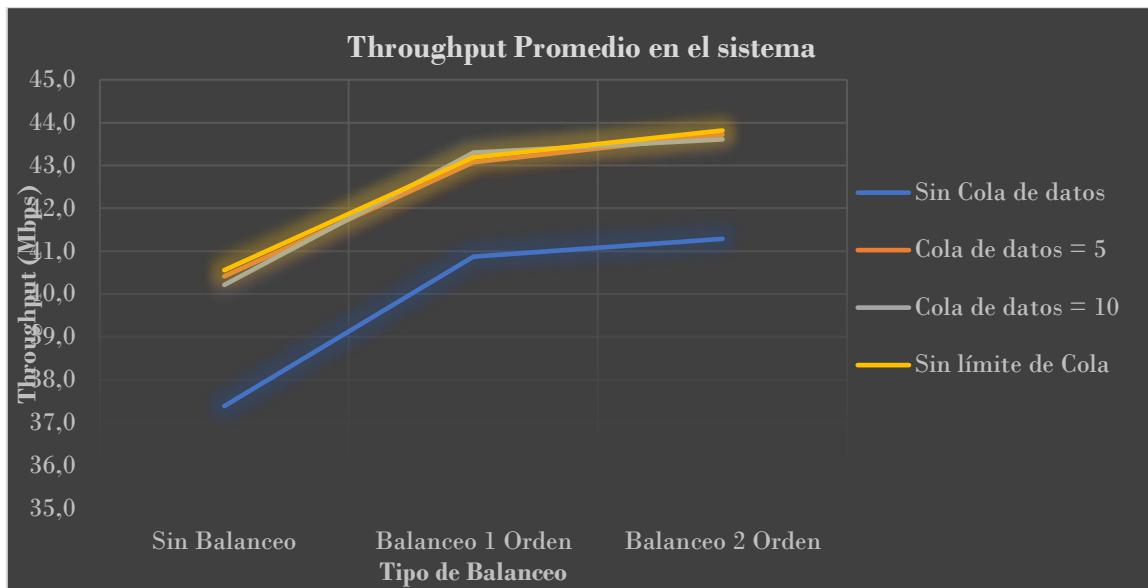


Figura 85. Thoughput promedio en el sistema bajo un escenario con saturación en la celda 3, lambda Voice = 25, lambda Data = 25

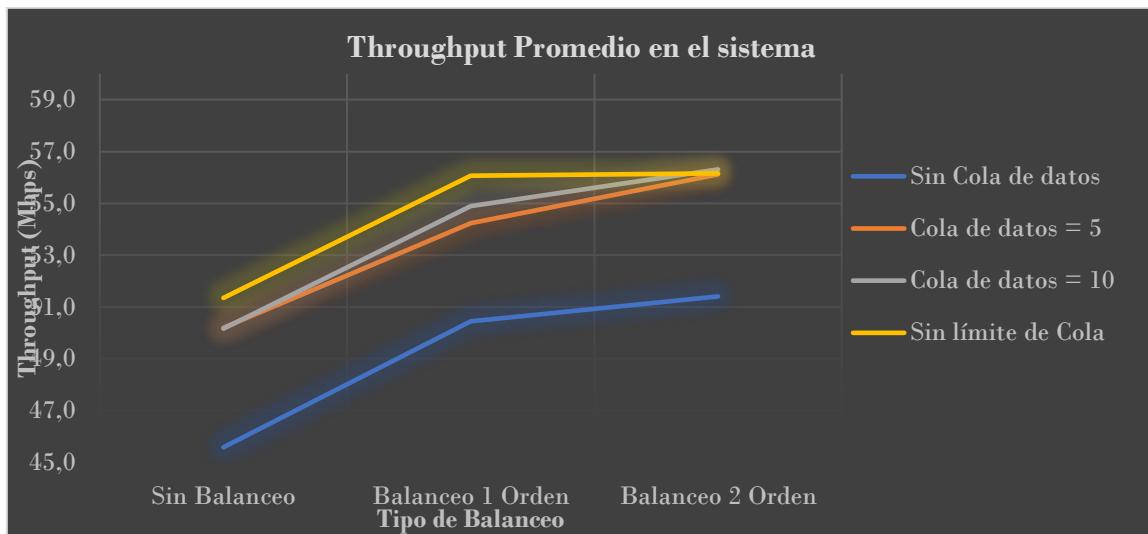


Figura 86. Thoughput promedio en el sistema bajo un escenario con saturación en la celda 3, lambda Voice = 35, lambda Data = 35

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

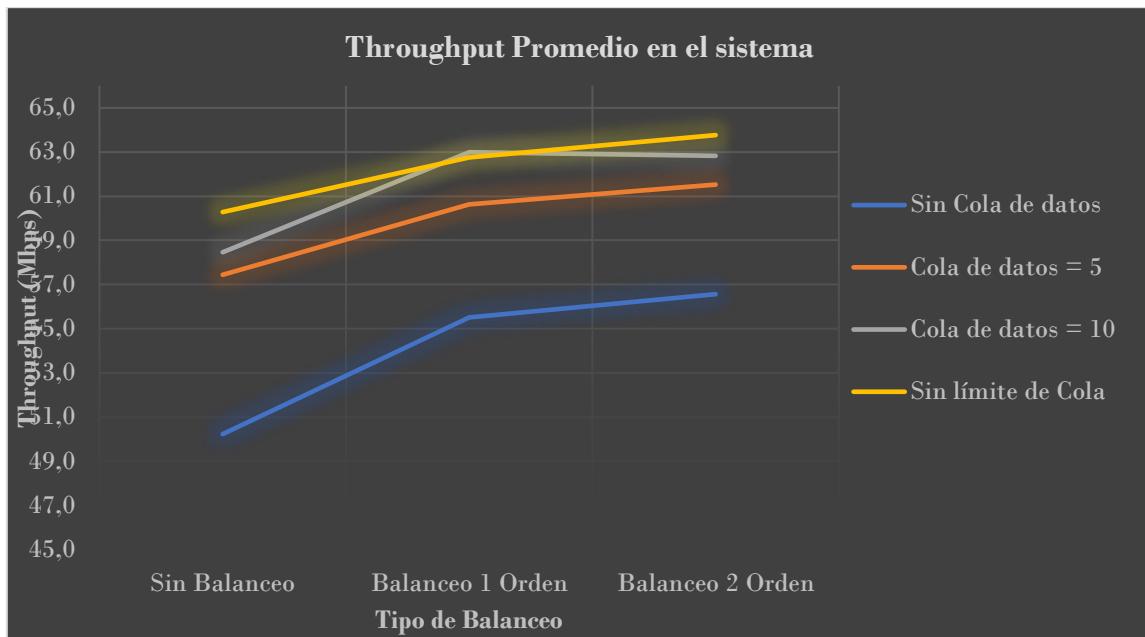


Figura 87. Throughput promedio en el sistema bajo un escenario con saturación en la celda 3, lambda Voice = 45, lambda Data = 45

De las gráficas anteriores se puede concluir lo siguiente:

- Se evidencia un claro aumento en el throughput global de la red cuando se implementa balanceo de carga en comparación al escenario donde no se implementa.
- El escenario con balanceo de carga de segundo orden presenta un mayor throughput global que el escenario de balanceo de carga de primer orden.
- En 45 de 48 escenarios de simulación se presenta mayor throughput cuando la cola (buffer) no tiene límite. Esto es totalmente coherente debido a que, al no descartar ningún paquete de datos estos serán procesados en algún periodo de tiempo por la celda incrementando el throughput global de la red.

En las figuras 88, 89 y 90 se observa como varía la tasa de pérdida de paquetes promedio por celda en función de las tasas de llegada para paquetes de voz y datos y del tipo de balanceo empleado.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

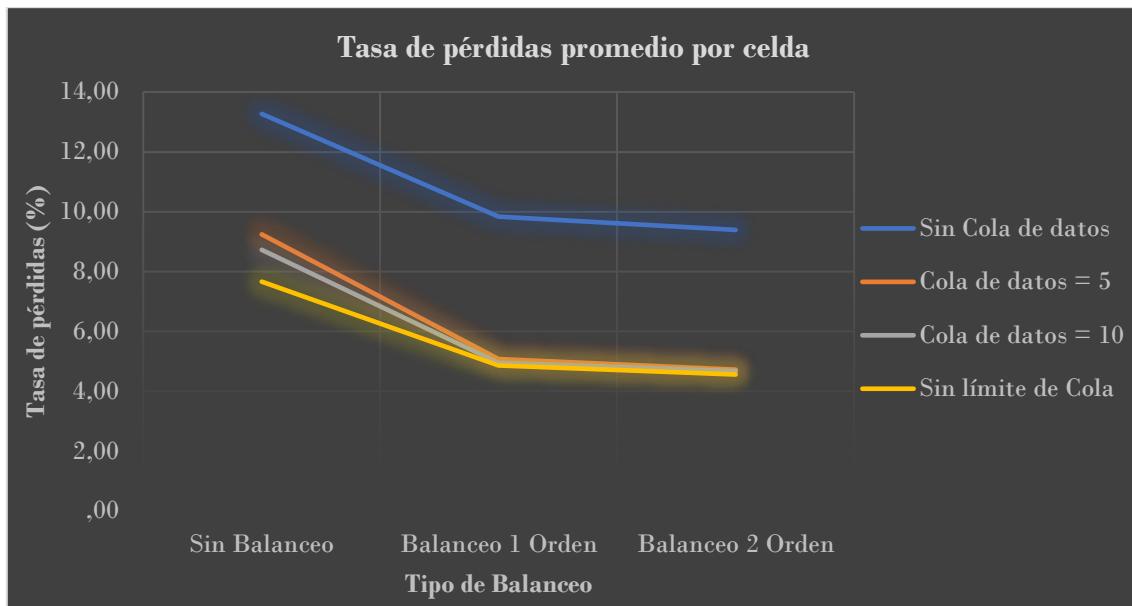


Figura 88. Tasa de pérdidas promedio por celda en el sistema bajo un escenario con saturación en la celda 3, lambda Voice = 25, lambda Data = 25

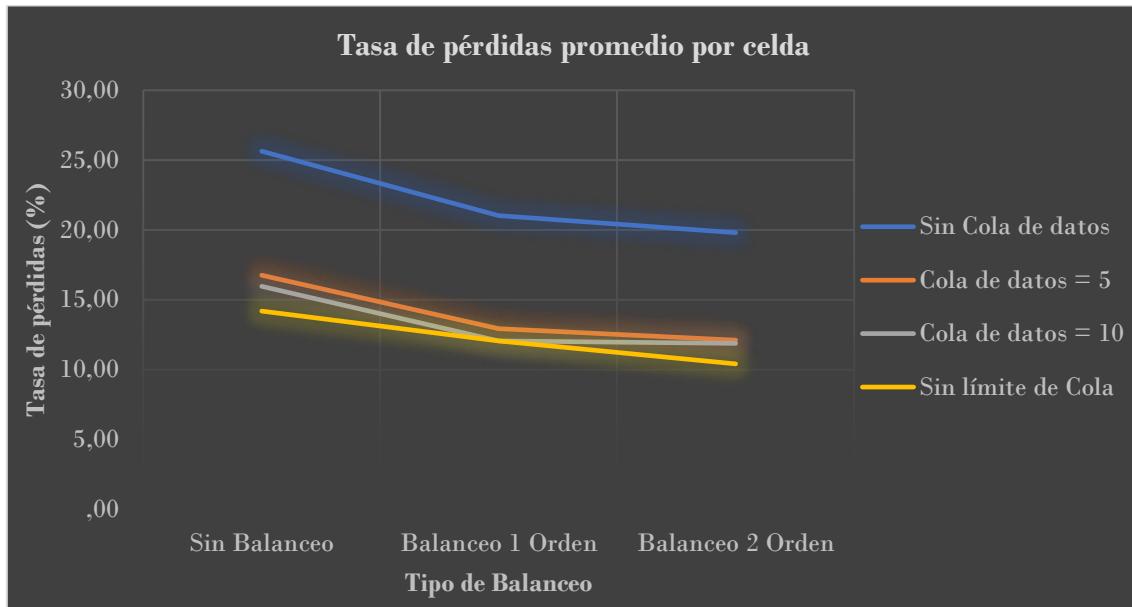


Figura 89. Tasa de pérdidas promedio por celda en el sistema bajo un escenario con saturación en la celda 3, lambda Voice = 35, lambda Data = 35

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

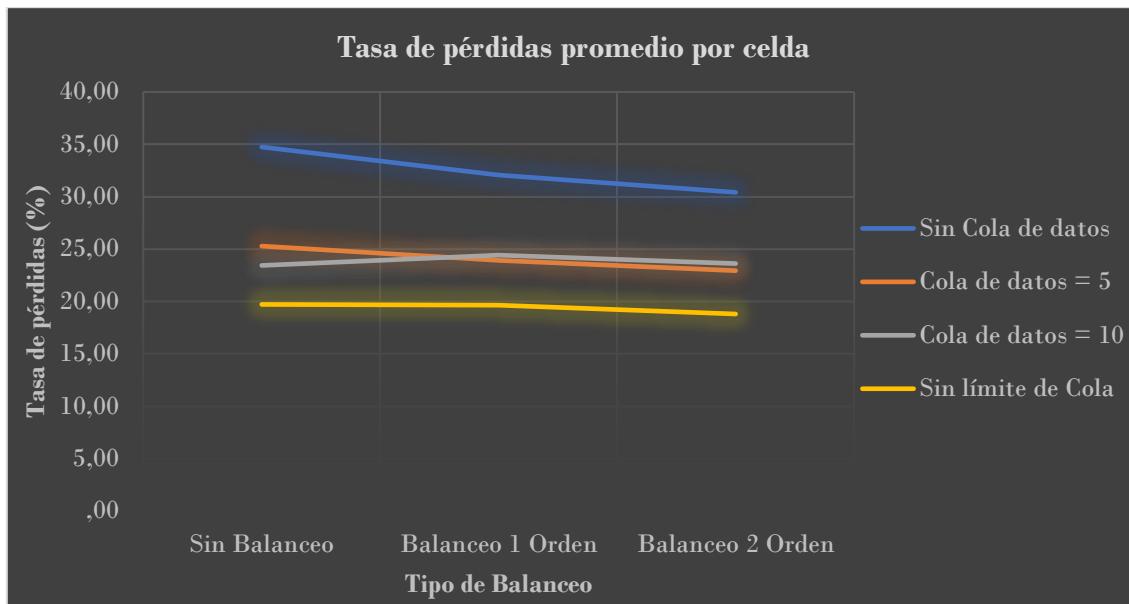


Figura 90. Tasa de pérdidas promedio por celda en el sistema bajo un escenario con saturación en la celda 3, $\lambda_{Voice} = 45$, $\lambda_{Data} = 45$

De las gráficas anteriores se puede concluir lo siguiente:

- Se evidencia una clara disminución en la tasa de pérdidas global de la red cuando se implementa balanceo de carga en comparación al escenario donde no se implementa.
- Las menores de tasas de pérdidas sobre la red se evidencian cuando se implementa el balanceo de segundo orden.
- En 46 de 48 escenarios de simulación se presenta menor tasa de pérdida de paquetes cuando la cola (buffer) no tiene límite. Al igual que en el análisis inmediatamente anterior, es totalmente coherente debido a que, al no descartar ningún paquete de datos la tasa de pérdidas debiese ser menor en comparación a escenarios con tamaño de cola finito.

10. Conclusiones

Mediante el desarrollo de la propuesta expuesta en el presente trabajo se puede concluir que la aplicación de balanceo de carga de segundo orden ofrece mejor rendimiento en términos de pérdida de paquetes (%) y de Throughput en la red (Mbps) en comparación con escenarios de balanceo de primer orden y sin balanceo de carga.

En primer lugar, se obtiene un modelo matemático que permite determinar aproximadamente la tasa de pérdida de paquetes para tráfico de voz en un escenario sin balanceo de carga y con balanceo de carga entre 2 celdas. Para condiciones iniciales iguales a $\lambda_1 = 2$, $\lambda_2 = 2$, $\mu_1 = 1$, $\mu_2 = 1$, el valor teórico de bloqueo de paquetes de voz para un escenario sin balanceo de carga es del 33.3 % y el valor teórico de bloqueo de paquetes de voz para un escenario con balanceo de carga es del 20%, con lo cual se concluye que bajo un escenario de balanceo de carga entre 2 celdas se obtiene una reducción del 13% en la tasa de pérdida de paquetes de voz. De igual manera se realiza el mismo ejercicio para escenarios con condiciones iniciales iguales a $\lambda_1 = 2$, $\lambda_2 = 2$, $\mu_1 = 3$, $\mu_2 = 4$ y $\lambda_1 = 1$, $\lambda_2 = 2$, $\mu_1 = 3$, $\mu_2 = 3$, donde se obtiene una reducción del 12% y 14% respectivamente cuando se emplea balanceo de carga.

Adicional a ello, se obtiene un modelo que permite determinar el número promedio de paquetes en un sistema de una celda en estado estacionario, y las condiciones para que el sistema sea estable. Este modelo permitió evidenciar la validez del simulador empleado en los escenarios descritos en el capítulo 9 mediante una comparación de los resultados teóricos y los resultados arrojados por el simulador. Con respecto a los resultados del número promedio de paquetes en el sistema descrito en el subcapítulo 8.6.3 el simulador presenta un margen de error entre el 1% y el 5% en comparación con los valores teóricos (tabla 11), y con respecto a los resultados del bloqueo de paquetes de voz para escenarios con y sin balanceo el simulador presenta un margen de error entre el 1% y el 3% en comparación con los valores teóricos (tabla 12 y tabla 13). Dependiendo del escenario, la cantidad de simulaciones varía hasta lograr el nivel de confianza requerido como se muestra en la tabla 11.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Posteriormente se realiza la simulación sobre un escenario de 12 celdas para comparar los 3 tipos de balanceo (sin balanceo, balanceo de primer orden y balanceo de segundo orden) bajo la topología evidenciada en la figura 38, con una distribución uniforme de usuarios sobre todas las celdas como se observa en la figura 51, con tasas de servicios constantes iguales a 1, variaciones uniformes de las tasas de llegada para el tráfico de voz y datos respectivamente y variaciones del tamaño de la cola para los paquetes de datos. Para el escenario de mayor congestión ($\lambda_1 = 40$, $\lambda_2 = 40$) y sin cola el tráfico de datos se obtiene que la tasa de pérdidas es del 45% cuando no se aplica balanceo de carga, 37% cuando se aplica balanceo de primer orden y 35% cuando se aplica balanceo de carga de segundo orden. A medida que el tamaño de la cola aumenta la tasa de pérdidas disminuye, por ejemplo cuando la cola se hace infinita la tasa de pérdidas es del 25% cuando no se aplica balanceo de carga, 22.7% cuando se aplica balanceo de carga de primer orden y 21% cuando se aplica balanceo de segundo orden. Con respecto al throughput se observa que para un escenario de cola infinita y sin balanceo de carga el valor promedio por celda es 5.32 Mbps, con balanceo de carga de primer orden es 5.46 Mbps y con balanceo de carga de segundo orden es 5.48 Mbps.

Finalmente, para la última simulación realizada se tomaron las condiciones de la simulación anterior a excepción de que la distribución de usuarios ya no será uniforme sobre todas las celdas. En su lugar se congestionó una de las celdas con mayor cantidad de usuarios como se observa en la figura 84, para evidenciar con mayor claridad el efecto de aplicar o no balanceo de carga en las celdas congestionadas. Para el escenario de $\lambda_1 = 25$, $\lambda_2 = 25$ la tasa de pérdidas máxima es 14% y el throughput global en la red es 40.5 Mbps cuando no se aplica balanceo de carga, la tasa de pérdidas máxima es 10% y el throughput global en la red es 43.1 Mbps cuando se aplica balanceo de carga de primer orden y la tasa de pérdidas máxima es 9% y el throughput global en la red es 43.8 Mbps cuando se aplica balanceo de carga de segundo orden. Para el escenario de $\lambda_1 = 35$, $\lambda_2 = 35$ la tasa de pérdidas máxima es 25% y el throughput global en la red es 51.3 Mbps cuando no se aplica balanceo de carga, la tasa de pérdidas máxima es 21% y el throughput global en la red es 56 Mbps cuando se aplica balanceo de carga de primer orden y la tasa de pérdidas máxima es 19% y el throughput global en la red es 56.3 Mbps cuando se aplica balanceo de carga de segundo orden. Para el escenario de $\lambda_1 = 45$, $\lambda_2 = 45$ la tasa de pérdidas máxima es 34% y el throughput global en la red es 60.2 Mbps cuando no se aplica balanceo de carga, la tasa de pérdidas máxima es 32% y el throughput global en la red es 62.9 Mbps cuando se aplica

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

balanceo de carga de primer orden y la tasa de pérdidas máxima es 30% y el throughput global en la red es 63.7 Mbps cuando se aplica balanceo de carga de segundo orden.

Es claro ver en los resultados mencionados que en los escenarios donde se aplica balanceo de segundo orden la tasa de pérdidas es menor y el throughput global en la red y promedio por celda es mayor en comparación con los escenarios donde se aplica balanceo de carga de primer orden o donde no se aplica balanceo de carga.

De la misma manera es importante mencionar que con la implementación del balanceo de segundo orden el número de handovers se incrementa en aproximadamente un 33% con respecto al escenario de primer orden para condiciones de alta congestión ($\lambda_1 = 40$, $\lambda_2 = 40$). Esto trae consigo efectos negativos debido a que el aumento en el número de handovers incrementa el tráfico del plano de control a nivel de RAN (Radio Access Network) y sobre la interfaz S1. Como trabajo futuro sería pertinente incluir un análisis puntual sobre la utilización del canal para tráfico de señalización debido al balanceo de segundo orden, sustentado por medio de un modelo probabilístico y a nivel de simulación.

11. Apéndices

En el presente capítulo se expondrán los apéndices que respaldan algunos desarrollos y resultados que se expusieron en los capítulos 8 y 9.

A. Desarrollo Matemático para la Obtención de la Función Generadora de Probabilidad

Con el fin de determinar la Función Generadora de Probabilidad se determinarán a continuación las ecuaciones de Balanceo Detallado obtenidas de la Cadena de Markov mostrada en la figura 42. La base de este desarrollo se tomó de [3].

$$(\lambda_1 + \lambda_2) * P_{0,0} = \mu_1 * p_{1,0} + \mu_2 * P_{0,1} \quad (43)$$

$$(\lambda_2 + \mu_1) * P_{1,0} = \lambda_1 * P_{0,0} \quad (44)$$

$$(\lambda_1 + \lambda_2 + \mu_2) * P_{0,i} = \lambda_2 * P_{0,i-1} + \mu_1 * P_{1,i} + \mu_2 * P_{0,i+1} \quad (45)$$

$$(\lambda_2 + \mu_1) * P_{1,i} = \lambda_2 * P_{1,i-1} + \lambda_1 * P_{0,i} \quad (46)$$

Con el fin de simplificar los términos constantes se agruparán de la siguiente manera:

$$\rho_1 = \frac{\lambda_1}{\mu_1}, \quad \rho_2 = \frac{\lambda_2}{\mu_2}, \quad \beta = \frac{\mu_2}{\mu_1}, \quad \beta * \rho_2 = \frac{\lambda_2}{\mu_2} * \frac{\mu_2}{\mu_1} = \frac{\lambda_2}{\mu_1}$$

Así las ecuaciones 43, 44, 45 y 46 expuestas anteriormente quedan representadas de la siguiente forma:

$$(\rho_1 + \beta * \rho_2) * P_{0,0} = P_{1,0} + \beta * P_{0,1} \quad (47)$$

$$(1 + \beta * \rho_2) * P_{1,0} = \rho_1 * P_{0,0} \quad (48)$$

$$(\rho_1 + \beta * \rho_2 + \beta) * P_{0,i} = \beta * \rho_2 * P_{0,i-1} + P_{1,i} + \beta * P_{0,i+1} \quad (49)$$

$$(1 + \beta * \rho_2) * P_{1,i} = \beta * \rho_2 * P_{1,i-1} + \rho_1 * P_{0,i} \quad (50)$$

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

De la ecuación 48 se puede expresar $P_{1,0}$ en términos de $P_{0,0}$:

$$P_{1,0} = \frac{\rho_1 * P_{0,0}}{1 + \beta * \rho_2} \quad (51)$$

De la ecuación 47 y 51 se puede expresar $P_{0,1}$ en términos de $P_{0,0}$:

$$\begin{aligned} (\rho_1 + \beta * \rho_2) * P_{0,0} &= \frac{\rho_1 * P_{0,0}}{1 + \beta * \rho_2} + \beta * P_{0,1} \\ P_{0,1} &= \rho_2 * P_{0,0} * \left(1 + \frac{\rho_1}{1 + \beta * \rho_2}\right) \end{aligned} \quad (52)$$

Las ecuaciones de recurrencia 49 y 50 se multiplicarán multiplicando por z^i :

$$\begin{aligned} (\rho_1 + \beta * \rho_2 + \beta) * P_{0,i} * z^i &= \beta * \rho_2 * P_{0,i-1} * z^i + P_{1,i} * z^i + \beta * P_{0,i+1} * z^i \\ (1 + \beta * \rho_2) * P_{1,i} * z^i &= \beta * \rho_2 * P_{1,i-1} * z^i + \rho_1 * P_{0,i} * z^i \end{aligned}$$

Como estas ecuaciones son válidas para todo $i \geq 1$, se puede hacer la suma desde i igual a 1 hasta infinito de la siguiente forma:

$$(\rho_1 + \beta * \rho_2 + \beta) * \sum_{i=1}^{\infty} P_{0,i} * z^i = \beta * \rho_2 * \sum_{i=1}^{\infty} P_{0,i-1} * z^i + \sum_{i=0}^{\infty} P_{1,i} * z^i + \beta * \sum_{i=0}^{\infty} P_{0,i+1} * z^i \quad (53)$$

$$(1 + \beta * \rho_2) * \sum_{i=1}^{\infty} P_{1,i} * z^i = \beta * \rho_2 * \sum_{i=1}^{\infty} P_{1,i-1} * z^i + \rho_1 * \sum_{i=1}^{\infty} P_{0,i} * z^i \quad (54)$$

Teniendo la ecuación 53 en términos de z se pueden expresar en términos de $G_0(z)$ y $G_1(z)$ usando las ecuaciones 14 y 15:

$$\begin{aligned} G_0(z) \left(\rho_1 + \beta * \rho_2 + \beta - \beta * \rho_2 - \frac{\beta}{z} \right) &= P_{0,0} * \left(\rho_1 + \beta * \rho_2 + \beta - \frac{\beta}{z} \right) - \beta * P_{0,1} + G_1(z) - P_{1,0} \\ G_0(z) &= \frac{P_{0,0} * \left(\rho_1 + \beta * \rho_2 + \beta - \frac{\beta}{z} \right) - \beta * P_{0,1} + G_1(z) - P_{1,0}}{\rho_1 + \beta * \rho_2 + \beta - \beta * \rho_2 - \frac{\beta}{z}} \end{aligned} \quad (55)$$

De forma similar la ecuación 54 puede expresarse en términos de $G_0(z)$ y $G_1(z)$:

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

$$G_1(z)(1 + \beta * \rho_2 - \beta * \rho_2 * z) = \rho_1 * (G_0(z) - P_{0,0}) + P_{1,0} * (1 + \beta * \rho_2)$$

$$G_1(z) = \frac{\rho_1 * (G_0(z) - P_{0,0}) + P_{1,0} * (1 + \beta * \rho_2)}{1 + \beta * \rho_2 - \beta * \rho_2 * z} \quad (56)$$

Reemplazando en las ecuaciones 55 y 56 los valores de $P_{0,1}$ y $P_{1,0}$ de las ecuaciones 51 y 52 se obtienen las siguientes ecuaciones:

$$G_0(z) = \frac{\beta * \rho_2 - \beta * \rho_2 * z + 1}{1 - \rho_2 * z + \beta * \rho_2 - \beta * \rho_2^2 * z + \beta * \rho_2^2 * z^2 - \beta * \rho_2 * z - \rho_1 * \rho_2 * z} * P_{0,0} \quad (57)$$

$$G_1(z) = \frac{\rho_1}{1 - \rho_2 * z + \beta * \rho_2 - \beta * \rho_2^2 * z + \beta * \rho_2^2 * z^2 - \beta * \rho_2 * z - \rho_1 * \rho_2 * z} * P_{0,0} \quad (58)$$

De lo que se obtiene la función Generadora de acuerdo con la ecuación 59 como:

$$G(z) = \frac{\rho_1 + \beta * \rho_2 - \beta * \rho_2 * z + 1}{1 - \rho_2 * z + \beta * \rho_2 - \beta * \rho_2^2 * z + \beta * \rho_2^2 * z^2 - \beta * \rho_2 * z - \rho_1 * \rho_2 * z} * P_{0,0} \quad (59)$$

Para hallar el valor de $P_{0,0}$ se debe evaluar $G(1)$ e igualar a 1:

$$P_{0,0} = \frac{1 - \rho_2 * z + \beta * \rho_2 - \beta * \rho_2^2 * z + \beta * \rho_2^2 * z^2 - \beta * \rho_2 * z - \rho_1 * \rho_2 * z}{\rho_1 + \beta * \rho_2 - \beta * \rho_2 * z + 1}$$

Evaluando $z = 1$:

$$P_{0,0} = \frac{1 - \rho_2 - \rho_1 * \rho_2}{1 + \rho_1} \quad (60)$$

Reemplazando el valor de $P_{0,0}$ en la ecuación 59 que caracteriza $G(z)$ se obtiene la ecuación 20, la cual se plasma en el capítulo 8:

$$G(z) = \frac{\rho_1 + \beta * \rho_2 - \beta * \rho_2 * z + 1}{1 - \rho_2 * z + \beta * \rho_2 - \beta * \rho_2^2 * z + \beta * \rho_2^2 * z^2 - \beta * \rho_2 * z - \rho_1 * \rho_2 * z} * \frac{1 - \rho_2 - \rho_1 * \rho_2}{1 + \rho_1} \quad (20)$$

B. Desarrollo Matemático para la Obtención de la Probabilidad de Bloqueo bajo un escenario de Balanceo de Carga

De las siguientes ecuaciones de Balanceo Detallado:

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

$$(\lambda_{v1} + \lambda_{v2}) * P_{0,0} = \mu_{v1} * P_{1,0} + \mu_{v2} * P_{0,1} \quad (61)$$

$$(\lambda_{v1} + \lambda_{v2} + \mu_{v1}) * P_{1,0} = \mu_{v2} * P_{1,1} + \lambda_{v1} * P_{0,0} \quad (62)$$

$$(\lambda_{v1} + \lambda_{v2} + \mu_{v2}) * P_{0,1} = \mu_{v1} * P_{1,1} + \lambda_{v2} * P_{0,0} \quad (63)$$

$$(\mu_{v1} + \mu_{v2}) * P_{1,1} = (\lambda_{v1} + \lambda_{v2}) * P_{1,0} + (\lambda_{v1} + \lambda_{v2}) * P_{0,1} \quad (64)$$

$$P_{0,0} + P_{0,1} + P_{1,0} + P_{1,1} = 1 \quad (65)$$

Mediante el uso del software MATLAB se obtiene la solución del sistema expuesto a continuación:

$$P_{0,0} = \frac{\mu_{v1}^2 * \mu_{v2} + \mu_{v2}^2 * \mu_{v1} + 2 * \lambda_{v1} * \mu_{v1} * \mu_{v2} + 2 * \lambda_{v2} * \mu_{v1} * \mu_{v2}}{\gamma}$$

$$P_{0,1} = \frac{\lambda_{v1}^2 * \mu_{v1} + 2 * \lambda_{v1} * \lambda_{v2} * \mu_{v1} + \lambda_{v2}^2 * \mu_{v1} + \lambda_{v2} * \mu_{v1} * \mu_{v2}}{\gamma}$$

$$P_{1,0} = \frac{\lambda_{v1}^2 * \mu_{v2} + 2 * \lambda_{v1} * \lambda_{v2} * \mu_{v2} + \lambda_{v1} * \mu_{v2}^2 + \lambda_{v1} * \mu_{v1} * \mu_{v2} + \lambda_{v2}^2 * \mu_{v2}}{\gamma}$$

$$P_{1,1} = \frac{2 * \lambda_{v1} * \lambda_{v2} * \mu_{v1} * \mu_{v2}}{\gamma}$$

Así la probabilidad de Bloqueo en el sistema ($P_{1,1}$) será:

$$P_{Bloqueo} = \frac{3 * \lambda_{v1}^2 * \lambda_{v2} + 3 * \lambda_{v2}^2 * \lambda_{v1} + \lambda_{v1}^2 * \mu_{v2} + \lambda_{v2}^2 * \mu_{v1} + \lambda_{v1}^3 + \lambda_{v2}^3 + \lambda_{v1} * \lambda_{v2} * \mu_{v1} + \lambda_{v1} * \lambda_{v2} * \mu_{v2}}{\gamma} \quad (31)$$

Donde γ es igual a:

$$\begin{aligned} \gamma = & \lambda_{v1}^3 + 3\lambda_{v1}^2\lambda_{v2} + \lambda_{v1}^2\mu_{v1} + 2\lambda_{v1}^2\mu_{v2} + 3\lambda_{v2}^2\lambda_{v1} + 3\lambda_{v1}\lambda_{v2}\mu_{v1} + 3\lambda_{v1}\lambda_{v2}\mu_{v2} + 3\lambda_{v1}\mu_{v1}\mu_{v2} + \lambda_{v1}\mu_{v2}^2 + \lambda_{v2}^3 + 2\lambda_{v2}^2\mu_{v1} \\ & + \lambda_{v2}^2\mu_{v2} + \lambda_{v2}\mu_{v1}^2 + 3\lambda_{v2}\mu_{v1}\mu_{v2} + \mu_{v1}^2\mu_{v2} + \mu_{v2}^2\mu_{v1} \end{aligned} \quad (32)$$

C. Código de Simulación para el escenario acotado empleado en el modelo matemático más el ejercicio del cálculo de la cantidad de Simulaciones (Nivel de Confianza).

A continuación se expone el código de simulación para el escenario que se estudió en el capítulo 8.6 [53]:

```

function [trace, packetsAverage, totalPackets, lostPackets] = testTesis2(lambda1, lambda2,
mu1, mu2, N1, N2, SimulationTime, balanceTrigger)
    Servers = 2;
    N = N1+N2; % 1 transmission channel
    futureEvents = inf*ones(Servers*N+2,1); %1,2 arrivals, i+2: output by channel i.
i=1:N
    futureEvents(1) = random('exp',1/(Servers*lambda1)); % Schedule next arrivals
    futureEvents(2) = random('exp',1/(Servers*lambda2));
    systemPackets = zeros(2*Servers,1); % On Service Packets + Queued Packets
    totalPackets = zeros(2*Servers,1); % Total Packets (voice + data)
    lostPackets = zeros(2*Servers,1); % Lost Packets (voice + data)
    lambda = Servers*lambda1 + Servers*lambda2; % total rate
    trace = zeros(ceil(2*SimulationTime*lambda), (2*Servers)+1);
    eventNumber = 0;
    clock = 0;
    voicePacket = zeros(Servers,1); % Indicate if the packet in service is voice or not
    while clock<SimulationTime
        eventNumber = eventNumber + 1;
        trace(eventNumber,:) = [clock transpose(systemPackets)];
        [clock,event] = min(futureEvents);
        switch event
            case 1 % Arrives voice packet
                futureEvents(1) = clock + random('exp',1/(Servers*lambda1));
                randomServer = ceil(Servers*rand());
                totalPackets(2*randomServer-1) =
totalPackets(2*randomServer-1) + 1;
                if futureEvents(randomServer+2) == inf
                    futureEvents(randomServer+2) = clock +
random('exp',1/mu1);
                    voicePacket(randomServer) = 1;
                    systemPackets(2*randomServer-1) =
systemPackets(2*randomServer-1) + 1;
                else
                    if voicePacket(randomServer) == 0
                        futureEvents(randomServer+2) = clock + random('exp',1/mu1);
                        voicePacket(randomServer) = 1;
                        systemPackets(2*randomServer) = systemPackets(2*randomServer) +
1;
                    else
                        I = find(voicePacket(:,1)==0);
                        if ~isempty(I) && balanceTrigger == 1

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        if futureEvents(I(1)+2) == inf
            futureEvents(I(1)+2) = clock + random('exp',1/mu1);
            voicePacket(I(1)) = 1;
            systemPackets(2*I(1)-1) = systemPackets(2*I(1)-1) + 1;
        else
            futureEvents(I(1)+2) = clock + random('exp',1/mu1);
            voicePacket(I(1)) = 1;
            systemPackets(2*I(1)) = systemPackets(2*I(1)) + 1;
        end
    else
        lostPackets(2*randomServer-1) = lostPackets(2*randomServer-1)
+ 1;
    end
end
end
end
case 2 % Arrives data packet
    futureEvents(2) = clock + random('exp',1/(Servers*lambda2));
    randomServer = ceil(Servers*rand());
    totalPackets(2*randomServer) = totalPackets(2*randomServer)
+ 1;
    if futureEvents(randomServer+2) == inf
        futureEvents(randomServer+2) = clock +
random('exp',1/mu2);
        voicePacket(randomServer) = 0;
        systemPackets(2*randomServer-1) =
systemPackets(2*randomServer-1) +
systemPackets(2*randomServer-1) + 1;
        else
            systemPackets(2*randomServer) =
systemPackets(2*randomServer) +
systemPackets(2*randomServer) + 1; % send packet to the queue
        end
        otherwise % A packet is processed
        randomServer = event-2;
        systemPackets(2*randomServer-1) = systemPackets(2*randomServer-1) - 1;
        % systemPackets = systemPackets - [1 0];
        if systemPackets(2*randomServer) > 0
            futureEvents(randomServer+2) = clock +
random('exp',1/mu2);
            systemPackets(2*randomServer-1) = systemPackets(2*randomServer-1) +
1;
            systemPackets(2*randomServer) = systemPackets(2*randomServer) - 1;
            % systemPackets = systemPackets + [1 -1];
            voicePacket(randomServer) = 0;
            else
                futureEvents(randomServer+2) = inf;
            end
        end
    end
trace = trace(1:eventNumber,:);
trace2 = [trace(:,1),trace(:,2)+trace(:,3)];
dt = trace(2:end,1) - trace(1:end-1,1);

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```
areas = (trace2(1:end-1,2)).*dt;
packetsAverage = sum(areas)/SimulationTime
end
```

A continuación se expone el código que se utilizó para hallar el número de simulaciones necesarias para alcanzar el nivel de confianza estudiado en el capítulo 9.2:

```
function [m,v,delta] = simulationConfidenceLevel(lambda1, lambda2, mu1, mu2, voiceChannel,
dataChannel, simulationTime, balance, interval, confidence)
sum1 = 0; % Accumulate the samples
sum2 = 0; % Accumulate the squared samples
simulationNumber = 0; % Number of simulations
percentage = 100; % Confidence Level
while ((percentage/100 > interval)*(simulationNumber < 50) + (simulationNumber < 5))
    simulationNumber = simulationNumber + 1; % Increase by one the Number of Simulations
    [trace, packetsAverage, totalPackets, lostPackets] = testTesis2(lambda1, lambda2,
mu1, mu2, N1, N2, SimulationTime, balanceTrigger); % Calculate the average number of packet on the system by trace.
    sum1 = sum1 + packetsAverage; % Accumulate the samples
    sum2 = sum2 + packetsAverage^2; % Accumulate the squared samples
    if simulationNumber > 1
        m = sum1/simulationNumber; % Calculate the temporary average
        v = (sum2 - sum1.*sum1/simulationNumber)/(simulationNumber - 1);
        percentil = tinv(0.5 + confidence/2, simulationNumber-1)
        delta = percentil*sqrt(v/simulationNumber);
        percentage = 100*delta(1)/m(1);
        disp(['Con ' num2str(simulationNumber) ' simulaciones, se estima el promedio de paquetes: ' num2str(m(1)) ' mas o menos ' num2str(percentage) '%']);
    end
end
end
```

D. Código de Simulación para los escenarios del capítulo 9.

A continuación se expone el código utilizado para generar la distribución de los 2000 usuarios al momento de iniciar la simulación:

```
function [users,antennaCell] = generateUsers2()
% 1. Determine the eNodeB locations
Nodes = 12; % Cells on the network
N = 6 % RB number
f = 1.9 % Carrier frequency (GHz)
R = 0.5; % Coverage Radius = 0.5km
```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```
% eNB1 = [3*R, 6*R];
% eNB2 = [6*R, 6*R];
% eNB3 = [3*R+3*R/2, 6*R-3*sqrt(3)*R/2];
% eNB4 = [3*R+9*R/2, 6*R-3*sqrt(3)*R/2];

eNBs = zeros(4,2); % eNodeB matrix

eNBs(1,1) = 3*R;
eNBs(1,2) = 6*R;
eNBs(2,1) = 6*R;
eNBs(2,2) = 6*R;
eNBs(3,1) = 3*R+3*R/2;
eNBs(3,2) = 6*R-3*sqrt(3)*R/2;
eNBs(4,1) = 3*R+9*R/2;
eNBs(4,2) = 6*R-3*sqrt(3)*R/2;

% 2.1 Determine the cell centers points

cell1 = [eNBs(1,1),eNBs(1,2)] + [-R,0];
cell2 = [eNBs(1,1),eNBs(1,2)] + [R/2,R*sqrt(3)/2];
cell3 = [eNBs(1,1),eNBs(1,2)] + [R/2,-R*sqrt(3)/2];

cell4 = [eNBs(3,1),eNBs(3,2)] + [-R,0];
cell5 = [eNBs(3,1),eNBs(3,2)] + [R/2,R*sqrt(3)/2];
cell6 = [eNBs(3,1),eNBs(3,2)] + [R/2,-R*sqrt(3)/2];

cell7 = [eNBs(2,1),eNBs(2,2)] + [-R,0];
cell8 = [eNBs(2,1),eNBs(2,2)] + [R/2,R*sqrt(3)/2];
cell9 = [eNBs(2,1),eNBs(2,2)] + [R/2,-R*sqrt(3)/2];

cell10 = [eNBs(4,1),eNBs(4,2)] + [-R,0];
cell11 = [eNBs(4,1),eNBs(4,2)] + [R/2,R*sqrt(3)/2];
cell12 = [eNBs(4,1),eNBs(4,2)] + [R/2,-R*sqrt(3)/2];

% 2.2 Determine the Antenna location by cell

antennaCell = zeros(Nodes,2);

antennaCell(1,1) = eNBs(1,1) + 0.001*(-R);
antennaCell(1,2) = eNBs(1,2);

antennaCell(2,1) = eNBs(1,1) + 0.001*(R/2);
antennaCell(2,2) = eNBs(1,2) + 0.001*(R*sqrt(3)/2);

antennaCell(3,1) = eNBs(1,1) + 0.001*(R/2);
antennaCell(3,2) = eNBs(1,2) + 0.001*(-R*sqrt(3)/2);

antennaCell(4,1) = eNBs(3,1) + 0.001*(-R);
```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

antennaCell(4,2) = eNBs(3,2);

antennaCell(5,1) = eNBs(3,1) + 0.001*(R/2);
antennaCell(5,2) = eNBs(3,2) + 0.001*(R*sqrt(3)/2);

antennaCell(6,1) = eNBs(3,1) + 0.001*(R/2);
antennaCell(6,2) = eNBs(3,2) + 0.001*(-R*sqrt(3)/2);

antennaCell(7,1) = eNBs(2,1) + 0.001*(-R);
antennaCell(7,2) = eNBs(2,2);

antennaCell(8,1) = eNBs(2,1) + 0.001*(R/2);
antennaCell(8,2) = eNBs(2,2) + 0.001*(R*sqrt(3)/2);

antennaCell(9,1) = eNBs(2,1) + 0.001*(R/2);
antennaCell(9,2) = eNBs(2,2) + 0.001*(-R*sqrt(3)/2);

antennaCell(10,1) = eNBs(4,1) + 0.001*(-R);
antennaCell(10,2) = eNBs(4,2);

antennaCell(11,1) = eNBs(4,1) + 0.001*(R/2);
antennaCell(11,2) = eNBs(4,2) + 0.001*(R*sqrt(3)/2);

antennaCell(12,1) = eNBs(4,1) + 0.001*(R/2);
antennaCell(12,2) = eNBs(4,2) + 0.001*(-R*sqrt(3)/2);

% 3. Define the Users basic parameters

users_size = 2000; % Define the users number in the network
users = zeros(users_size,7); % Define the users_matrix (x_coordinates,
y_coordinates, cell, distance, user_type, traffic_type, neighbor cell)

adjacency_matrix = [0 0 0 0 0 0 0 0 0 0 0 0;
                    0 0 0 0 0 0 1 0 0 0 0 0;
                    0 0 0 1 1 0 1 0 0 0 0 0;
                    0 0 1 0 0 0 0 0 0 0 0 0;
                    0 0 0 0 0 0 1 0 1 1 0 0;
                    0 0 0 0 0 0 0 0 0 1 0 0;
                    0 1 1 0 0 0 0 0 0 0 0 0;
                    0 0 0 0 0 0 0 0 0 0 0 0;
                    0 0 0 0 1 0 0 0 0 1 1 0;
                    0 0 0 0 1 1 0 0 1 0 0 0;
                    0 0 0 0 0 0 0 0 1 0 0 0;
                    0 0 0 0 0 0 0 0 0 0 0 1]; % Define the links
between nodes

% 4. Define the Users allocation (random)

x = zeros(users_size,1);

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

Y = zeros(users_size,1);
contador = 1;
while contador < users_size+1
    neighborDistance = inf;
    overload = rand();
    if overload < 0.0
        x_input = rand() - 0.5 + 1.75;
        y_input = 0.5*sqrt(3)*rand() - 0.5*sqrt(3)/2 + 3 - 0.25*sqrt(3);
        if x_input>abs(y_input-cell13(2))/sqrt(3)-R+cell13(1) && x_input<(-
1*abs(y_input-cell13(2))/sqrt(3)+R+cell13(1)) && y_input<cell13(2)+sqrt(3)*R/2 &&
y_input>cell13(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input; % X Coordinate
            users(contador,2) = y_input; % Y Coordinate
            users(contador,3) = 3; % Cell
            d = sqrt((x_input-antennaCell(1,1))^2+(y_input-antennaCell(1,2))^2); %
Distance from antenna to user
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R,6*R,x_input,y_input,150)) - 10*log10(12*N); % RSRP, f
(GHz), d (km)
            if users(contador,4) > -66
                users(contador,5) = 0; % Indicates if it is normal user or edge user.
0 normal, 1 edge
            else
                users(contador,5) = 1;
            end
            users(contador,6) = 1; % Indicates the traffic type

            for i = 1:Nodes
                if adjacency_matrix(users(contador,3),i) == 1
                    if neighborDistance > sqrt((x_input-antennaCell(i,1))^2+(y_input-
antennaCell(i,2))^2) && sqrt((x_input-antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) <
1.2
                        neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                        users(contador,7) = i; % Indicate the neighbor cell with
better RSRP
                    end
                end
            contador = contador + 1;
        end

    else
        x_input = 8*R*rand() + R;
        y_input = 7*sqrt(3)*R*rand()/2 + 6*R - 5*sqrt(3)*R/2;
        %neighborDistance = inf;
    end
end

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        if      x_input>abs(y_input-cell1(2))/sqrt(3)-R+cell1(1)      &&      x_input<(-
1*abs(y_input-cell1(2))/sqrt(3)+R+cell1(1))      &&      y_input<cell1(2)+sqrt(3)*R/2      &&
y_input>cell1(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input; % X Coordinate
            users(contador,2) = y_input; % Y Coordinate
            users(contador,3) = 1; % Cell
            d           = sqrt((x_input-antennaCell(1,1))^2+(y_input-
antennaCell(1,2))^2); % Distance from antenna to user
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R,6*R,x_input,y_input,270)) - 10*log10(12*N); % RSRP, f
(GHz), d (km)
            if users(contador,4) > -66
                users(contador,5) = 0; % Indicates if it is normal user or
edge user. 0 normal, 1 edge
            else
                users(contador,5) = 1;
            end
            users(contador,6) = 1; % Indicates the traffic type

            for i = 1:Nodes
                if adjacency_matrix(users(contador,3),i) == 1
                    if      neighborDistance      >      sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2)      &&      sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                        neighborDistance      =      sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                        users(contador,7) = i; % Indicate the
neighbor cell with better RSRP
                end
            end
            contador = contador + 1;
        end

        if      x_input>abs(y_input-cell2(2))/sqrt(3)-R+cell2(1)      &&      x_input<(-
1*abs(y_input-cell2(2))/sqrt(3)+R+cell2(1))      &&      y_input<cell2(2)+sqrt(3)*R/2      &&
y_input>cell2(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input;
            users(contador,2) = y_input;
            users(contador,3) = 2;
            d           = sqrt((x_input-antennaCell(2,1))^2+(y_input-
antennaCell(2,2))^2);
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R,6*R,x_input,y_input,30)) - 10*log10(12*N); % RSRP, f (GHz),
d (km)
            if users(contador,4) > -66
                users(contador,5) = 0;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        else
            users(contador,5) = 1;
        end
        users(contador,6) = 1;

        for i = 1:Nodes
            if adjacency_matrix(users(contador,3),i) == 1
                if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                    neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                    users(contador,7) = i;
                end
            end
            contador = contador + 1;
        end

        if x_input>abs(y_input-cell3(2))/sqrt(3)-R+cell3(1) && x_input<(-
1*abs(y_input-cell3(2))/sqrt(3)+R+cell3(1)) && y_input<cell3(2)+sqrt(3)*R/2 &&
y_input>cell3(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input;
            users(contador,2) = y_input;
            users(contador,3) = 3;
            d = sqrt((x_input-antennaCell(3,1))^2+(y_input-
antennaCell(3,2))^2);
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R,6*R,x_input,y_input,150)) - 10*log10(12*N); % RSRP, f
(GHz), d (km)
            if users(contador,4) > -66
                users(contador,5) = 0;
            else
                users(contador,5) = 1;
            end
            users(contador,6) = 1;

            for i = 1:Nodes
                if adjacency_matrix(users(contador,3),i) == 1
                    if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                        neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                        users(contador,7) = i;
                    end
                end
            contador = contador + 1;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

    end

    if      x_input>abs(y_input-cell14(2))/sqrt(3)-R+cell14(1)      &&      x_input<(-
1*abs(y_input-cell14(2))/sqrt(3)+R+cell14(1))      &&      y_input<cell14(2)+sqrt(3)*R/2      &&
y_input>cell14(2)-sqrt(3)*R/2
        X(contador,1) = x_input;
        Y(contador,1) = y_input;
        users(contador,1) = x_input;
        users(contador,2) = y_input;
        users(contador,3) = 4;
        d                  = sqrt((x_input-antennaCell(4,1))^2+(y_input-
antennaCell(4,2))^2);
        users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R + 3*R/2,6*R - 3*sqrt(3)*R/2,x_input,y_input,270)) -
10*log10(12*N); % RSRP, f (GHz), d (km)
        if users(contador,4) > -66
            users(contador,5) = 0;
        else
            users(contador,5) = 1;
        end
        users(contador,6) = 1;

        for i = 1:Nodes
            if adjacency_matrix(users(contador,3),i) == 1
                if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2)      &&      sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                    neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                    users(contador,7) = i;
                end
            end
            contador = contador + 1;
        end

        if      x_input>abs(y_input-cell15(2))/sqrt(3)-R+cell15(1)      &&      x_input<(-
1*abs(y_input-cell15(2))/sqrt(3)+R+cell15(1))      &&      y_input<cell15(2)+sqrt(3)*R/2      &&
y_input>cell15(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input;
            users(contador,2) = y_input;
            users(contador,3) = 5;
            d                  = sqrt((x_input-antennaCell(5,1))^2+(y_input-
antennaCell(5,2))^2);
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R + 3*R/2,6*R - 3*sqrt(3)*R/2,x_input,y_input,30)) -
10*log10(12*N); % RSRP, f (GHz), d (km)
            if users(contador,4) > -66
                users(contador,5) = 0;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        else
            users(contador,5) = 1;
        end
        users(contador,6) = 1;

        for i = 1:Nodes
            if adjacency_matrix(users(contador,3),i) == 1
                if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && neighborDistance < 1.2
                    neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                    users(contador,7) = i;
                end
            end
            contador = contador + 1;
        end

        if x_input>abs(y_input-cell6(2))/sqrt(3)-R+cell6(1) && x_input<(-
1*abs(y_input-cell6(2))/sqrt(3)+R+cell6(1)) && y_input<cell6(2)+sqrt(3)*R/2 &&
y_input>cell6(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input;
            users(contador,2) = y_input;
            users(contador,3) = 6;
            d = sqrt((x_input-antennaCell(6,1))^2+(y_input-
antennaCell(6,2))^2);
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R + 3*R/2,6*R - 3*sqrt(3)*R/2,x_input,y_input,150)) -
10*log10(12*N); % RSRP, f (GHz), d (km)
            if users(contador,4) > -66
                users(contador,5) = 0;
            else
                users(contador,5) = 1;
            end
            users(contador,6) = 1;

            for i = 1:Nodes
                if adjacency_matrix(users(contador,3),i) == 1
                    if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && neighborDistance < 1.2
                        neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                        users(contador,7) = i;
                    end
                end
            end
            contador = contador + 1;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

    end

    if      x_input>abs(y_input-cell17(2))/sqrt(3)-R+cell17(1)      &&      x_input<(-
1*abs(y_input-cell17(2))/sqrt(3)+R+cell17(1))      &&      y_input<cell17(2)+sqrt(3)*R/2      &&
y_input>cell17(2)-sqrt(3)*R/2
        X(contador,1) = x_input;
        Y(contador,1) = y_input;
        users(contador,1) = x_input;
        users(contador,2) = y_input;
        users(contador,3) = 7;
        d                  = sqrt((x_input-antennaCell(7,1))^2+(y_input-
antennaCell(7,2))^2);
        users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(6*R,6*R,x_input,y_input,270)) - 10*log10(12*N); % RSRP, f
(GHz), d (km)
        if users(contador,4) > -66
            users(contador,5) = 0;
        else
            users(contador,5) = 1;
        end
        users(contador,6) = 1;

        for i = 1:Nodes
            if adjacency_matrix(users(contador,3),i) == 1
                if neighborDistance      >      sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2)      &&      sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                    neighborDistance      =      sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                    users(contador,7) = i;
                end
            end
            contador = contador + 1;
        end

        if      x_input>abs(y_input-cell18(2))/sqrt(3)-R+cell18(1)      &&      x_input<(-
1*abs(y_input-cell18(2))/sqrt(3)+R+cell18(1))      &&      y_input<cell18(2)+sqrt(3)*R/2      &&
y_input>cell18(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input;
            users(contador,2) = y_input;
            users(contador,3) = 8;
            d                  = sqrt((x_input-antennaCell(8,1))^2+(y_input-
antennaCell(8,2))^2);
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(6*R,6*R,x_input,y_input,30)) - 10*log10(12*N); % RSRP, f (GHz),
d (km)
            if users(contador,4) > -66
                users(contador,5) = 0;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

    else
        users(contador,5) = 1;
    end
    users(contador,6) = 1;

    for i = 1:Nodes
        if adjacency_matrix(users(contador,3),i) == 1
            if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                users(contador,7) = i;
            end
        end
        contador = contador + 1;
    end

    if x_input>abs(y_input-cell9(2))/sqrt(3)-R+cell9(1) && x_input<(-
1*abs(y_input-cell9(2))/sqrt(3)+R+cell9(1)) && y_input<cell9(2)+sqrt(3)*R/2 &&
y_input>cell9(2)-sqrt(3)*R/2
        X(contador,1) = x_input;
        Y(contador,1) = y_input;
        users(contador,1) = x_input;
        users(contador,2) = y_input;
        users(contador,3) = 9;
        d = sqrt((x_input-antennaCell(9,1))^2+(y_input-
antennaCell(9,2))^2);
        users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(6*R,6*R,x_input,y_input,150)) - 10*log10(12*N); % RSRP, f
(GHz), d (km)
        if users(contador,4) > -66
            users(contador,5) = 0;
        else
            users(contador,5) = 1;
        end
        users(contador,6) = 1;

        for i = 1:Nodes
            if adjacency_matrix(users(contador,3),i) == 1
                if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                    neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                    users(contador,7) = i;
                end
            end
        contador = contador + 1;
    end

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

    end

    if x_input>abs(y_input-cell10(2))/sqrt(3)-R+cell10(1) && x_input<(-
1*abs(y_input-cell10(2))/sqrt(3)+R+cell10(1)) && y_input<cell10(2)+sqrt(3)*R/2 &&
y_input>cell10(2)-sqrt(3)*R/2
        X(contador,1) = x_input;
        Y(contador,1) = y_input;
        users(contador,1) = x_input;
        users(contador,2) = y_input;
        users(contador,3) = 10;
        d = sqrt((x_input-antennaCell(10,1))^2+(y_input-
antennaCell(10,2))^2);
        users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R + 9*R/2,6*R - 3*sqrt(3)*R/2,x_input,y_input,270)) -
10*log10(12*N); % RSRP, f (GHz), d (km)
        if users(contador,4) > -66
            users(contador,5) = 0;
        else
            users(contador,5) = 1;
        end
        users(contador,6) = 1;

        for i = 1:Nodes
            if adjacency_matrix(users(contador,3),i) == 1
                if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                    neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                    users(contador,7) = i;
                end
            end
            contador = contador + 1;
        end

        if x_input>abs(y_input-cell11(2))/sqrt(3)-R+cell11(1) && x_input<(-
1*abs(y_input-cell11(2))/sqrt(3)+R+cell11(1)) && y_input<cell11(2)+sqrt(3)*R/2 &&
y_input>cell11(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input;
            users(contador,2) = y_input;
            users(contador,3) = 11;
            d = sqrt((x_input-antennaCell(11,1))^2+(y_input-
antennaCell(11,2))^2);
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R + 9*R/2,6*R - 3*sqrt(3)*R/2,x_input,y_input,30)) -
10*log10(12*N); % RSRP, f (GHz), d (km)
            if users(contador,4) > -66
                users(contador,5) = 0;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        else
            users(contador,5) = 1;
        end
        users(contador,6) = 1;

        for i = 1:Nodes
            if adjacency_matrix(users(contador,3),i) == 1
                if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) < 1.2
                    neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                    users(contador,7) = i;
                end
            end
            contador = contador + 1;
        end

        if x_input>abs(y_input-cell12(2))/sqrt(3)-R+cell12(1) && x_input<(-
1*abs(y_input-cell12(2))/sqrt(3)+R+cell12(1)) && y_input<cell12(2)+sqrt(3)*R/2 &&
y_input>cell12(2)-sqrt(3)*R/2
            X(contador,1) = x_input;
            Y(contador,1) = y_input;
            users(contador,1) = x_input;
            users(contador,2) = y_input;
            users(contador,3) = 12;
            d = sqrt((x_input-antennaCell(12,1))^2+(y_input-
antennaCell(12,2))^2);
            users(contador,4) = 30 - 20*log10(d) - 20*log10(f) - 92.45 -
lossFunction(angleFunction(3*R + 9*R/2,6*R - 3*sqrt(3)*R/2,x_input,y_input,150)) -
10*log10(12*N); % RSRP, f (GHz), d (km)
            if users(contador,4) > -66
                users(contador,5) = 0;
            else
                users(contador,5) = 1;
            end
            users(contador,6) = 1;

            for i = 1:Nodes
                if adjacency_matrix(users(contador,3),i) == 1
                    if neighborDistance > sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2) && neighborDistance = sqrt((x_input-
antennaCell(i,1))^2+(y_input-antennaCell(i,2))^2);
                        users(contador,7) = i;
                end
            end
            contador = contador + 1;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

    end
    end
end
nbins = 12;
[counts,centers] = hist(users(:,3),nbins);
bar(centers,counts)
end

```

A continuación se expone el código utilizado para generar la simulación (llegadas y salida de paquetes, invocación de las funciones de tabla de enrutamiento y de balanceo de carga).

```

function
[trace,totalPackets2,packetsLoss,packetsTransmitted,handovers,throughputAverage] =
twoFlows4(lambda1, lambda2, mu1, mu2, N1, N2, buffer_size, simulationTime, balance)
    % If BW = 1.4 MHz, the subcarriers number is 72 --> RB = 6
    % If BW = 3 MHz, the subcarriers number is 180 --> RB = 15
    % If BW = 5 MHz, the subcarriers number is 300 --> RB = 25
    % If BW = 10 MHz, the subcarriers number is 600 --> RB = 50
    % If BW = 15 MHz, the subcarriers number is 900 --> RB = 75
    % If BW = 20 MHz, the subcarriers number is 1200 --> RB = 100
    Nodes = 12; % Number of Cells on the network
    N = N1 + N2; % Total Channels (RB) by cell
    N_total = 12*N; % The quantity of channels (RB) are 12 times (N1 + N2). N1 + N2
depends on BW
    modulationNormalUsers = 6; % Modulation 64QAM
    modulationEdgeUsers = 4; % Modulation 16QAM
    [users,antennaCell] = generateUsers2(); % Users matrix and antenna Cell info
    futureEvents = inf*ones(Nodes*N+2,2); % This vector will save the arrival info (2
first elements) and service time (next elements).
    % The first cell
will save from 3rd element to 8th element
    % The second cell
will save from 9th element to 14th and so on

    futureEvents(1,1) = random('exp',1/lambda1); % Schedule the first voice packet
    futureEvents(2,1) = random('exp',1/lambda2); % Schedule the first data packet
    totalPackets = zeros(2*Nodes,1); % Total number of packets sent on the network
divided by voice packets and data packets
    totalPackets2 = zeros(2,Nodes);
    channelsAvailables = zeros(Nodes,1);
    handovers = zeros(Nodes,2);
    systemPackets = zeros(2,Nodes*3); % This vector will save the info related to
packets system (voice, data, queue) by cell
    % Row 1 is related to
Normal users and Row 2 is related to Edge users
    systemPacketsTotal = zeros(1,Nodes*3); % Sum the Total packets
    packetsTransmitted = zeros(4,Nodes); % This array will save transmitted packets
information, Row 1 Voice transmitted packets from normal users,

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

% Row 2 the Data transmitted packets from normal
users, Row 3 the Voice transmitted packets
% from edge users and Row 4 Data transmitted
packets from edge user
packetsLoss = zeros(4,Nodes); % This array will save packet loss information, Row
1 Voice lost packets from normal users,
% Row 2 the Data lost packets
from normal users, Row 3 the Voice lost packets
% from edge users and Row 4
Data lost packets from edge user
lambda = lambda1 + lambda2; % Total Rate
throughput = zeros(ceil(2*simulationTime*lambda),Nodes); % Vector to calculate
throughput
trace = zeros(ceil(2*simulationTime*lambda),3*Nodes+1); % The event i occurred in the
time trace(i,1) and the system has

% trace(i,3*Cell-1) voice packets, trace(i,3*Cell) data packets and

% trace(i,3*Cell+1) packets in queue.
numEvent = 0;
clock = 0;
while clock < simulationTime
    numEvent = numEvent + 1; % Increase the event number
    systemPacketsTotal = systemPackets(1,:) + systemPackets(2,:); % Sum the total
    packets
    trace(numEvent,:) = [clock systemPacketsTotal]; % Writes in trace the clock
    and system packets at current time
    [clock,event] = min(abs(futureEvents(:,1))); % Clock will be the min value
    and event the array position
    for i = 1:Nodes
        normalPackets = size(find(futureEvents(3+(i-1)*N:2+i*N,2)==0),1);
        edgePackets = size(find(futureEvents(3+(i-1)*N:2+i*N,2)==1),1);
        throughput(numEvent,i) = 12*normalPackets*modulationNormalUsers +
        12*edgePackets*modulationEdgeUsers;
    end
    switch event
        case 1 % Arrives a voice packet
            futureEvents(1,1) = clock + random('exp',1/lambda1); %
Schedule next voice packet arrival
            user_random = ceil(2000*rand()); % Choose a random user
            cell_random = users(user_random,3); % Select the cell
            belongs to random user
            totalPackets(2*(cell_random-1)+1) =
totalPackets(2*(cell_random-1)+1) + 1; % Increase 1 voice packet to totalPackets variable
            if users(user_random,5) == 0 % Normal user
                I = find(futureEvents(N*(cell_random-
1)+3:N*(cell_random-1)+N1+2,1)==inf); % Find a voice channel available
                if ~isempty(I)
                    futureEvents(I(1)+2+N*(cell_random-1),1) =
clock + random('exp',1/mu1); % Assign a service time at current channel
                end
            end
    end
end

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        futureEvents(I(1)+2+N*(cell_random-1),2)      =
0; % Link the channel with a normal user
        systemPackets(1,3*(cell_random-1)+1)      =
systemPackets(1,3*(cell_random-1)+1) + 1; % Add a voice packet in the system
        channelsAvailables(cell_random,1)      =
channelsAvailables(cell_random,1) + 1;
        else
            I      = find(futureEvents(N*(cell_random-
1)+3:N*(cell_random-1)+N1+2,1)<0); % If there is one data packet in a voice channel
            if ~isempty(I)
                futureEvents(I(1)+2+N*(cell_random-
1),1) = clock + random('exp',1/mu1); % Remove the data packet and replace by

% the voice packet
        futureEvents(I(1)+2+N*(cell_random-
1),2) = 0; % Link the channel with a normal user

        systemPackets(1,3*(cell_random-1)+1)
= systemPackets(1,3*(cell_random-1)+1) + 1; % Add a voice packet in the system
        systemPackets(1,3*(cell_random-1)+2)
= systemPackets(1,3*(cell_random-1)+2) - 1; % Remove a data packet
            if           systemPackets(1,3*(cell_random-1)+3) +
systemPackets(2,3*(cell_random-1)+3) < buffer_size
                systemPackets(1,3*(cell_random-1)+3)
= systemPackets(1,3*(cell_random-1)+3) + 1; % Add a data packet to queue
            else
                packetsLoss(2,cell_random) = packetsLoss(2,cell_random) +
1;
            end
            else
                I = find(futureEvents(N*(cell_random-
1)+3+N1:N*(cell_random-1)+2+N,1)==inf); % Find a data channel available
                if ~isempty(I)
                    futureEvents(N*(cell_random-
1)+2+N1+I(1),1) = -(clock + random('exp',1/mu1)); % Schedule data time service in a data

% channel available
        futureEvents(N*(cell_random-
1)+2+N1+I(1),2) = 0; % Link the channel with a normal user

        systemPackets(1,3*(cell_random-1)+1) = systemPackets(1,3*(cell_random-1)+1) + 1;
% Add a voice packet in the system

        channelsAvailables(cell_random,1) = channelsAvailables(cell_random,1) + 1;
        else % LB or lost packet
            [users, orderLB, cellBalance]
= runLB2(users, cell_random, channelsAvailables, N, antennaCell, balance);
            if orderLB ~= 0
                handovers(cell_random,orderLB)
= handovers(cell_random,orderLB) + 1;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

I      =
find(futureEvents(N*(cellBalance-1)+3:N*(cellBalance-1)+N1+2,1)==inf); % Find a voice
channel available
if isempty(I)

futureEvents(I(1)+2+N*(cellBalance-1),1) = clock + random('exp',1/mu1); % Assign
a service time at current channel

futureEvents(I(1)+2+N*(cellBalance-1),2) = 0; % Link the channel with a normal
user

systemPackets(1,3*(cellBalance-1)+1) = systemPackets(1,3*(cellBalance-1)+1) + 1;
% Add a voice packet in the system

channelsAvailables(cellBalance,1) = channelsAvailables(cellBalance,1) + 1;
else
I      =
find(futureEvents(N*(cellBalance-1)+3:N*(cellBalance-1)+N1+2,1)<0); % If there is one data
packet in a voice channel
if isempty(I)

futureEvents(I(1)+2+N*(cellBalance-1),1) = clock + random('exp',1/mu1); % Remove
the data packet and replace by

% the voice packet

futureEvents(I(1)+2+N*(cellBalance-1),2) = 0; % Link the channel with a normal
user

systemPackets(1,3*(cellBalance-1)+1) = systemPackets(1,3*(cellBalance-1)+1) + 1;
% Add a voice packet in the system
systemPackets(1,3*(cellBalance-1)+2) = systemPackets(1,3*(cellBalance-1)+2) - 1; % Remove data packet
if
systemPackets(1,3*(cellBalance-1)+3) + systemPackets(2,3*(cellBalance-1)+3) < bufferSize
systemPackets(1,3*(cellBalance-1)+3) =
systemPackets(1,3*(cellBalance-1)+3) + 1; % Add a data packet to queue
else
packetsLoss(2,cell_random) =
packetsLoss(2,cell_random) + 1;
end
else
I      =
find(futureEvents(N*(cellBalance-1)+3:N*(cellBalance-1)+2+N,1)==inf); % Find a data
channel available
if
isempty(I)

futureEvents(N*(cellBalance-1)+2+N1+I(1),1) = -(clock + random('exp',1/mu1)); % Schedule data time service in a data

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```
%  
channel available  
  
futureEvents(N*(cellBalance-1)+2+N1+I(1),2) = 0; % Link the channel with a normal  
user  
  
systemPackets(1,3*(cellBalance-1)+1) = systemPackets(1,3*(cellBalance-1)+1) + 1;  
% Add a voice packet in the system  
  
channelsAvailables(cellBalance,1) = channelsAvailables(cellBalance,1) + 1;  
end  
end  
else  
  
packetsLoss(1,cell_random) = packetsLoss(1,cell_random) + 1;  
end  
  
end  
end  
else % Edge user  
I = find(futureEvents(N*(cell_random-  
1)+3:N*(cell_random-1)+N1+2,1)==inf); % Find a voice channel available  
if ~isempty(I)  
    futureEvents(I(1)+2+N*(cell_random-1),1) =  
clock + random('exp',1/mul); % Assign a service time in at current channel  
    futureEvents(I(1)+2+N*(cell_random-1),2) =  
1; % Link the channel with an edge user  
    systemPackets(2,3*(cell_random-1)+1) =  
systemPackets(2,3*(cell_random-1)+1) + 1; % Add a voice packet in the system  
    channelsAvailables(cell_random,1) =  
channelsAvailables(cell_random,1) + 1;  
else  
    I = find(futureEvents(N*(cell_random-  
1)+3:N*(cell_random-1)+N1+2,1)<0); % If there is one data packet in a voice channel  
    if ~isempty(I)  
        futureEvents(I(1)+2+N*(cell_random-  
1),1) = clock + random('exp',1/mul); % Remove the data packet and replace by  
  
    % the voice packet  
    futureEvents(I(1)+2+N*(cell_random-  
1),2) = 1; % Link the channel with an edge user  
  
    systemPackets(2,3*(cell_random-1)+1) =  
systemPackets(2,3*(cell_random-1)+1) + 1; % Add a voice packet in the system  
    systemPackets(2,3*(cell_random-1)+2) =  
systemPackets(2,3*(cell_random-1)+2) - 1;  
    if systemPackets(1,3*(cell_random-1)+3) +  
systemPackets(2,3*(cell_random-1)+3) < buffer_size
```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        systemPackets(2,3*(cell_random-1)+3) = 
systemPackets(2,3*(cell_random-1)+3) + 1; % Add a data packet to queue
    else
        packetsLoss(4,cell_random) = packetsLoss(4,cell_random) +
1;
    end
else
    I = find(futureEvents(N*(cell_random-
1)+3+N1:N*(cell_random-1)+2+N,1)==inf); % Find a data channel available
    if ~isempty(I)
        futureEvents(N*(cell_random-
1)+2+N1+I(1),1) = -(clock + random('exp',1/mul)); % Schedule data time service in a data

        % channel available
        futureEvents(N*(cell_random-
1)+2+N1+I(1),2) = 1; % Link the channel with an edge user

        systemPackets(2,3*(cell_random-1)+1) = systemPackets(2,3*(cell_random-1)+1) + 1;
% Add a voice packet in the system

        channelsAvailables(cell_random,1) = channelsAvailables(cell_random,1) + 1;
        else % LB or lost packet
            [users, orderLB, cellBalance]
= runLB2(users, cell_random, channelsAvailables, N, antennaCell, balance);
            if orderLB ~= 0
                handovers(cell_random,orderLB) =
handovers(cell_random,orderLB) + 1;
                I =
find(futureEvents(N*(cellBalance-1)+3:N*(cellBalance-1)+N1+2,1)==inf); % Find a voice
channel available
                if ~isempty(I)

                    futureEvents(I(1)+2+N*(cellBalance-1),1) = clock + random('exp',1/mul); % Assign
a service time in at current channel

                    futureEvents(I(1)+2+N*(cellBalance-1),2) = 1; % Link the channel with an edge user

                    systemPackets(2,3*(cellBalance-1)+1) = systemPackets(2,3*(cellBalance-1)+1) + 1;
% Add a voice packet in the system

                    channelsAvailables(cellBalance,1) = channelsAvailables(cellBalance,1) + 1;
                    else
                        I =
find(futureEvents(N*(cellBalance-1)+3:N*(cellBalance-1)+N1+2,1)<0); % If there is one data
packet in a voice channel
                        if ~isempty(I)

                            futureEvents(I(1)+2+N*(cellBalance-1),1) = clock + random('exp',1/mul); % Remove
the data packet and replace by

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

% the voice packet

futureEvents(I(1)+2+N*(cellBalance-1),2) = 1; % Link the channel with an edge user

systemPackets(2,3*(cellBalance-1)+1) = systemPackets(2,3*(cellBalance-1)+1) + 1;
% Add a voice packet in the system
systemPackets(2,3*(cellBalance-1)+2) =
systemPackets(2,3*(cellBalance-1)+2) - 1;
if systemPackets(1,3*(cellBalance-1)+3) +
systemPackets(2,3*(cellBalance-1)+3) < buffer_size
    systemPackets(2,3*(cellBalance-1)+3) =
systemPackets(2,3*(cellBalance-1)+3) + 1; % Add a data packet to queue
else
    packetsLoss(4,cell_random) =
packetsLoss(4,cell_random) + 1;
end
else
I =
find(futureEvents(N*(cellBalance-1)+3+N1:N*(cellBalance-1)+2+N,1)==inf); % Find a data
channel available
if
~isempty(I)

futureEvents(N*(cellBalance-1)+2+N1+I(1),1) = -(clock + random('exp',1/mu1)); %
Schedule data time service in a data

% channel available

futureEvents(N*(cellBalance-1)+2+N1+I(1),2) = 1; % Link the channel with an edge
user

systemPackets(2,3*(cellBalance-1)+1) = systemPackets(2,3*(cellBalance-1)+1) + 1;
% Add a voice packet in the system

channelsAvailables(cellBalance,1) = channelsAvailables(cellBalance,1) + 1;
end
end
else

packetsLoss(3,cell_random) = packetsLoss(3,cell_random) + 1;
end
end
end
end
end
case 2 % Arrives a data packet

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        futureEvents(2,1) = clock + random('exp',1/lambda2); %

Schedule next voice packet arrival

        user_random = ceil(2000*rand()); % Choose a random user
        cell_random = users(user_random,3); % Select the cell
belongs to random user

        totalPackets(2*(cell_random-1)+2) =
totalPackets(2*(cell_random-1)+2) + 1; % Increase 1 data packet to totalPackets variable
        if users(user_random,5) == 0 % Normal user
            %systemPackets(1,3*(cell_random-1)+2) =
systemPackets(1,3*(cell_random-1)+2) + 1; % Add a data packet in the system
        I = find(futureEvents(N*(cell_random-1)+3+N1:N*(cell_random-1)+2+N,1)==inf); % Find a data channel available
        if ~isempty(I)

            futureEvents(N*(cell_random-1)+2+N1+I(1),1)
= clock + random('exp',1/mu2); % Assign a service time in at current channel
            futureEvents(N*(cell_random-1)+2+N1+I(1),2)
= 0; % Link the channel with a normal user
            systemPackets(1,3*(cell_random-1)+2) =
systemPackets(1,3*(cell_random-1)+2) + 1; % Add a data packet in the system
            channelsAvailables(cell_random,1) =
channelsAvailables(cell_random,1) + 1;
        else
            I = find(futureEvents(N*(cell_random-1)+3+N1:N*(cell_random-1)+2+N)<0); % If there is one voice packet in a data channel
            if ~isempty(I)
                futureEvents(N*(cell_random-1)+2+N1+I(1),1) = clock + random('exp',1/mu2); % Remove the voice packet and assign a

% service time in at current channel
                futureEvents(N*(cell_random-1)+2+N1+I(1),2) = 0; % Link the channel with a normal user
                systemPackets(1,3*(cell_random-1)+2) =
systemPackets(1,3*(cell_random-1)+2) + 1; % Add a data packet in the system
                systemPackets(1,3*(cell_random-1)+1) =
systemPackets(1,3*(cell_random-1)+1) - 1; % Remove a voice packet in the system
                packetsLoss(1,cell_random) =
packetsLoss(1,cell_random) + 1;
            else
                I = find(futureEvents(N*(cell_random-1)+3:N*(cell_random-1)+N1+2,1)==inf); % Find a voice channel available
                if ~isempty(I)
                    futureEvents(N*(cell_random-1)+2+I(1),1) = -(clock + random('exp',1/mu2)); % Schedule data time service in a voice
channel

% available
                futureEvents(N*(cell_random-1)+2+I(1),2) = 0; % Link the channel with a normal user

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        systemPackets(1,3*(cell_random-1)+2) = 
systemPackets(1,3*(cell_random-1)+2) + 1; % Add a data packet in the system

        channelsAvailables(cell_random,1) = channelsAvailables(cell_random,1) + 1;

        else
            if
systemPackets(1,3*(cell_random-1)+3) + systemPackets(2,3*(cell_random-1)+3) < buffer_size
% Queue available

            systemPackets(1,3*(cell_random-1)+3) = systemPackets(1,3*(cell_random-1)+3) + 1;
% Add a data packet to queue
            else % LB or Queue not
available, Lost data packet
                [users, orderLB, cellBalance] =
runLB2(users, cell_random, channelsAvailables, N, antennaCell, balance);
                if orderLB ~= 0
                    handovers(cell_random,orderLB) =
handovers(cell_random,orderLB) + 1;
                    I =
find(futureEvents(N*(cellBalance-1)+3+N1:N*(cellBalance-1)+2+N,1)==inf); % Find a data
channel available
                    if ~isempty(I)

                        futureEvents(N*(cellBalance-1)+2+N1+I(1),1) = clock + random('exp',1/mu2); %
Assign a service time in at current channel

                        futureEvents(N*(cellBalance-1)+2+N1+I(1),2) = 0; % Link the channel with a normal
user
                        systemPackets(1,3*(cellBalance-1)+2) =
systemPackets(1,3*(cellBalance-1)+2) + 1; % Add a data packet in the system

                        channelsAvailables(cellBalance,1) = channelsAvailables(cellBalance,1) + 1;
                        else
                            I =
find(futureEvents(N*(cellBalance-1)+3:N*(cellBalance-1)+N1+2,1)==inf); % Find a voice
channel available
                            if
isempty(I)

                                futureEvents(N*(cellBalance-1)+2+I(1),1) = -(clock + random('exp',1/mu2)); %
Schedule data time service in a voice channel

                                % available

                                futureEvents(N*(cellBalance-1)+2+I(1),2) = 0; % Link the channel with a normal
user
                                systemPackets(1,3*(cell_random-1)+2) =
systemPackets(1,3*(cell_random-1)+2) + 1; % Add a data packet in the system

                                channelsAvailables(cellBalance,1) = channelsAvailables(cellBalance,1) + 1;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

end
end
else

    packetsLoss(2,cell_random) = packetsLoss(2,cell_random) + 1;
    end
end
end
end

else % Edge user
    % systemPackets(2,3*(cell_random-1)+2) =
systemPackets(2,3*(cell_random-1)+2) + 1; % Add a data packet in the system
    I = find(futureEvents(N*(cell_random-1)+3+N1:N*(cell_random-1)+2+N,1)==inf); % Find a data channel available
    if ~isempty(I)
        futureEvents(N*(cell_random-1)+2+N1+I(1),1)
= clock + random('exp',1/mu2); % Assign a service time in at current channel
        futureEvents(N*(cell_random-1)+2+N1+I(1),2)
= 1; % Link the channel with an edge user
        systemPackets(2,3*(cell_random-1)+2) =
systemPackets(2,3*(cell_random-1)+2) + 1; % Add a data packet in the system
        channelsAvailables(cell_random,1) =
channelsAvailables(cell_random,1) + 1;
    else
        I = find(futureEvents(N*(cell_random-1)+3+N1:N*(cell_random-1)+2+N)<0); % If there is one voice packet in a data channel
        if ~isempty(I)
            futureEvents(N*(cell_random-1)+2+N1+I(1),1) = clock + random('exp',1/mu2); % Remove the voice packet and assign a
            % service time in at current channel
            futureEvents(N*(cell_random-1)+2+N1+I(1),2) = 1; % Link the channel with an edge user
            systemPackets(2,3*(cell_random-1)+2) =
systemPackets(2,3*(cell_random-1)+2) + 1; % Add a data packet in the system
            systemPackets(2,3*(cell_random-1)+1) =
systemPackets(2,3*(cell_random-1)+1) - 1; % Remove a voice packet in the system
            packetsLoss(3,cell_random) =
packetsLoss(3,cell_random) + 1;
        else
            I = find(futureEvents(N*(cell_random-1)+3:N*(cell_random-1)+N1+2,1)==inf); % Find a voice channel available
            if ~isempty(I)
                futureEvents(N*(cell_random-1)+2+I(1),1) = -(clock + random('exp',1/mu2)); % Schedule data time service in a voice
channel

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

% available

futureEvents(N*(cell_random-
1)+2+I(1),2) = 1; % Link the channel with an edge user
systemPackets(2,3*(cell_random-1)+2) =
systemPackets(2,3*(cell_random-1)+2) + 1; % Add a data packet in the system

channelsAvailables(cell_random,1) = channelsAvailables(cell_random,1) + 1;
else
if
systemPackets(1,3*(cell_random-1)+3) + systemPackets(2,3*(cell_random-1)+3) < buffer_size
% Queue available

systemPackets(2,3*(cell_random-1)+3) = systemPackets(2,3*(cell_random-1)+3) + 1;
% Add a data packet to queue
else % LB or Queue not
available, Lost data packet
[users, orderLB, cellBalance]
= runLB2(users, cell_random, channelsAvailables, N, antennaCell, balance);
if orderLB ~= 0
handovers(cell_random,orderLB) =
handovers(cell_random,orderLB) + 1;
I =
find(futureEvents(N*(cellBalance-1)+3+N1:N*(cellBalance-1)+2+N,1)==inf); % Find a data
channel available
if ~isempty(I)

futureEvents(N*(cellBalance-1)+2+N1+I(1),1) = clock + random('exp',1/mu2); %
Assign a service time in at current channel

futureEvents(N*(cellBalance-1)+2+N1+I(1),2) = 1; % Link the channel with an edge
user
systemPackets(2,3*(cellBalance-1)+2) =
systemPackets(2,3*(cellBalance-1)+2) + 1; % Add a data packet in the system

channelsAvailables(cellBalance,1) = channelsAvailables(cellBalance,1) + 1;
else
I =
find(futureEvents(N*(cellBalance-1)+3:N*(cellBalance-1)+N1+2,1)==inf); % Find a voice
channel available
if
isempty(I)

futureEvents(N*(cellBalance-1)+2+I(1),1) = -(clock + random('exp',1/mu2)); %
Schedule data time service in a voice channel

% available

futureEvents(N*(cellBalance-1)+2+I(1),2) = 1; % Link the channel with an edge user

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        systemPackets(2,3*(cellBalance-1)+2)      =
systemPackets(2,3*(cellBalance-1)+2) + 1; % Add a data packet in the system

    channelsAvailables(cellBalance,1) = channelsAvailables(cellBalance,1) + 1;
                                         end
                                         end
                                         else

    packetsLoss(4,cell_random) = packetsLoss(4,cell_random) + 1;
                                         end
                                         end
                                         end
                                         end
                                         end
                                         end
otherwise % Packet departure

    cell_random = floor((abs(event-3))/N)+1; % Obtain the cell
related to current event
    if event > N*(cell_random-1)+2+N1 % If packet to release is
in a data channel
        if futureEvents(event,1) > 0 % If it is a data packet
in a data channel
            if futureEvents(event,2) == 0 % Normal user
                systemPackets(1,3*(cell_random-1)+2)
= systemPackets(1,3*(cell_random-1)+2) - 1; % data packet was transmitted
                packetsTransmitted(2,cell_random)           =
packetsTransmitted(2,cell_random) + 1;
            else % Edge user
                systemPackets(2,3*(cell_random-1)+2)
= systemPackets(2,3*(cell_random-1)+2) - 1; % data packet was transmitted
                packetsTransmitted(4,cell_random)           =
packetsTransmitted(4,cell_random) + 1;
            end
            else % If it is a voice packet in a data channel
                if futureEvents(event,2) == 0 % Normal user
                    systemPackets(1,3*(cell_random-1)+1)
= systemPackets(1,3*(cell_random-1)+1) - 1; % voice packet was transmitted
                    packetsTransmitted(1,cell_random)           =
packetsTransmitted(1,cell_random) + 1;
                else % Edge user
                    systemPackets(2,3*(cell_random-1)+1)
= systemPackets(2,3*(cell_random-1)+1) - 1; % voice packet was transmitted
                    packetsTransmitted(3,cell_random)           =
packetsTransmitted(3,cell_random) + 1;
                end
                sign = 1;

            else % If packet to release is in a voice channel

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

if futureEvents(event,1) > 0 % If it is a voice packet
in a voice channel

    if futureEvents(event,2) == 0 % Normal user
        systemPackets(1,3*(cell_random-1)+1)
    = systemPackets(1,3*(cell_random-1)+1) - 1; % voice packet was transmitted
        packetsTransmitted(1,cell_random) =
    packetsTransmitted(1,cell_random) + 1;
        else % Edge user
            systemPackets(2,3*(cell_random-1)+1)
    = systemPackets(2,3*(cell_random-1)+1) - 1; % voice packet was transmitted
        packetsTransmitted(3,cell_random) =
    packetsTransmitted(3,cell_random) + 1;
    end

    else % If it is a data packet in a voice channel
        if futureEvents(event,2) == 0 % Normal user
            systemPackets(1,3*(cell_random-1)+2)
    = systemPackets(1,3*(cell_random-1)+2) - 1; % data packet was transmitted
        packetsTransmitted(2,cell_random) =
    packetsTransmitted(2,cell_random) + 1;
        else % Edge user
            systemPackets(2,3*(cell_random-1)+2)
    = systemPackets(2,3*(cell_random-1)+2) - 1; % data packet was transmitted
        packetsTransmitted(4,cell_random) =
    packetsTransmitted(4,cell_random) + 1;
    end

    sign = -1;
end

if systemPackets(1,3*(cell_random-1)+3) > 0 % If exists
packets in a queue (normal users)

    systemPackets(1,3*(cell_random-1)+3) =
systemPackets(1,3*(cell_random-1)+3) - 1; % Remove one packet from queue
    systemPackets(1,3*(cell_random-1)+2)
= systemPackets(1,3*(cell_random-1)+2) + 1; % Add an existing data packet (queue) in system
    futureEvents(event,1) = sign*(clock +
random('exp',1/mu2)); % Schedule next time processing for data packet
    futureEvents(event,2) = 0;

else
    if systemPackets(2,3*(cell_random-1)+3)> 0 % If
exists packets in a queue (edge users)

        systemPackets(2,3*(cell_random-1)+3) =
systemPackets(2,3*(cell_random-1)+3) - 1; % Remove one packet from queue
        systemPackets(2,3*(cell_random-1)+2)
= systemPackets(2,3*(cell_random-1)+2) + 1; % Add an existing data packet (queue) in system
        futureEvents(event,1) = sign*(clock +
random('exp',1/mu2)); % Schedule next time processing for data packet
        futureEvents(event,2) = 1;

else
    futureEvents(event,1) = inf;
    futureEvents(event,2) = inf;

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

channelsAvailables(cell_random,1)      =
channelsAvailables(cell_random,1) - 1;
end
end
end
for i = 1:Nodes
    totalPackets2(1,i) = totalPackets(2*i-1);
    totalPackets2(2,i) = totalPackets(2*i);
end
trace = trace(1:numEvent,:); % Saves the System Occupation
throughput = throughput(1:numEvent,:); % Saves the throughput
%figure
%for i = 1:1
%    plot(trace(:,1),throughput(:,i)); % Draw throughput by cell
%    hold on
%end
throughputAverage = zeros(Nodes,1);
dt = trace(2:end,1)-trace(1:end-1,1);
for i = 1:Nodes
    area = dt.*throughput(2:end,i);
    throughputAverage(i,1) = sum(area)/(0.000066*simulationTime);
end
end

```

A continuación se expone el código utilizado para generar la tabla de rutas:

```

function Z = routing_table(weight_nodes_matrix_input,N) % The input is weights by nodes
and the total channels
    Nodes = 12; % Number of nodes in Cluster
    adjacency_matrix = [0 0 0 0 0 0 0 0 0 0 0 0; % Create an adjacency matrix
                        0 0 0 0 0 0 1 0 0 0 0 0;
                        0 0 0 1 1 0 1 0 0 0 0 0;
                        0 0 1 0 0 0 0 0 0 0 0 0;
                        0 0 0 0 0 0 1 0 1 1 0 0;
                        0 0 0 0 0 0 0 0 0 1 0 0;
                        0 1 1 0 0 0 0 0 0 0 0 0;
                        0 0 0 0 0 0 0 0 0 0 0 0;
                        0 0 0 0 1 0 0 0 0 1 1 0;
                        0 0 0 0 1 1 0 0 1 0 0 0;
                        0 0 0 0 0 0 0 0 1 0 0 0;
                        0 0 0 0 0 0 0 0 0 0 0 1];
    % Define the links between nodes
    C = zeros(Nodes,Nodes);
    % weight_nodes_matrix = zeros(length(weight_nodes_matrix_input),1);
    weight_nodes_matrix = N*ones(12,1) - weight_nodes_matrix_input;
    for i = 1:Nodes
        for j = 1:Nodes

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        if adjacency_matrix(i,j)==1
            %          C(i,j)      =      weight_nodes_matrix(1,j)
            weight_nodes_matrix_input(i);
            C(i,j) = weight_nodes_matrix(j,1) - weight_nodes_matrix(i,1);
            end
        end
    end
% Creation of Routing Table
% Initialize the route table in Time = 0
a = 1;
for i = 1:Nodes
    for j = 1:Nodes
        if adjacency_matrix(i,j)==1
            X(a,1) = i; % Source Node
            X(a,2) = 0; % Time
            X(a,3) = j; % Destination Node
            X(a,4) = j; % Next Hop
            X(a,5) = C(i,j); % Cumulative Available Cost
            a = a+1;
        end
    end
end
c = a;
z = X;
for t = 1:1 % N iterations, in this case will be 1 more (2nd order)
    Y = X;
    for i = 1:Nodes
        for j = 1:Nodes
            if adjacency_matrix(i,j)==1
                % I = find(Y(:,1)==j);
                A = Y(find(Y(:,1)==j),:);
                [rows,~] = size(A);
                for m = 1:rows
                    X(a,1) = i;
                    X(a,2) = 1;
                    X(a,3) = A(m,3);
                    X(a,4) = j;
                    X(a,5) = A(m,5) + C(i,j);
                    a = a + 1;
                end
            end
        end
    end
    D = Z(find(Z(:,1)==i),:);
    for j = 1:Nodes
        if isempty(find(D(:,3)==j))==1
            W = X(find(X(:,1)==i),:);
            W = W(find(W(:,3)==j),:);

```

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

```

        if(isempty(W)==0 && i~=j)
            maxW = max(W(:,5));
            b = find(W(:,5)==maxW);
            Z(c,:) = W(b(1),:); % In case the result includes
more than 1 route.
            c = c+1;
        end
    end
end
end
end

```

A continuación se expone el código utilizado para correr el Balanceo de carga:

```

function [finalUsermatrix, orderLB, cellBalance] = runLB2(usersMatrix, sourceCell,
channelsAvailables, N, antennaCell, balance)

    % Input parameters: Users matrix, Source cell, Channel availables (Vector that
contains the load info by cell), N is the total number of channels (RB) by cell
    % Output parameters: finalUsermatrix (Users matrix updated), success LB (indicates
if LB was run or not), orderLB indicates the LB order

    orderLB = 0; % Indicates the LB order (0: LB not applied, 1: LB - first order
applied, 2: LB - second order applied)
    routeTable = routing_table(channelsAvailables, N); % Calculate the current Routing
Table
    partialRouteTable = routeTable(find(routeTable(:,1)==sourceCell),:); % Find the
records with source node equal to source cell
    partialDest = partialRouteTable(find(partialRouteTable(:,2)==0),:); % Calculate
first the neighbors with time = 0
    if ~isempty(partialDest)
        [partialMax,partialMaxPosition] = max(partialDest(:,5)); % Find the cell
with more resources to balance
        if partialMax > 0 && balance > 0
            orderLB = 1;
            cellBalance = partialDest(partialMaxPosition,3); % Find the
neighbor cell to balance
            finalUsermatrix = LoadBalancing(usersMatrix, sourceCell,
cellBalance, antennaCell); % Apply the Load Balancing
        else
            partialDest =
partialRouteTable(find(partialRouteTable(:,2)==1),:); % Calculate first the neighbors with
time = 1
            if ~isempty(partialDest)
                [partialMax,partialMaxPosition] = max(partialDest(:,5)); %
Find the cell with more resources to balance
                if partialMax > 0 && balance == 2
                    orderLB = 2;
                    cellIntermediate =
partialDest(partialMaxPosition,4); % Find the neighbor cell to balance (intermediate)
                end
            end
        end
    end

```

```
cellBalance = partialDest(partialMaxPosition,3); %  
Find the neighbor cell to balance  
finalUsermatrix = LoadBalancing(usersMatrix,  
sourceCell, cellIntermediate, antennaCell); % Apply the Load Balancing  
finalUsermatrix = LoadBalancing(finalUsermatrix,  
cellIntermediate, cellBalance, antennaCell); % Apply the Load Balancing  
end  
end  
end  
if orderLB == 0  
    finalUsermatrix = usersMatrix;  
    cellBalance = 0;  
end  
end
```

E. Aproximación patrón radiación Sistemas Radiantes

A partir del patrón de radiación de la figura 50a, se propone la siguiente linealización (línea roja) expuesta en la figura 91.

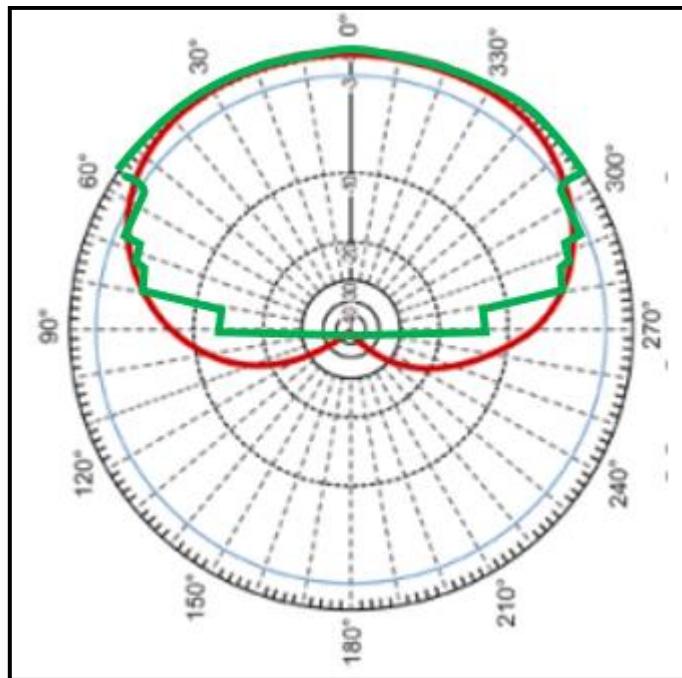


Figura 91. Aproximación del patrón de radiación (plano horizontal) de la figura 50a.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

Como se observa en la figura 91 se han suprimido los lóbulos traseros y se realizó una aproximación del plano horizontal entre los +/- 90°. A continuación se expone el código empleado para aproximar el plano horizontal de radiación de cada sistema radiante:

```
function angle = angleFunction(x1,y1,x2,y2,reference)

if (x2-x1)>=0 & (y2-y1)>=0
    angle = atan2((x2-x1)/(y2-y1)) - reference;
end

if (x2-x1)<0 & (y2-y1)>=0
    angle = atan2((x2-x1)/(y2-y1)) - reference + 360;
    if angle > 300
        angle = angle - 360;
    end
end

if (x2-x1)>=0 & (y2-y1)<0
    angle = atan2((x2-x1)/(y2-y1)) - reference + 180;
end

if (x2-x1)<0 & (y2-y1)<0
    angle = atan2((x2-x1)/(y2-y1)) - reference + 180;
end

end

function loss = lossFunction(tetha)

if abs(tetha)<=60
    loss = 0
end

if abs(tetha)>60 & abs(tetha)<=70
    loss = 3
end

if abs(tetha)>70 & abs(tetha)<=75
    loss = 5
end

if abs(tetha)>75 & abs(tetha)<=80
    loss = 10
end

if abs(tetha)>80 & abs(tetha)<=90
    loss = 20
end
```

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

```
if abs(tetha)>90
    loss = 100
end

end
```

Después de implementar el código anterior se puede obtener el mapa de cobertura (figura 50b) a través de la medición del RSRP, mediante la siguiente relación:

$$RSRP = RSSI - 10 * \log(12 * N) \quad (67)$$

Donde el RSSI se obtiene mediante la ecuación 41.

F. Cálculo de la tasa de pérdidas

Como se evidencia en el código expuesto en el apéndice D, por cada evento dentro de la simulación (llegada o salida de un paquete del sistema) se ejecuta un ciclo en el que se evalúa el evento actual. Si el evento es una llegada de paquete y todos los canales de la celda están ocupados y no es factible realizar balanceo de carga debido a que las celdas vecinas también lo están el contador de paquetes perdidos se incrementa en 1.

La variable que almacenará los paquetes perdidos durante el periodo de simulación será **packetsLoss** que será una matriz de 4 filas y 12 columnas (una por cada celda). En la primera fila se almacenará la cantidad de paquetes de voz perdidos para usuarios cercanos a la celda, en la segunda fila se almacenará la cantidad de paquetes de datos perdidos para usuarios cercanos a la celda, en la tercera fila se almacenará la cantidad de paquetes de voz perdidos para usuarios de borde de celda y en la cuarta fila se almacenará la cantidad de paquetes de datos perdidos para usuarios de borde de la celda.

Después de transurrido el tiempo de simulacióm n la tasa de pérdidas se calcula de la siguiente forma:

$$\text{Tasa de perdida paquetes voz} = \frac{\# \text{ Paquetes voz perd. usuarios cerca de celda} + \# \text{ Paquetes voz perd. usuarios borde de celda}}{\# \text{ Total de paquetes de voz entrantes al sistema}}$$

$$\text{Tasa de perdida paquetes datos} = \frac{\# \text{ Paquetes datos perd. usuarios cerca de celda} + \# \text{ Paquetes datos perd. usuarios borde de celda}}{\# \text{ Total de paquetes de voz entrantes al sistema}}$$

G. Cálculo del Throughput promedio por celda

Como se mencionó en el apéndice F, por cada evento dentro de la simulación se ejecuta un ciclo en el que se evalúa el evento actual. Para calcular el throughput promedio por celda se creará inicialmente una variable llamada **throughput** que será una matriz de 12 columnas (una por cada celda) y de m filas, donde m será la cantidad de eventos durante la simulación. Es decir que desde el instante t=0 hasta t=Tiempo_simulación la variable **throughput** almacenará el throughput por celda en todos los instantes de simulación.

Para cada ciclo de simulación el throughput se calculará de la siguiente forma:

$$\begin{aligned} \text{Throughput} = & 12 \text{ subportadoras} * BW_{\text{subportadora}} * (\# \text{ paquetes usuarios cerca celda}) \\ & * (\text{bits por modulación usuarios cerca celda}) + 12 \text{ subportadoras} * BW_{\text{subportadora}} \\ & * (\# \text{ paquetes usuarios borde celda}) * (\text{bits por modulación usuarios borde celda}) \end{aligned}$$

Donde para un esquema de 64QAM (usuarios cerca de celda) se transmiten 6 bits por símbolo y para un esquema de 16QAM (usuarios de borde celda) se transmiten 4 bits por símbolo.

Finalmente, cuando ha transcurrido el tiempo de simulación el throughput promedio por celda se halla sumando todos los valores de la variable **throughput** por celda y diviendo sobre el tiempo total de simulación de la siguiente manera:

$$\text{Throughput}_{\text{celda}} = \frac{1}{\text{Tiempo}_{\text{simulación}}} * \sum_{i=0}^m \text{Throughput}(i, \text{celda})$$

H. Cálculo del número de handovers de balanceo

Para el caso del cálculo de handovers ejecutados durante el tiempo de simulación el cálculo es muy sencillo. Se utilizará una variable llamada **handovers** que será una matriz de 12 filas (una por cada celda) y 2 columnas (para identificar los balanceos de primer y segundo orden). A medida que los eventos se ejecutan y se activan los procesos de Balanceo de carga la variable **handovers** va incrementando. Al final de la simulación la variable **handovers** arrojará el número total de handovers ejecutados por celda.

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

I. Tablas de Resultados

Escenario $\lambda_{voice} = 10$ y $\lambda_{voice} = 10$.

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
0		0						

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	0	0.666	0.52	0.808	0	0.377	0	0.516
2	0.41	0.575	1.109	0	0.686	0	0.883	0
3	0.539	0.678	0.538	0	0.485	0	0.544	0
4	0.556	0.522	0.518	0	0.884	0	0.664	0
5	0.758	0.703	0.53	0	0.593	0	0.435	0
6	0.733	0.691	0.692	0	0.711	0	0.534	0
7	0.585	0.461	0.438	0	0.377	0	0.68	0
8	0.504	0.595	0.518	0	0.501	0	0.835	0
9	0.528	0.546	0.492	0	0.948	0	0.34	0
10	0.482	0.526	0.582	0	0.34	0	0.659	0
11	0.518	0.609	0.624	0.006	0.421	0	0.447	0
12	0.357	0.467	0.413	0	0.744	0	0.643	0
5.97		7.04	6.98	8.1	6.70	0.38	6.70	0.52

celda	Throughput (Mbps)								
	I = 0		I = 5		I = 10		I = inf		
celda	I = 0	I = 5	I = 10	I = inf	I = 0	I = 5	I = 10	I = inf	
1	161480	188200	169480	177860	2	172580	160200	161240	155120
3	159870	155510	174880	160520	4	173070	160820	163040	150000
5	174900	168880	173260	160730	6	158340	157190	143880	170820
7	159560	155300	159460	180880	8	169140	157360	175530	149000
9	157100	158920	143390	168680	10	163060	160800	161410	157380
11	152610	161200	158490	167940	12	163500	174080	147420	181970
19.65210		19.58550	19.41480	19.60900					

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	40	0	39.5	0	38.5	0	36.5	0
3	117	0	57	0	61	0	55.5	0
4	75	0	32.5	0	33	0	29	0
5	73	0	32.5	0	57	0	36	0
6	42	0	39.5	0	36	0	27.5	0
7	99	0	64	0	60.5	0	51.5	0
8	0	0	0	0	0	0	0	0
9	110	0	58	0	54.5	0	62	0
10	118.5	0	86	0	91.5	0	60.5	0
11	44.5	0	20.5	0	27	0	28	0
12	0	0	0	0	0	0	0	0
719		0	430	0	459	0	387	0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	0.603	0.578	0.46	0	0.769	0	0.8	0
2	0.395	0.249	0.326	0	0.314	0	0.23	0
3	0.388	0.48	0.382	0	0.483	0	0.38	0
4	0.394	0.417	0.38	0	0.343	0	0.359	0
5	0.351	0.365	0.248	0	0.431	0	0.344	0
6	0.263	0.239	0.383	0	0.334	0	0.237	0
7	0.443	0.364	0.297	0	0.472	0	0.319	0
8	0.613	0.737	0.645	0	0.43	0	0.624	0
9	0.383	0.452	0.468	0	0.404	0	0.366	0
10	0.455	0.438	0.6	0	0.501	0	0.554	0
11	0.28	0.277	0.194	0	0.23	0	0.363	0
12	0.731	0.653	0.638	0	0.558	0	0.568	0
5.30		5.25	5.02	0.00	5.33	0.00	5.20	0.00

celda	Throughput (Mbps)								
	I = 0		I = 5		I = 10		I = inf		
celda	I = 0	I = 5	I = 10	I = inf	I = 0	I = 5	I = 10	I = inf	
1	158290	153430	168470	174180	2	152150	164390	159620	154870
3	166750	163180	169380	164000	4	162320	149550	154900	166610
5	168320	163210	177150	170900	6	164020	159110	158380	150000
7	175410	165960	171510	175150	8	166770	163600	162820	165280
9	176170	173510	166730	163460	10	163480	182140	180370	175500
11	153150	141160	150280	153780	12	170380	169300	151330	164260
19.56710		19.59540	19.73540	19.66620					

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	445.5	0	96.5	0	111.5	0	97.5	0
3	665.5	0	186.5	0	116	0	116	0
4	373.5	0	96	0	72.5	0	72.5	0
5	522.5	0	188	0	112.5	0	153.5	0
6	293	0	101.5	0	72	0	85	0
7	894.5	0	158	0	192	0	185.5	0
8	0	0	0	0	0	0	0	0
9	915	0	185.5	0	201	0	189.5	0
10	680.5	0	202	0	152.5	0	226	0
11	359.5	0	79.5	0	74.5	0	77	0
12	0	0	0	0	0	0	0	0
5097		0	1294	0	1105	0	1213	0

celda	Tasa de Pérdidas (%)							
I = 0		I = 5		I = 10		I = inf		
celda	voice	data	voice	data	voice	data	voice	data

<tbl_r cells="2" ix="1" maxcspan="8" maxrspan="2" usedcols="

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Escenario $\lambda_{voice} = 10$ y $\lambda_{voice} = 30$.

Handovers												
celda	I = 0		I = 5		I = 10		I = inf		I = 0		I = 2nd	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0

Tasa de Pérdidas (%)												
celda	I = 0		I = 5		I = 10		I = inf		celda		Throughput (Mbps)	
	voice	data	voice	data	voice	data	voice	data	voice	data	I = 0	I = 5
1	1.203	4.44	2.64	0.82	1.85	0.01	1.97	0	1	2.78320	3.53370	3.05630
2	1.377	4.832	1.33	0.16	1.03	0	1.803	0	2	2.74420	2.85300	2.66200
3	1.21	4.51	1.29	0.16	2.449	0.036	1.941	0	3	2.80900	2.89750	3.33730
4	1.568	4.378	1.71	0.091	1.96	0	1.968	0	4	2.74340	2.88770	3.01220
5	1.32	3.801	1.73	0.262	1.68	0	1.381	0	5	2.60090	3.00570	2.83570
6	1.051	3.842	1.52	0.103	2.36	0.005	1.263	0	6	2.63710	2.73140	3.33910
7	1.482	4.961	1.28	0.061	1.443	0.005	1.91	0	7	2.83670	2.78360	2.91940
8	1.569	5.804	1.06	0.11	1.703	0	1.477	0	8	2.9050	2.82570	3.06450
9	1.742	5.185	1.512	0.125	1.394	0.005	2.148	0	9	2.82370	2.91850	2.96630
10	0.999	4.234	1.786	0.375	1.032	0	1.461	0	10	2.69260	3.31940	2.63390
11	0.779	3.202	1.099	0.085	1.372	0	1.64	0	11	2.54160	2.79250	3.06190
12	2.008	6.303	2.002	0.142	1.393	0.019	1.523	0	12	3.0150	3.32850	3.12590
	16.31	55.29	18.12	2.44	19.05	0.08	18.14	0.00		33.14100	35.84820	36.02110
	16.31	55.29	18.12	2.44	19.05	0.08	18.14	0.00		33.14100	35.84820	36.02110
	16.31	55.29	18.12	2.44	19.05	0.08	18.14	0.00		33.14100	35.84820	36.02110

Handovers												
celda	I = 0		I = 5		I = 10		I = inf		I = 0		I = 2nd	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0	0	0	0	0
2	1538	0	186	0	200	0	106	0	0	0	0	0
3	2683	0	353	0	287	0	257	0	0	0	0	0
4	1308	0	177	0	130	0	133	0	0	0	0	0
5	1716	0	293	0	246	0	241	0	0	0	0	0
6	1141	0	196	0	126	0	140	0	0	0	0	0
7	2839	0	538	0	327	0	238	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	2889	0	363	0	387	0	252	0	0	0	0	0
10	3337	0	471	0	262	0	262	0	0	0	0	0
11	1134	0	163	0	143	0	132	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0
	18585	0	2740	0	2108	0	1761	0	0	0	0	0

Tasa de Pérdidas (%)												
celda	I = 0		I = 5		I = 10		I = inf		celda		Throughput (Mbps)	
	voice	data	voice	data	voice	data	voice	data	voice	data	I = 0	I = 5
1	1738	6.304	2.933	0.554	1.313	0	2.821	0	1	3.04310	3.58070	2.79100
2	1441	4.033	0.803	0.061	1.11	0.008	0.807	0	2	3.14830	3.1190	3.27780
3	1092	3.188	0.767	0.08	0.652	0.031	1.071	0	3	3.29440	3.31840	3.38830
4	104	3.033	0.743	0.132	0.836	0	0.672	0	4	3.28070	3.08270	2.99210
5	0.896	2.395	0.886	0.07	0.769	0.008	0.958	0	5	3.06280	3.44120	3.54590
6	0.859	2.904	0.999	0.116	0.866	0.005	0.792	0	6	2.97010	3.08830	2.79040
7	1.097	3.256	0.859	0.179	0.955	0.008	0.525	0	7	3.31800	3.51780	3.36820
8	1.653	6.006	1.255	0.666	2.593	0.008	2.967	0	8	2.94550	2.91820	3.42240
9	1.163	3.083	0.854	0.078	0.883	0.018	0.709	0	9	3.31820	3.26790	3.64710
10	1.409	3.511	0.878	0.064	0.819	0.004	0.685	0	10	3.49060	4.00960	3.17080
11	0.876	2.795	1.073	0.093	0.938	0.018	1.08	0	11	2.94170	3.12380	3.12510
12	1.582	5.77	3.001	0.421	3.08	0.064	1.852	0	12	2.84870	3.52320	3.85640
	14.64	46.88	14.32	1.31	14.56	0.17	14.92	0.00		37.36210	39.27480	39.16450
	14.64	46.88	14.32	1.31	14.56	0.17	14.92	0.00		37.36210	39.27480	39.16450
	14.64	46.88	14.32	1.31	14.56	0.17	14.92	0.00		37.36210	39.27480	39.16450

Handovers												
celda	I = 0		I = 5		I = 10		I = inf		celda		Throughput (Mbps)	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0	0	0	0	0
2	2128	0	1062	0	473	0	365	0	0	0	0	0
3	4652	0	2577	0	676	0	574	0	0	0	0	0
4	2243	0	1178	0	369	0	319	0	0	0	0	0
5	4717	0	2320	0	953	0	449	0	0	0	0	0
6	4717	0	2320	0	953	0	449	0	0	0	0	0
7	5963	0	2337	0	1316	0	743	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	5193	0	1926	0	115	0	994	0	0	0	0	0
10	5282	0	1662	0	124	0	725	0	0	0	0	0
11	2175	0	633	0	430	0	375	0	0</td			

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Escenario $\lambda_{voice} = 20$ y $\lambda_{data} = 10$.

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	33.0	16.6	37.3	0.0	37.0	0.0	35.4	0.0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	3.172	1725	2.398	0	2.446	0	3.404	0
2	1.92	1036	4.416	0	3.426	0	1.61	0
3	2.135	0.984	4.553	0	3.534	0	1.785	0
4	3.596	1849	3.52	0	3.744	0	2.257	0
5	2.973	1471	3.042	0	4.146	0	3.876	0
6	2.994	1353	2.719	0	3.082	0	2.738	0
7	3.171	1641	2.178	0	1.804	0	2.334	0
8	2.089	1302	2.61	0	1.913	0	4.917	0
9	2.16	1.037	2.956	0	3.334	0	3.455	0
10	2.758	1155	3.379	0	2.57	0	3.208	0
11	2.876	181	2.106	0	3.245	0	2.623	0
12	3.179	1369	3.415	0	3.123	0	3.172	0
	33.0	16.6	37.3	0.0	37.0	0.0	35.4	0.0

celda	Throughput (Mbps)							
	I = 0		I = 5		I = 10		I = inf	
celda	I = 0	I = 5	I = 10	I = inf				
1	2.48780	2.32080	2.34030	2.48840				
2	2.16700	2.69370	2.39580	2.06820				
3	2.18280	2.66570	2.44870	2.10590				
4	2.48460	2.46640	2.54560	2.25100				
5	2.44930	2.42880	2.60230	2.45270				
6	2.45780	2.25950	2.38020	2.43780				
7	2.41490	2.12350	2.07660	2.25090				
8	2.25030	2.32300	2.19370	2.71590				
9	2.18940	2.41480	2.51650	2.44510				
10	2.35660	2.42710	2.32270	2.46710				
11	2.39580	2.13760	2.45180	2.25310				
12	2.41260	2.45030	2.41630	2.42750				
	28.26190	28.72820	28.72110	28.46560				

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	223	0	167	0	291	0	150	0
3	366	0	325	0	407	0	245	0
4	222	0	141	0	222	0	166	0
5	254	0	295	0	209	0	303	0
6	132	0	192	0	212	0	182	0
7	368	0	274	0	465	0	421	0
8	0	0	0	0	0	0	0	0
9	424	0	441	0	268	0	259	0
10	371	0	419	0	326	0	413	0
11	168	0	200	0	107	0	122	0
12	0	0	0	0	0	0	0	0
	2528	0	2454	0	2507	0	2261	0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	2.778	1533	2.486	0	3.739	0	2.237	0
2	1.489	971	1.17	0	1.885	0	1.349	0
3	1.746	1221	1.806	0	1.924	0	1.333	0
4	1.533	6388	1.585	0	1.782	0	1.211	0
5	1.37	1.903	1.623	0	1.047	0	1.421	0
6	0.982	0.655	1.416	0	1.762	0	1.261	0
7	1.538	1.982	1.696	0	1.618	0	2.08	0
8	2.944	1.888	2.268	0	2.123	0	2.282	0
9	1.801	0.9	1.482	0	1.316	0	1.46	0
10	1.705	0.796	1.469	0	1.337	0	1.469	0
11	1.197	0.935	1.602	0	0.932	0	0.95	0
12	2.182	1.896	2.672	0	2.311	0	5.398	0
	21.3	12.6	21.3	0.0	21.8	0.0	22.6	0.0

celda	Throughput (Mbps)							
	I = 0		I = 5		I = 10		I = inf	
celda	I = 0	I = 5	I = 10	I = inf				
1	2.38570	2.26540	2.48700	2.19760				
2	2.44020	2.32170	2.51580	2.30690				
3	2.50690	2.61670	2.63030	2.41250				
4	2.30130	2.26420	2.48420	2.30700				
5	2.47620	2.54510	2.53540	2.58930				
6	2.17830	2.33330	2.47360	2.23190				
7	2.50000	2.47190	2.57490	2.73490				
8	2.39330	2.20000	2.22970	2.23960				
9	2.56570	2.50240	2.43230	2.45930				
10	2.53200	2.61680	2.48590	2.59610				
11	2.27270	2.41440	2.11400	2.22870				
12	2.28620	2.28490	2.29860	2.77560				
	28.61850	28.83870	29.08120	29.13930				

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	972	0	571	0	758	0	791	0
3	1726	0	1306	0	887	0	1813	0
4	1044	0	795	0	454	0	874	0
5	2017	0	879	0	1254	0	874	0
6	1186	0	716	0	489	0	651	0
7	2262	0	124	0	1771	0	1681	0
8	0	0	0	0	0	0	0	0
9	2630	0	1357	0	1822	0	1283	0
10	2929	0	1761	0	1353	0	1476	0
11	1180	0	542	0	654	0	539	0
12	0	0	0	0	0	0	0	0
	16009	0	9141	0	9442	0	9982	0

|
<th rowspan="
| |

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Escenario $\lambda_{voice} = 20$ y $\lambda_{data} = 30$.

Handovers									
celda	I = 0		I = 5		I = 10		I = inf		I = 0
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Tasa de Pérdidas (%)									
celda	I = 0		I = 5		I = 10		I = inf		I = 0
	voice	data	voice	data	voice	data	voice	data	
1	7.96	11.67	7.754	0.535	9.255	0.121	9.379	0	1
2	6.82	10.01	8.57	0.483	8.186	0	7.802	0	2
3	8.16	11.95	10.66	1.311	10.94	0.3	8.327	0	3
4	6.667	9.721	7.142	0.391	10.03	0.058	8.647	0	4
5	6.162	9.495	10.77	1.092	8.539	0.025	9.291	0	5
6	6.541	10.32	8.37	0.67	8.186	0.027	7.67	0	6
7	7.231	11.72	8.949	0.834	8.785	0.03	8.764	0	7
8	8.736	12.76	8.842	0.436	7.394	0.005	8.866	0	8
9	9.789	13.99	10.68	1.194	7.905	0.021	10.07	0	9
10	8.397	11.54	5.944	0.158	9.383	0.043	6.536	0	10
11	5.413	8.056	10.85	1.067	7.007	0.01	9.638	0	11
12	9.33	13.3	7.577	0.52	9.1	0.064	12.41	0	12
	9.11	134.7	10.54	8.5	104.4	0.7	105.5	0.0	

Throughput (Mbps)									
celda	I = 0		I = 5		I = 10		I = inf		I = 0
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	
1	3.55940	3.72200	4.02360	4.27010	4.31960	4.37810	4.36750	4.66040	1
2	3.41960	3.73170	4.36490	4.38210	4.35620	4.35870	4.36040	4.35870	2
3	3.55430	4.16040	4.35620	4.35870	4.36040	4.35870	4.36040	4.35870	3
4	3.33250	3.60220	4.05250	4.05250	4.05250	4.05250	4.05250	4.05250	4
5	3.33360	4.17490	4.36430	4.36430	4.36430	4.36430	4.36430	4.36430	5
6	3.36520	3.84050	3.78260	3.78260	3.78260	3.78260	3.78260	3.78260	6
7	3.49330	3.96000	3.94290	3.94290	3.94290	3.94290	3.94290	3.94290	7
8	3.68760	4.13680	3.68410	4.20980	4.20980	4.20980	4.20980	4.20980	8
9	3.56130	3.35840	3.94560	3.58930	3.58930	3.58930	3.58930	3.58930	9
10	3.20670	4.14280	3.60600	4.02530	4.02530	4.02530	4.02530	4.02530	10
11	3.71200	3.70080	3.91280	4.47020	4.47020	4.47020	4.47020	4.47020	11
12	4.19640	46.36530	46.72750	46.73020	46.73020	46.73020	46.73020	46.73020	12
	41.96400	46.36530	46.72750	46.73020	46.73020	46.73020	46.73020	46.73020	

Handovers									
celda	I = 0		I = 5		I = 10		I = inf		I = 0
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	
1	0	0	0	0	0	0	0	0	0
2	1958	0	356	0	856	0	722	0	2
3	3950	0	1923	0	1430	0	1779	0	3
4	1868	0	635	0	590	0	643	0	4
5	4725	0	1512	0	1794	0	890	0	5
6	1873	0	812	0	586	0	665	0	6
7	4112	0	2021	0	1647	0	2280	0	7
8	0	0	0	0	0	0	0	0	8
9	5925	0	2012	0	1834	0	1617	0	9
10	4215	0	1389	0	1524	0	1929	0	10
11	2397	0	714	0	555	0	659	0	11
12	0	0	0	0	0	0	0	0	12
	31023	0	11979	0	10616	0	11164	0	

Tasa de Pérdidas (%)									
celda	I = 0		I = 5		I = 10		I = inf		I = 0
	voice	data	voice	data	voice	data	voice	data	
1	6.137	9.537	7.676	0.582	8.059	0	7.754	0	1
2	4.941	7.324	6.304	0.534	4.521	0.005	6.51	0	2
3	4.223	6.44	3.828	0.23	3.521	0.005	4.143	0	3
4	4.94	7.266	5.04	0.179	4.098	0	5.058	0	4
5	5.157	7.984	3.457	0.234	3.751	0.027	4.411	0	5
6	4.101	6.92	5.275	0.228	4.31	0.023	5.234	0	6
7	4.327	6.925	3.918	0.299	3.48	0.029	4.859	0	7
8	5.731	8.614	9.56	0.653	11.12	0.030	10.36	0	8
9	5.263	7.792	3.783	0.31	3.799	0.023	3.298	0	9
10	4.423	6.148	2.96	0.195	4.356	0.002	3.87	0	10
11	6.568	9.867	5.151	0.334	4.088	0	4.425	0	11
12	8.753	13.27	7.982	0.701	10.26	0.19	8.813	0	12
	64.6	98.0	65.5	4.5	64.5	0.6	68.8	0.0	

Throughput (Mbps)									
celda	I = 0		I = 5		I = 10		I = inf		I = 0
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	
1	3.41510	3.88010	3.44250	4.02390	3.61230	4.31960	3.62250	4.32180	1
2	3.66250	3.93810	3.95750	3.62250	3.62250	4.04240	4.07830	4.15540	2
3	3.89890	4.02410	4.07830	4.15540	4.15540	4.32150	4.37230	4.388240	3
4	4.32170	4.51360	5.00890	4.80930	4.80930	4.05450	4.45020	4.79610	4
5	3.84200	4.21310	4.27470	4.27470	4.27470	4.27470	4.27470	4.30730	5
6	3.76810	3.83200	4.25950	4.25950	4.25950	4.25950	4.25950	4.25950	6
7	3.96460	4.39630	4.36900	4.36900	4.36900	4.36900	4.36900	4.36900	7
8	3.36340	4.19330	3.66080	3.71440	3.66080	4.26390	4.42760	4.43760	8
9	3.84250	4.04000	4.26390	4.26390	4.26390	4.26390	4.26390	4.26390	9
10	3.90730	4.36400	4.64020	5.26930	4.64020	4.64020	4.64020	4.64020	10
11	3.84490	4.58600	4.57230	5.18130	4.57230	4.57230	4.57230	4.57230	11</

MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE REDES LTE

Escenario $\lambda_{voice} = 30$ y $\lambda_{voice} = 10$.

Handovers											
I = 0			I = 5			I = 10			I = inf		
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	
1	celda	0	0	0	0	0	0	0	0	0	0
2	1524	0	1105	0	1461	0	1816	0			
3	2653	0	2108	0	3173	0	2315	0			
4	1248	0	1157	0	1523	0	1322	0			
5	2541	0	2192	0	1286	0	2142	0			
6	851	0	1424	0	1126	0	397	0			
7	2795	0	2423	0	3269	0	3109	0			
8	0	0	0	0	0	0	0	0			
9	3007	0	3238	0	2317	0	3734	0			
10	2313	0	3340	0	3234	0	2606	0			
11	1624	0	1247	0	1172	0	1271	0			
12	0	0	0	0	0	0	0	0			
18546	0	18234	0	18561	0	19319	0				

Handovers									
	I = 0		I = 5		I = 10		I = inf		
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd	
1	0	0	0	0	0	0	0	0	0
2	1681	347	1079	147	1213	161	1413	239	
3	2894	24	2324	10	2374	15	2832	12	
4	1244	209	1023	159	1368	233	1057	152	
5	1896	15	2144	23	2682	24	1942	14	
6	1034	121	1203	136	1648	300	1232	229	
7	4321	176	2620	92	2302	48	2388	74	
8	0	0	0	0	0	0	0	0	
9	2566	3	3231	21	3060	19	3133	12	
10	2479	10	2469	16	3244	28	3331	24	
11	939	118	1115	174	954	151	126	204	
12	0	0	0	0	0	0	0	0	
19044		1023	1728	778	18845	979	18544	360	

celda	Tasa de Pérdidas (%)					i = inf		
	I = 0	I = 5	I = 10	I = 15	I = 20			
	voice	data	voice	data	voice	data		
1	6.074	17.5	5.947	10.013	7.693	0	7.746	0
2	8.339	2.457	10.51	0	7.559	0	6.242	0
3	8.093	2.646	6.197	0	7.188	0	10.02	0
4	2.761	2.101	6.00	0	7.572	0	4.116	0
5	4.124	1.954	6.00	0	7.444	0	4.000	0
6	7.801	2.339	5.644	0	8.762	0	5.292	0
7	7.807	1.859	5.685	0	4.389	0	6.541	0
8	8.859	2.847	9.7	0	4.454	0	7.76	0
9	5.83	1.861	6.239	0	7.078	0	6.393	0
10	8.512	2.636	9.022	0	5.821	0	5.916	0
11	4.13	1.08	7.961	0	7.859	0	5.574	0
12	5.608	17.65	5.155	0	7.771	0	8.748	0
	82.6	24.5	84.3	0.0	82.9	0.0	84.0	0.0

Tasa de Pérdidas (%)							
celda	I = 0		I = 5		I = 10		I = inf
	voice	data	voice	data	voice	data	
1	8.128	2.247	5.537	0	7.813	0	4.831
2	3.899	1.449	3.041	0	3.572	0	4.369
3	3.131	1.167	2.702	0	3.753	0	3.05
4	2.733	1.009	2.881	0	4.319	0	3.597
5	3.471	1.371	2.971	0	2.205	0	2.941
6	2.354	0.883	3.451	0	2.931	0	2.575
7	2.937	1.24	2.841	0	3.355	0	3.368
8	4.733	1.55	7.088	0	6.78	0	5.345
9	3.025	1.301	3.304	0	2.504	0	3.886
10	2.82	0.991	3.265	0	3.473	0	2.935
11	4.259	1.385	3.153	0	3.096	0	3.295
12	2.704	2.217	7.555	0	5.11	0	6.807
48.7		16.4	47.8	0.0	48.9	0.0	47.0

	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	7.968	2.764	7.027	0	6.804	0	8.238	0
2	3.894	1.179	2.276	0	2.311	0	3.08	0
3	3.398	0.993	2.924	0	2.781	0	3.245	0
4	2.469	118	2.25	0.013	2.713	0	2.094	0
5	2.75	0.919	2.599	0	3.248	0	2.781	0
6	2.037	0.826	2.302	0	2.818	0	2.367	0
7	4.278	1.71	2.639	0	2.825	0	3.103	0
8	5.994	1749	7.591	0	7.03	0	3.971	0
9	2.742	0.909	3.319	0	3.508	0	2.95	0
10	2.658	0.92	2.697	0	2.8	0	3.925	0
11	2.02	0.832	2.087	0	1.942	0	2.571	0
12	5.732	1.736	9.48	0.5	5.724	0	6.751	0
	45.7	15.7	47.2	0.0	44.5	0.0	44.7	0.0

celda	Throughput (Mbps)				
	I = 0	I = 5	I = 10	I = inf	
1	2.9160	2.9260	3.1510	3.1780	
2	3.1940	3.0450	3.1240	2.9500	
3	3.1570	2.9230	3.0680	3.0400	
4	3.0510	2.9230	3.0680	3.0400	
5	3.0400	2.9140	3.0610	3.1200	
6	3.0200	2.80750	3.21060	2.79640	
7	3.0780	2.86130	2.67110	2.96190	
8	2.24060	3.21680	2.75620	3.10620	
9	3.29420	3.27570	3.05690	3.05430	
10	3.24190	3.19330	2.89120	2.93190	
11	2.66507	3.14480	3.17520	3.00680	
12	2.9160	2.79530	3.17200	3.23300	

celda	Throughput (Mbps)			
	I = 0	I = 5	I = 10	I = inf
1	3.14680	2.83720	3.15820	2.76270
2	3.95100	3.00170	3.07210	3.30150
3	3.26390	3.13440	3.49420	3.27420
4	2.93240	2.95830	3.21980	3.12550
5	3.41070	3.30960	3.01680	3.31650
6	2.81380	3.07450	2.90890	2.73270
7	3.29160	3.25730	3.40810	3.38810
8	2.70660	3.06110	2.93410	2.83210
9	3.36090	3.43130	3.17490	3.58980
10	3.20500	3.36560	3.44000	3.29570
11	3.30240	3.02780	3.04820	3.04820
12	3.08310	3.10500	2.76100	3.0160

celda	Throughput (Mbps)			
	I = 0	I = 5	I = 10	I = inf
1	3.17280	3.03650	3.04660	3.20780
2	3.27980	2.83990	2.98160	3.21640
3	3.38320	3.24010	3.20000	3.37030
4	3.00850	2.94080	3.16220	2.91780
5	3.18530	3.26790	3.44770	3.23270
6	2.82960	2.99870	3.19700	2.98630
7	3.67380	3.23000	3.21440	3.31730
8	2.95190	3.09030	3.06420	2.61220
9	3.19980	3.38000	3.40100	3.41600
10	3.19090	3.19430	3.31800	3.48860
11	2.75560	2.91760	2.83990	3.11000
12	2.89760	3.29250	2.89540	2.98680
37.52880				
37.42860				
37.74730				
37.86120				

Escenario $\lambda_{voice} = 30$ y $\lambda_{voice} = 20$.

	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	2018	0	1536	0	1567	0	1560	0
3	3618	0	3596	0	3095	0	3551	0
4	1818	0	1468	0	1558	0	1411	0
5	3945	0	3267	0	3722	0	3140	0
6	1438	0	1336	0	1375	0	1396	0
7	4289	0	4769	0	3514	0	3526	0
8	0	0	0	0	0	0	0	0
9	4250	0	4269	0	3922	0	4052	0
10	5348	0	3264	0	3381	0	3623	0
11	1616	0	1326	0	1283	0	1327	0
12	0	0	0	0	0	0	0	0
	28240	0	42821	0	23417	0	23788	0

celda	I = 0		I = 5		I = 10		I = inf	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	1813	635	1352	394	1389	504	1557	582
3	4788	154	3969	124	3181	94	2985	85
4	2160	774	1612	594	1506	565	1180	361
5	3316	128	3631	117	2766	136	2818	149
6	1909	572	1445	420	1270	442	1404	512
7	5569	466	3107	242	3965	320	4053	333
8	0	0	0	0	0	0	0	0
9	5219	123	2830	53	3230	79	3859	105
10	5288	0	2861	70	3916	77	3857	105
11	1568	473	967	234	1344	422	1241	447
12	0	0	0	0	0	0	0	0
	21621	21	21294	2249	2679	2709	23339	2304

Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf
celda	voice	data	voice	data	voice	data	voice
1	13.83	9.83	16.67	0.271	19.75	0	15.43
2	11.96	8.022	12.91	0.133	12.66	0	12.86
3	13.17	8.894	15.971	0.041	16.832	0	16.11
4	16.11	7.011	3.398	0.049	15.71	0	16.67
5	12.16	8.589	10.73	0.064	12.27	0	14.89
6	10.82	7.165	10.77	0.088	15.58	0.021	8.319
7	12.44	8.255	12.09	0.101	12.65	0	11.97
8	11.05	7.514	14.75	0.164	11.83	0	14.74
9	10.45	7.445	13.29	0.133	10.29	0	11.56
10	8.457	5.494	13.19	0.094	10.33	0	11.26
11	12.38	9.21	11.06	0.089	12.29	0	9.797
12	9.022	5.774	13.36	0.195	11.61	0	15.01
138.3		9.23	148.2	1.4	149.7	0.0	146.6

Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf
celda	voice	data	voice	data	voice	data	voice
1	10.97	17.79	13.91	0.142	12.97	0	11.3
2	7.747	5.507	8.83	0.032	7.027	0	7.952
3	5.649	3.7	6.652	0.028	5.59	0	6.316
4	6.157	4.15	7.029	0.024	7.634	0	6.923
5	5.959	3.952	6.736	0.029	6.637	0	6.288
6	6.391	3.935	6.331	0.017	7.19	0	6.77
7	6.651	4.331	7.098	0.058	6.578	0	6.942
8	11.95	7.623	11.27	0.086	14.14	0.007	11.57
9	5.663	4.306	6.721	0.053	6.468	0	6.461
10	7.408	5.383	5.69	0.029	5.916	0	6.276
11	6.552	4.388	7.443	0.033	6.998	0	7.483
12	14.65	10.36	8.688	0.028	10.6	0	11.94
95.9		65.2	35.4	6.6	98.8	0.0	95.8

celda	Tasa de respuesta					nº de respuestas		
	I = 0	I = 5	I = 10	I = inf				
1	1143	7751	1316	0.053	1837	0.007	1323	0
2	6373	4732	6399	0.025	6988	0	6344	0
3	6571	4739	5348	0.083	6034	0	5388	0
4	5751	4111	5020	0.047	5636	0	4446	0
5	6394	4128	688	0.023	5508	0	6396	0
6	5502	3382	5684	0.005	4802	0	5262	0
7	7406	4842	5515	0.051	6673	0	6371	0
8	9345	6191	11,05	0.065	134	0	1303	0
9	6896	4319	3193	0	5381	0	6092	0
10	6436	4633	5631	0	5743	0	5040	0
11	4484	3768	5323	0	5153	0	4008	0
12	1054	8302	16,86	0.311	22,22	0	1194	0
	97,8%	98,9%	92,3%	0,1%	96,1%	0,0%	96,7%	0,0%

celda	Throughput (Mbps)			
	I = 0	I = 5	I = 10	I = inf
1	3.70150	4.04200	3.77200	4.00020
2	3.57290	3.75000	3.68950	3.82700
3	3.67820	3.41470	3.28120	3.44490
4	3.41670	3.27510	3.36830	3.47780
5	3.64790	3.45090	3.71980	3.94680
6	3.47500	3.54060	3.91820	3.23410
7	3.58830	3.62660	3.69760	3.64670
8	3.52040	3.88030	3.59720	3.90510
9	3.45600	3.75680	3.52630	3.62270
10	3.16500	3.75480	3.52300	3.59120
11	3.67780	3.54600	3.83550	3.49320
12	3.28660	3.76790	3.52390	3.87540
42.18170		43.95570	43.98540	43.92080

Throughput (Mbps)					
celda	I=0	I=5	I=10	I=inf	
1	3.45980	3.81630	3.74290	3.53990	
2	3.64310	3.77880	3.56840	3.68190	
3	3.73800	4.08460	3.91040	4.04510	
4	3.47610	3.62550	3.68310	3.60630	
5	3.81570	4.09870	4.18550	3.97270	
6	3.38640	3.91240	3.56590	3.62960	
7	3.88380	4.23170	4.00570	4.04880	
8	3.52510	3.51060	3.87390	3.60850	
9	3.83220	4.28430	4.13260	4.18650	
10	4.07600	3.92540	4.05950	4.09430	
11	3.48830	3.85900	3.63000	3.70280	
12	3.79500	3.06750	3.45470	3.61050	
44.11308		45.51480	45.86370	45.72660	

celda	Throughput (Mbps)				l = inf
	I = 0	I = 5	I = 10	I = inf	
1	3.56730	3.71920	3.83450	3.78230	
2	3.66020	3.64870	3.72320	3.85020	
3	3.95030	4.06780	3.94150	3.93000	
4	3.67830	3.79330	3.76740	3.40340	
5	3.85070	4.16200	3.84480	3.99880	
6	3.56790	3.79550	3.58720	3.65010	
7	4.09060	3.93560	4.17250	4.20520	
8	3.26250	3.61440	3.76290	3.73095	
9	3.99770	3.79000	3.90420	4.17720	
10	4.00200	3.89700	3.69500	3.65000	
11	3.47100	3.20970	3.65900	3.87110	
12	1.14700	4.07630	3.62170	3.65260	
44,529.80					
45,752.30					
45,817.00					
45,610.00					

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Escenario $\lambda_{voice} = 30$ y $\lambda_{data} = 30$.

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	16.65	12.54	0.772	12.11	0.009	17.57	0	
2	15.64	16.33	14.01	0.671	17.66	0.173	14.85	0
3	14.85	13.5	18.51	1.71	18.95	0.137	16.62	0
4	13.57	11.43	17.51	1.774	18.85	0.31	16.73	0
5	17.88	17.43	11.75	0.359	14.36	0.123	15.44	0
6	15	15.02	21.36	2.432	17.39	0.071	16.95	0
7	15.17	15.82	16.3	1.176	14.11	0.101	13.88	0
8	12.78	12.07	14.08	0.754	16.67	0.146	15.51	0
9	11.2	11.59	19.47	1.891	17.12	0.098	18.59	0
10	17.23	17.77	13.82	0.878	14.98	0.008	12.6	0
11	13.82	13.12	20.19	2.309	13.9	0.018	17.3	0
12	14.2	15.04	12.45	0.27	19.01	0.385	20.15	0
	176.1	175.9	193.0	1.50	195.9	1.6	196.2	0.0

celda	Throughput (Mbps)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	4.09790	4.03330	3.90480	4.21200	4.56450	4.21200	4.47880	
2	4.04560	4.12200	4.56450	4.21200				
3	3.9120	4.54860	4.60580	4.36550				
4	3.85120	4.46030	4.61460	4.37870				
5	4.16100	3.90290	4.24440	4.27220				
6	3.94200	4.76400	4.46260	4.42330				
7	3.97070	4.39420	4.24600	4.09350				
8	3.75530	4.09860	4.40890	4.19240				
9	3.64530	4.56270	4.36730	4.56780				
10	4.15080	4.15600	4.28660	3.98490				
11	3.78470	4.72870	4.06160	4.43690				
12	3.92230	3.93430	4.70580	4.75240				
	47.03900	51.69560	52.47190	52.15840				

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	2028	0	1120	0	1213	0	1040	0
3	4890	0	2518	0	2539	0	2368	0
4	2006	0	950	0	1328	0	1212	0
5	3715	0	3095	0	2380	0	2377	0
6	2161	0	1007	0	825	0	987	0
7	5762	0	4090	0	2829	0	3155	0
8	0	0	0	0	0	0	0	0
9	6332	0	3088	0	2306	0	2709	0
10	5306	0	2567	0	2467	0	3279	0
11	1936	0	864	0	851	0	879	0
12	0	0	0	0	0	0	0	0
	34136	0	19259	0	16738	0	18006	0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	10.75	11.05	15.73	1.11	18.72	0.093	16.91	0
2	10.7	10.73	16.76	1.203	14.99	0.101	13.99	0
3	7.969	8.83	8.419	0.256	8.42	0	8.616	0
4	10.86	10.93	10.33	0.314	13.66	0.04	12.18	0
5	9.133	9.134	9.31	0.596	8.725	0.019	8.964	0
6	11.41	11.76	11.64	0.341	9.371	0	11.96	0
7	10.25	10.27	10.24	0.567	9.664	0.097	9.683	0
8	14.65	15.42	12.64	0.689	16.22	0.133	14.96	0
9	9.312	9.847	7.841	0.523	7.322	0.008	7.288	0
10	8.77	9.031	8.448	0.239	6.704	0	7.639	0
11	10.06	11.52	10.97	0.414	9.788	0.003	10.04	0
12	12.4	12.29	16.12	0.901	19.39	0.3	15.55	0
	127.1	130.9	138.0	8.2	142.7	0.8	137.8	0.0

celda	Throughput (Mbps)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	3.89210	4.46090	4.04350	4.38870				
2	4.00570	4.21080	4.34540	4.20500				
3	4.35920	4.53620	4.74720	4.59500				
4	4.34760	4.19580	4.30390	4.41910				
5	4.42290	5.03610	4.82130	4.82440				
6	4.02640	4.10260	4.09620	4.26730				
7	4.45210	4.92170	4.71050	4.81910				
8	3.94520	4.23410	4.27430	4.08410				
9	4.29840	4.72700	4.97020	4.58700				
10	4.37290	4.71970	4.68610	5.21260				
11	4.10540	4.78140	4.81860	4.34930				
12	3.95530	4.36890	4.43210	4.36410				
	50.07880	54.54560	54.08930	54.45670				

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	3172	0	947	0	696	0	502	0
3	7368	0	3230	0	2427	0	2154	0
4	3618	0	1016	0	805	0	750	0
5	7741	0	2427	0	1976	0	2093	0
6	2507	0	963	0	816	0	746	0
7	7013	0	2470	0	2465	0	2149	0
8	0	0	0	0	0	0	0	0
9	7741	0	3412	0	3262	0	2089	0
10	6860	0	3429	0	2100	0	2933	0
11	2432	0	1084	0	803	0		

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Escenario $\lambda_{voice} = 40$ y $\lambda_{data} = 10$.

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	1447	0	1429	0	1581	0	1608	0
3	3351	0	3064	0	2394	0	3345	0
4	1481	0	1598	0	1399	0	1353	0
5	3775	0	2795	0	4026	0	3494	0
6	1137	0	1440	0	1639	0	1451	0
7	3816	0	2991	0	3087	0	3833	0
8	0	0	0	0	0	0	0	0
9	2802	0	2470	0	4216	0	2662	0
10	2640	0	3568	0	3981	0	3893	0
11	1208	0	1259	0	1451	0	1062	0
12	0	0	0	0	0	0	0	0
	21657	0	20604	0	23374	0	22701	0

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	1653	508	1348	464	1512	540	1478	505
3	3066	122	2091	140	3259	126	4031	173
4	1602	539	1487	478	1226	482	1634	703
5	2771	85	3452	168	3983	196	2970	150
6	1245	379	1392	455	1181	347	1265	496
7	3810	309	4168	351	3747	356	3664	377
8	0	0	0	0	0	0	0	0
9	2905	65	3560	120	4264	131	3051	95
10	2827	62	3179	101	2672	109	3342	100
11	1091	281	1352	468	1101	405	1177	319
12	0	0	0	0	0	0	0	0
	20570	2350	23029	2745	22954	2692	22773	2918

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	9.911	1679	13.02	0	12.72	0	11.58	0
2	12.46	2.335	15.34	0	13.22	0	9.904	0
3	12.12	2.458	12.27	0	14.61	0	12.21	0
4	11.7	2.22	12.24	0	10.86	0	8.726	0
5	11.63	2.144	14.39	0	10.96	0	11.27	0
6	14.27	3.477	11.59	0.045	9.641	0	10.29	0
7	6.735	1.219	11.04	0	8.896	0	12.4	0
8	9.545	2.159	9.195	0	11.9	0	14.62	0
9	13.09	2.664	6.359	0	8.804	0	15.02	0
10	13.7	3.151	11.77	0	10.08	0	11.67	0
11	11.63	1.852	14.76	0	12.55	0	11.66	0
12	8.338	1.582	11.14	0	9.518	0	10.33	0
	135.2	26.9	143.1	0.0	134.8	0.0	139.7	0.0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	14.68	3.232	13.56	0	11	0	12.13	0
2	7.269	1.319	6.651	0	7.464	0	7.057	0
3	4.397	1.495	4.432	0	3.555	0	5.225	0
4	6.988	1.582	6.951	0	6.119	0	6.384	0
5	5.547	2.435	4.511	0	5.348	0	5.395	0
6	4.893	0.761	6.171	0	7.876	0	6.463	0
7	6.315	1.9	5.382	0	4.539	0	5.84	0
8	12.37	2.465	13.69	0	8.274	0	13.68	0
9	3.886	0.795	3.45	0	5.717	0	3.768	0
10	4.134	1.203	4.729	0	5.234	0	4.973	0
11	5.552	1.184	5.217	0	7.401	0	5.125	0
12	10.31	1.409	10.71	0	9.799	0	12.23	0
	86.9	19.8	85.5	0.0	83.4	0.0	83.3	0.0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	13.64	3.05	7.631	0	11.46	0	10.34	0
2	5.803	1.394	4.36	0	5.703	0	5.782	0
3	4.682	1.137	4.19	0	4.241	0	4.882	0
4	4.955	1.324	4.607	0	3.685	0	5.508	0
5	4.231	1.387	5.177	0	5.208	0	4.723	0
6	3.935	1.343	4.392	0	3.773	0	4.202	0
7	4.794	1.15	4.948	0	4.704	0	5.109	0
8	11.68	2.423	12.78	0	9.374	0	10.8	0
9	4.313	1.302	4.406	0.022	4.913	0	4.312	0
10	3.884	1.189	3.935	0	3.986	0	4.747	0
11	3.855	0.838	4.612	0	3.445	0	3.632	0
12	14.76	2.745	10.94	0	13.76	0	17.24	0
	80.3	19.3	72.6	0.0	74.4	0.0	75.8	0.0

celda	Throughput (Mbps)							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	3.67940	3.15520	2.54940	3.45730				
2	3.91480	3.72770	3.7780	3.84990				
3	3.99320	3.21050	4.00580	4.1790				
4	3.88680	3.80080	3.56810	3.32270				
5	3.33650	4.13590	4.21610	4.98210				
6	3.53280	3.74110	3.55980	3.74940				
7	4.07260	4.17360	4.09830	4.47470				
8	3.56550	3.63770	3.36330	3.49720				
9	3.90570	4.0350	4.17040	3.93980				
10	3.87590	3.93630	3.82740	4.05510				
11	3.50130	3.77870	3.50150	3.49900				
12	3.76740	3.52240	3.7470	3.60020	</td			

**MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE**

Escenario $\lambda_{voice} = 40$ y $\lambda_{data} = 30$.

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	3063	0	1821	0	1577	0	1526	0
3	5919	0	4368	0	3922	0	4827	0
4	2655	0	2200	0	1537	0	1611	0
5	5334	0	4906	0	4793	0	4306	0
6	3030	0	1714	0	1885	0	1772	0
7	7737	0	4287	0	4509	0	4605	0
8	0	0	0	0	0	0	0	0
9	7593	0	5028	0	4971	0	4690	0
10	7351	0	5729	0	4856	0	4151	0
11	2490	0	1454	0	1301	0	1490	0
12	0	0	0	0	0	0	0	0
	45772	0	31507	0	29081	0	28978	0

celda	Handovers							
	I = 0		I = 5		I = 10		I = inf	
celda	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	0	0	0	0	0	0	0	0
2	2831	1474	1452	1398	1436	1320	1664	829
3	6138	1469	3900	171	4903	1255	5556	1438
4	3768	2693	1849	116	1644	1315	1288	1605
5	7159	3366	6438	2601	4562	2131	4656	2272
6	3210	2173	1352	185	1387	1261	1621	1505
7	8020	1821	5850	1720	4822	1889	4451	1882
8	0	0	0	0	0	0	0	0
9	7664	892	4864	1052	5404	1008	5681	1091
10	8168	1203	5183	1039	4988	374	4895	1315
11	2236	1357	1557	1144	1325	1571	1227	1422
12	0	0	0	0	0	0	0	0
	51194	14438	32445	12475	30011	12834	31033	13429

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	22.42	18.21	19.82	0.823	23.73	0.267	16.95	0
2	21.61	16.42	27.95	3.56	19.11	0.092	24.16	0
3	22.67	18.71	20.53	1.03	23.62	0.319	23.44	0
4	18.18	14.63	23.66	1.664	22.04	0.198	23.21	0
5	22.43	17.93	20.23	0.673	19.3	0.062	16.53	0
6	21.3	16.11	23.16	1.491	25.43	0.252	22.74	0
7	20.87	17.12	22.19	1.688	25.95	0.35	22.51	0
8	20.52	16.21	23.88	1.694	23.84	0.329	21.56	0
9	19.53	15.14	22.3	1.064	22.05	0.025	22.63	0
10	16.61	12.54	23.27	1.327	21.27	0.148	21.39	0
11	24.19	19.37	25.51	1.852	20.36	0.017	20.51	0
12	16.12	13.01	23.3	2.014	24.63	0.192	24.54	0
	24.64	19.65	27.65	1.89	27.30	2.3	26.22	0.0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	25.3	21	23.38	1.805	28.53	0.676	22.22	0
2	20.43	15.11	22.56	0.932	20.37	0.193	20	0
3	17.73	9.82	12.32	0.473	11.31	0.025	13.17	0
4	16.27	12.95	20.81	1.031	16.61	0.045	20.61	0
5	17.39	11.78	13.83	0.518	14.68	0.046	15	0
6	18.98	14.89	19.71	0.98	19.04	0.04	18.6	0
7	15.63	12.09	16.45	0.59	14.36	0.023	15.81	0
8	22.66	17.64	23.33	1.346	23.48	0.143	30.52	0
9	11.95	12.64	12.64	0.555	12.96	0.055	11.62	0
10	12.16	10.36	13.36	0.361	13.07	0	11.54	0
11	13.81	11.35	11.95	0.571	11.76	0.027	16.42	0
12	19.85	15.33	22.53	1.875	23.09	0.599	25.96	0
	201.8	161.5	287.6	1.8	222.5	0.0		

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	24.59	24.93	28.38	0.947	24.45	1.824	21.36	0
2	24.02	23.94	27.21	7.604	26.65	2.685	22.13	0
3	25.48	24.83	20.54	2.281	23.52	0.755	24.43	0
4	20.69	20.44	23.67	3.997	24.54	1.24	27.52	0
5	24.22	24.47	26.26	6.932	29.8	5.289	27.13	0
6	22.94	23.8	24.46	5.803	29.14	3.433	23.16	0
7	22.61	22.8	26.36	2.464	24.07	2.01	26.84	0
8	22.59	22.93	23.6	5.459	24.35	1.428	22.83	0
9	21.16	21.11	26.36	5.594	22.59	1.553	26.68	0
10	18.1	18.45	24.68	5.748	24.76	1.785	26.14	0
11	26.33	25.77	26.65	6.687	26.02	1.779	27.84	0
12	18.26	18.44	28.07	7.126	22.22	1.145	24.95	0
	271.8	271.7	300.6	6.83	302.1	24.9	301.0	0.0

celda	Tasa de Pérdidas (%)							
	I = 0		I = 5		I = 10		I = inf	
celda	voice	data	voice	data	voice	data	voice	data
1	23.32	23.29	28.08	6.571	22.88	0.653	23.43	0
2	24.43	23.21	27.4	7.604	26.65	2.685	22.13	0
3	13.65	14.17	15.03	2.861	17.32	0.654	18.41	0
4	20.79	21.72	22.05	1.061	21.31	0.339	22.05	0
5	17.23	17.18	16.86	5.229	19.35	0.516	22.34	0
6	19.54	19.4	21.2	2.753	22.24	1.136	22.9	0
7	18.76	18.1	19.8	4.88	23.77	1.261</		

Bibliografía

- [1] H. Hu, J. Zhang, X. Zheng, Y. Yang y P. Wu, Self-configuration and self-optimization for LTE networks, vol. 48, *Self-configuration and self-optimization for LTE networks*, 2010.
- [2] C. Johnson, *Long Term Evolution in Bullets*, 2012.
- [3] Z. Altman, S. Sallem, R. Nasri, B. Sayrac y M. Clerc, «Particle swarm optimization for Mobility Load Balancing SON in LTE networks,» *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2014.
- [4] P. Szilágyi, Z. Vincze y C. Vulkán, «Integrated Mobility Load Balancing and Traffic Steering mechanism in LTE,» *IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013.
- [5] O. Iacoboaeia, B. Sayrac, S. B. Jemaa y P. Bianchi, «SON Coordination for parameter conflict resolution: A reinforcement learning framework,» *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2014.
- [6] L. Xu, Y. Chen, K. K. Chai, Y. Luan y D. Liu, «Cooperative mobility load balancing in relay cellular networks,» *IEEE/CIC International Conference on Communications in China (ICCC)*, 2013.
- [7] N. Zia y A. Mitschele-Thiel, «Self-organized neighborhood mobility load balancing for LTE networks,» *IFIP Wireless Days (WD)*, 2013.
- [8] J. Bartelt, A. Fehske, H. Klessig, G. Fettweis y J. Voigt, «Joint Bandwidth Allocation and Small Cell Switching in Heterogeneous Networks,» *IEEE 78th Vehicular Technology Conference (VTC Fall)*, 2013.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- [9] T. Yamamoto y S. Konishi, «Impact of small cell deployments on mobility performance in LTE-Advanced systems,» *IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*, 2013.
- [10] S. Oh, «User mobility impacts to mobility load balancing for self-organizing network over LTE system,» *14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, 2018.
- [11] S. Oh, H. Kim, J. Na y Y. Kim, «User performance impacts by mobility load balancing enhancement for self-organizing network over LTE system,» *XVII Workshop on Information Processing and Control (RPIC)*, 2017.
- [12] N. M. Sumita Mishra, «Load Balancing Optimization in LTE/LTE-A Cellular Networks: A Review,» *Networking and Internet Architecture*, 2014.
- [13] M. M. Hasan, S. Kwon y J.-H. Na, «Adaptive Mobility Load Balancing Algorithm for LTE Small-Cell Networks,» *IEEE Transactions on Wireless Communications*, vol. 17, 2018.
- [14] J. Park, Y. Kim y J.-R. Lee, «Mobility Load Balancing Method for Self-Organizing Wireless Networks Inspired by Synchronization and Matching With Preferences,» *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 2594 - 2606, 2018.
- [15] B.-H. Lee, E. K. Dewi y M. F. Wajdi, «Using Load Balancing Mechanism to Reduce Overload in LTE/EPC Defined Network,» *7th International Conference on Smart Computing & Communications (ICSCC)*, 2019.
- [16] L. Xu, X. Cheng, Y. Liu y W. Chen, «Mobility load balancing aware radio resource allocation scheme for LTE-Advanced cellular networks,» *Conference: 2015 IEEE 16th International Conference on Communication Technology (ICCT)*, 2015.
- [17] Y. L. Lee, T. C. Chuah, J. K.-K. Loo y A. Vinel, «Recent Advances in Radio Resource Management for Heterogeneous LTE/LTE-A Networks,» *IEEE Communications Surveys & Tutorials*, 2014.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- [18] I. S. G. K. Sofoklis Kyriazakos, 4G Mobile & Wireless Communications Technologies, River Publishers Series in Communications.
- [19] J. M. Ruiz-Avilés, M. Toril, S. Luna-Ramírez, V. Buenestado y M. A. Regueira, «Analysis of Limitations of Mobility Load Balancing in a Live LTE System,» *IEEE Wireless Communications Letters*, vol. 4, pp. 417 - 420, 2015.
- [20] P. Szilágyi, Z. Vincze y C. Vulkán, «Enhanced Mobility Load Balancing Optimisation in LTE,» *IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, 2012.
- [21] Z. Gao, C. Chen, Y. Li, B. Wen, L. Huang y Y. Zhao, «A mobility load balancing algorithm based on handover optimization in LTE network,» *10th International Conference on Computer Science & Education (ICCSE)*, 2015.
- [22] M. Sauter, From GSM to LTE-Advanced Pro and 5G: An Introduction to Mobile Networks and Mobile Broadband, wiley, 2017.
- [23] S. P. J. S. Erik Dahlman, 4G LTE-Advanced Pro and The Road to 5G, 2016.
- [24] R. R. Amitabha Ghosh, Essentials of LTE and LTE-A (The Cambridge Wireless Essentials Series), Cambridge University Press, 2011.
- [25] S. P. J. S. Erik Dahlman, 4G: LTE LTE-Advanced for Mobile Broadband, Academic Press, 2011.
- [26] 3gpp, «<https://www.3gpp.org/>,» 2020. [En línea]. Available: <https://www.3gpp.org/specifications/releases/71-release-9>.
- [27] A. Krendzel, «LTE-A Mobile Relay Handling: Architecture Aspects,» *European Wireless 2013; 19th European Wireless Conference*, 2013.
- [28] R. Comes, F. Alvarez, F. Palacio, R. Ferrus, J. Perez y O. Sallent, LTE: Nuevas Tendencias en Comunicaciones Móviles, 2010.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- [29] X. W. R. L. Junyi Li, OFDMA Mobile Broadband Communications: A Systems Approach, Cambridge University Press, 2013.
- [30] C. Cox, An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications, 2012.
- [31] S. C. Y. L. S. P. S. J. SeungJune Yi, Radio Protocols for LTE and LTE-Advanced, wiley, 2012.
- [32] C. L. Jean-Gabriel Rémy, LTE Standards, wiley, 2014.
- [33] 3gpp, «<https://www.3gpp.org/>,» 2020. [En línea]. Available: <https://www.3gpp.org/specifications/releases/70-release-10>.
- [34] «<https://usatcorp.com/>,» 2019. [En línea]. Available: <https://usatcorp.com/faqs/understanding-lte-signal-strength-values/>.
- [35] D. Jungnickel, Graphs, Networks and Algorithms, Springer-Verlag Berlin Heidelberg, 2013.
- [36] D. C. Scott Miller, Probability and Random Processes, Second Edition: With Applications to Signal Processing and Communications, Academic Press, 2012.
- [37] «<http://lya.fciencias.unam.mx/>,» 2020. [En línea]. Available: <http://lya.fciencias.unam.mx/lars/0626/>.
- [38] J. F. C. Kingman, Poisson Processes (Oxford Studies in Probability), Oxford University Press, USA, 1993.
- [39] W. C. Chan, Performance Analysis of Telecommunications and Local Area Network, Kluwer Academic Publisher, 2002.
- [40] M. K. Yuri Suhov, Probability and statistics by example. Markov chains: a primer in random processes and their applications, Cambridge University Press, 2008.
- [41] M. A. Alzate, Apuntes de Clase.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

- [42] A. M. E. S. Vijay K. Rohatgi, *An Introduction to Probability and Statistics*, wiley, 2015.
- [43] S. M. Ross, *Simulation*, Fifth Edition, Academic Press, 2012.
- [44] B. G. Steven I. Gordon, *Introduction to Modeling and Simulation with MATLAB® and Python*, Chapman and Hall/CRC;Taylor & Francis, 2017.
- [45] J. W. M. A. T. C. Robert V. Hogg, *Introduction to Mathematical Statistics*, Pearson, 2018.
- [46] S. M. Ross, *Introduction to probability and statistics for engineers and scientists*, Academic Press, 2013.
- [47] G. C. Canavos, *Probabilidad y Estadística. Aplicaciones y Métodos*, Mc GRAW HILL, 1988.
- [48] «<https://www.statisticshowto.com/>,» 2020. [En línea]. Available: <https://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/sample-variance/>. [Último acceso: 2020].
- [49] «<http://www.farnell.com/>,» 2019. [En línea]. Available: <http://www.farnell.com/datasheets/1924741.pdf>.
- [50] «<https://www.electronics-notes.com/>,» 2019. [En línea]. Available: <https://www.electronics-notes.com/articles/antennas-propagation/propagation-overview/free-space-path-loss.php>.
- [51] C. J. T. E. James Pyles, *Guide to TCP/IP: IPv6 and IPv4*, Cengage Learning, 2016.
- [52] A. Soysal, B. Mizrahi, I. Ali, K. B. Loğoglu y A. Akman, «A self-organizing downlink ICIC scheme for LTE,» *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2016.

*MEJORAMIENTO DEL MODELO BASADO EN MLB (MOBILITY LOAD BALANCING) SOBRE
REDES LTE*

[53] H. Zarrinkoub, Understanding LTE with MATLAB: From Mathematical Modeling to Simulation and Prototyping, Wiley Desktop Editions, 2014.

[54] «<http://mathworld.wolfram.com>,» 2019. [En línea]. Available: <http://mathworld.wolfram.com/MaclaurinSeries.html>.