



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍAS EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE**  
**COMUNICACIÓN**

**“EVALUACIÓN DEL IMPACTO DE LA INTERFERENCIA INTER-  
PROTOCOLO ENTRE LAS TECNOLOGÍAS 802.11AH Y LORA EN UN  
ENTORNO DE SIMULACIÓN”**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**AUTOR: MATEO ALEXANDER TITO LARA**  
**DIRECTOR: MSC. HERNÁN MAURICIO DOMÍNGUEZ LIMAICO**

**IBARRA-ECUADOR**

**2022**



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA**  
**UNIVERSIDAD**  
**TÉCNICA DEL NORTE**

**IDENTIFICACIÓN DE LA OBRA**

En cumplimiento del Art. 144 de la ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información.

<b>DATOS DEL CONTACTO</b>	
<b>CÉDULA DE IDENTIDAD</b>	1004497473
<b>APELLIDOS Y NOMBRES</b>	Tito Lara Mateo Alexander
<b>DIRECCIÓN</b>	Av. José Miguel Vaca Flores; Víctor Manuel Guzmán
<b>E-MAIL</b>	<a href="mailto:matitol@utn.edu.ec">matitol@utn.edu.ec</a>
<b>TELÉFONO MÓVIL</b>	0984515533
<b>DATOS DE LA OBRA</b>	
<b>TEMA</b>	“Evaluación del impacto de la interferencia inter-Protocolo entre las tecnologías 802.11ah y Lora en un entorno de simulación”
<b>AUTOR</b>	Tito Lara Mateo Alexander
<b>FECHA</b>	03/01/2022
<b>PROGRAMA</b>	PREGRADO <u> X </u> POSTGRADO <u> ___ </u>
<b>TÍTULO</b>	Ingeniero en Electrónica y Redes de Comunicación
<b>DIRECTOR</b>	MSc. Hernán Mauricio Domínguez Limaico



**UNIVERSIDAD TÉCNICA DEL NORTE**

**BIBLIOTECA UNIVERSITARIA**

### **CONSTANCIA**

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 03 días del mes de enero del 2022

**EL AUTOR:**

.....  
Mateo Alexander Tito Lara

C.I.: 100449747-3



**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CERTIFICACIÓN**

MAGISTER MAURICIO DOMÍNGUEZ, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que, el presente Trabajo de Titulación "EVALUACIÓN DEL IMPACTO DE LA INTERFERENCIA INTER-PROTOCOLO ENTRE LAS TECNOLOGÍAS 802.1 IAH Y LORA EN UN ENTORNO DE SMULACIÓN". Ha sido desarrollado por el señor Tito Lara Mateo Alexander, bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.

MSc. Mauricio Domínguez

DIRECTOR

## **DEDICATORIA**

*A mi familia y amigos.*

## **AGRADECIMIENTOS**

*Quiero agradecer a mi madre por su paciencia y apoyo incondicional durante todos mis años de formación académica, es gracias a sus enseñanzas y ejemplo que me he mantenido firme hasta culminar otra etapa de mi vida.*

*A las maravillosas personas que tuve la suerte de conocer durante mi paso por la universidad. A Adri, Juan, Bryan, Cristian y Joha, amigos con quienes compartí de cerca cada uno de los buenos y malos momentos de la carrera.*

*Por último, agradecer a mi tutor del proyecto, quien dedicó su tiempo y guía durante el desarrollo de este trabajo de titulación.*

*Gracias a todos.*

# CONTENIDO

1.	Capítulo I: Planteamiento del Proyecto .....	1
1.1	Antecedentes.....	1
1.2	Definición del Problema.....	2
1.3	Justificación del Problema.....	3
1.4	Objetivos.....	4
1.4.1	Objetivo General.....	4
1.4.2	Objetivos Específicos .....	5
1.5	Alcance .....	5
2.	Capítulo II: Fundamento Teórico .....	7
2.1	Estándar 802.11ah .....	7
2.1.1	Arquitectura 802.11ah .....	10
2.1.1.1	Capa Física PHY 802.11ah.....	11
2.1.1.1.1	Canalización.....	11
2.1.1.1.2	Modos de Operación .....	13
2.1.1.1.3	Modelos de Propagación.....	17
2.1.1.1.4	Formato del PHY PPDU.....	18
2.1.1.2	Capa MAC 802.11ah .....	21
2.1.1.2.1	Soporte a gran cantidad de estaciones .....	22
2.1.1.2.2	Ahorro de Energía.....	23
2.1.1.2.3	TIM y Segmentación de Página.....	25
2.1.1.2.4	Mecanismos de Acceso al Canal .....	26
2.1.1.2.5	Ventana de Acceso Restringido.....	29
2.1.1.2.6	Cabecera Reducida .....	32
2.2	Estándar LoRA .....	34
2.2.1	Clases.....	35
2.2.2	Arquitectura LoRA.....	37

2.2.2.1	Capa Física LoRa.....	39
2.2.2.1.1	Formatos De Mensajes De Trama Física.....	40
2.2.2.1.2	Ventana De Recepción.....	41
2.2.2.1.3	Chirp Spread Spectrum.....	42
2.2.2.1.4	Factor De Propagación.....	44
2.2.2.2	Capa MAC LoRa.....	47
2.2.2.2.1	Tipos De Mensajes Mac.....	48
2.2.2.2.2	Payload Mac.....	48
2.2.2.2.3	Seguridad.....	51
2.2.2.2.4	Comandos Mac.....	53
3.	Capítulo III: Diseño e Implementación del Proyecto.....	56
3.1	Arquitectura de los módulos en NS-3.....	56
3.1.1	Módulo IEEE 802.11ah.....	57
3.1.1.1	Modelo para la capa PHY IEEE 802.11ah.....	57
3.1.1.2	Modelo para la capa MAC IEEE 802.11ah.....	59
3.1.2	Módulo LoRa.....	60
3.1.2.1	Modelo para la capa PHY LoRa.....	60
3.1.2.2	Modelo para la capa MAC LoRa.....	63
3.2	Modificación de las clases dentro del simulador.....	64
3.3	Desarrollo de un solo modelo de simulación.....	72
3.4	Modelo único para controlar las iteraciones.....	85
3.5	Recolección de datos.....	89
4.	Capítulo IV: Resultados.....	92
4.1	Comprobación Teórico-Matemática.....	92
4.2	Evaluación de las métricas.....	99
4.2.1	Throughput.....	108
4.2.2	Total de paquetes perdidos.....	118

4.2.3	SINR.....	118
4.3	Consideraciones adicionales.....	134
	Conclusiones.....	138
	Recomendaciones.....	141
5.	Bibliografía.....	143
6.	Anexos.....	148

## ÍNDICE DE FIGURAS

Figura 1. Arquitectura de una red 802.11ah .....	11
Figura 2. Canales según su ancho de banda para 802.11ah.....	12
Figura 3. Asignación de canales para varios países.....	13
Figura 4. Formatos de preámbulo para IEEE 802.11ah dentro del simulador .....	19
Figura 5. Formato S1G_SHORT.....	20
Figura 6. Formato S1G_LONG.....	20
Figura 7. Formato S1G_1M .....	21
Figura 8. Estructura jerárquica de una AID.....	23
Figura 9. STA con energía limitada.....	25
Figura 10. Distribución de RAW y PRAW entre beacons .....	27
Figura 11. Formato de un elemento TWT .....	28
Figura 12. Ventana de Acceso Restringido (RAW) .....	30
Figura 13. Formato de un elemento RPS.....	31
Figura 14. Formato del subcampo RAW Assignments .....	31
Figura 15. Formato del subcampo de control RAW .....	31
Figura 16. Formato del subcampo RAW Slot Definition.....	32
Figura 17. Comparación de cabeceras MAC 802.11ah y 802.11 legacy.....	33
Figura 18. Formato de un SID.....	34
Figura 19. Clases de LoRaWAN .....	35
Figura 20. Ventanas de recepción para una clase A .....	36
Figura 21. Ventana de recepción programada para una clase B.....	36
Figura 22. Ventana de recepción continua para una clase C .....	37
Figura 23. Arquitectura de una red LoRaWAN .....	38
Figura 24. Disposición de canales en Norteamérica (FCC) .....	40
Figura 25. Estructura del uplink PHY .....	41
Figura 26. Estructura del downlink PHY .....	41
Figura 27. Ventanas de recepción.....	42
Figura 28. Estructura de un payload PHY .....	47
Figura 29. Estructura del payload MAC.....	48
Figura 30. Tipos de mensajes MAC .....	48
Figura 31. Formato de cabecera de trama.....	49
Figura 32. Contenido del campo FCtrl para downlink .....	49

Figura 33. Contenido del campo FCtrl para uplink .....	49
Figura 34. Esquema de seguridad de LoRa .....	52
Figura 35. Uso de las llaves AppSKey y NwkSKey .....	53
Figura 36. Lista de comandos MAC.....	54
Figura 37. Fragmento de la definición de MCS .....	58
Figura 38. Cálculo del SNR, extracto de las funciones de interferencia.....	59
Figura 39. Extracto de la función para asignar una RAW.....	59
Figura 40. Parámetros de sensibilidad para end-device SX1272 .....	61
Figura 41. Parámetros de sensibilidad para Gateway SX1301.....	62
Figura 42. Matriz de Colisión para determinar la interferencia en la transmisión .....	63
Figura 43. Diagrama de clases módulo IEEE 802.11ah en NS-3.....	65
Figura 44. Diagrama de clases módulo LoRaWAN en NS-3.....	67
Figura 45. Diagrama de secuencia módulo IEEE 802.1ah en NS-3.....	68
Figura 46. Diagrama de secuencia módulo LoRaWAN en NS-3.....	69
Figura 47. Creación de las clases helper para las capas PHY y MAC .....	70
Figura 48. Cambio de las frecuencias de los Receive Paths.....	70
Figura 49. Cambio de la región por defecto en el constructor .....	71
Figura 50. Configuración de end-device y Gateway en la región US .....	71
Figura 51. Cambio de la frecuencia de la segunda ventana de recepción .....	71
Figura 52. Cambio en la asignación de sub-bandas.....	72
Figura 53. Cambio en los canales lógicos .....	72
Figura 54. Bloque de código para exportar datos de la simulación de LoRa .....	73
Figura 55. Datos exportados de la simulación LoRa.....	73
Figura 56. Caso 1: la señal deseada se transmite antes que la interferente, no hay solapamiento.....	74
Figura 57. Caso 2: la señal interferente se transmite antes que la deseada, no hay solapamiento.....	75
Figura 58. Caso 3: la señal interferente se transmite durante la señal deseada, sí hay solapamiento.....	75
Figura 59. Caso 4: la señal interferente se transmite antes que finalice la señal deseada, sí hay solapamiento .....	75
Figura 60. Caso 5: la señal deseada se transmite durante la señal interferente, sí hay solapamiento.....	76

Figura 61. Caso 6: la señal deseada se transmite antes que finalice la señal interferente, sí hay solapamiento .....	76
Figura 62. Relación entre BER y SNR .....	80
Figura 63. Relación en BER y SINR .....	81
Figura 64. Diagrama de herencia para la clase SpectrumWifiPhy .....	81
Figura 65. Simulación del PER para IEEE 802.11ah con SNR entre 5 y 45 dB.....	83
Figura 66. Simulación del PER para IEEE 802.11ah con SNR entre 15 y 20 dB.....	84
Figura 67. Ejecutable para controlar las iteraciones.....	88
Figura 68. Demostración del ejecutable .....	88
Figura 69. Fichero resultado de la iteración .....	88
Figura 70. Conjunto de tiempos de duración del PPDU para un preámbulo SHORT ...	93
Figura 71. Conjunto de tiempos de duración del PPDU para un preámbulo SIG_1M..	93
Figura 72. Suma de los tiempos de cabecera y preámbulo del PPDU .....	95
Figura 73. Suma de los tiempos de cabecera y preámbulo del PPDU .....	96
Figura 74. PSD de la señal de LoRa.....	97
Figura 75. PSD de la señal IEEE 802.11ah .....	97
Figura 76. PSD de la interferencia entre LoRa e IEEE 802.11ah .....	98
Figura 77. GUI en Matlab para evaluar métricas .....	99
Figura 78. Gráfica lineal del throughput para los primeros resultados obtenidos.....	100
Figura 79. Gráfica logarítmica del throughput para los primeros resultados obtenidos	101
Figura 80. Gráfica lineal del porcentaje de paquetes perdidos para los primeros resultados obtenidos.....	101
Figura 81. Gráfica logarítmica del porcentaje de paquetes perdidos para los primeros resultados obtenidos .....	102
Figura 82. Gráfica lineal del SINR para los primeros resultados obtenidos .....	102
Figura 83. Gráfica logarítmica del SINR para los primeros resultados obtenidos .....	103
Figura 84. Gráfica lineal del throughput para 50 nodos de IEEE 802.11ah.....	109
Figura 85. Gráfica logarítmica del throughput para 500 nodos de IEEE 802.11ah .....	110
Figura 86. Gráfica lineal del throughput para 100 nodos de IEEE 802.11ah.....	111
Figura 87. Gráfica logarítmica del throughput para 100 nodos de IEEE 802.11ah .....	111
Figura 88. Gráfica lineal del throughput para 150 nodos de IEEE 802.11ah.....	112
Figura 89. Gráfica logarítmica del throughput para 150 nodos de IEEE 802.11ah .....	112
Figura 90. Gráfica lineal del throughput para 200 nodos de IEEE 802.11ah.....	114
Figura 91. Gráfica logarítmica del throughput para 200 nodos de IEEE 802.11ah .....	114

Figura 92. Gráfica lineal del throughput para todas las variaciones de IEEE 802.11a	115
Figura 93. Gráfica logarítmica del throughput para todas las variaciones de IEEE 802.11ah	116
Figura 94. Gráfica lineal del Throughput para 100 nodos de IEEE 802.11ah, variando el Radio de LoRa	116
Figura 95. Gráfica lineal del Throughput para 100 nodos de IEEE 802.11ah, variando el Payload	118
Figura 96. Gráfica lineal de los paquetes perdidos para 50 nodos de IEEE 802.11ah	119
Figura 97. Gráfica logarítmica de los paquetes perdidos para 50 nodos de IEEE 802.11ah	119
Figura 98. Gráfica lineal de los paquetes perdidos para 100 nodos de IEEE 802.11ah	120
Figura 99. Gráfica logarítmica de los paquetes perdidos para 100 nodos de IEEE 802.11ah	120
Figura 100. Gráfica lineal de los paquetes perdidos para 150 nodos de IEEE 802.11ah	121
Figura 101. Gráfica logarítmica de los paquetes perdidos para 150 nodos de IEEE 802.11ah	121
Figura 102. Gráfica lineal de los paquetes perdidos para 200 nodos de IEEE 802.11ah	122
Figura 103. Gráfica logarítmica de los paquetes perdidos para 200 nodos de IEEE 802.11ah	123
Figura 104. Gráfica lineal de los paquetes perdidos para todas las variaciones de IEEE 802.11ah	123
Figura 105. Gráfica logarítmica de los paquetes perdidos para todas las variaciones de IEEE 802.11ah	124
Figura 106. Gráfica lineal del Porcentaje de Paquetes Perdidos para 100 nodos de IEEE 802.11ah	125
Figura 107. Gráfica lineal del PPP para 100 nodos de IEEE 802.11ah, variando el Payload	126
Figura 108. Gráfica lineal del SINR para 50 nodos de IEEE 802.11ah	127
Figura 109. Gráfica logarítmica del SINR para 50 nodos de IEEE 802.11ah	127
Figura 110. Gráfica lineal del SINR para 100 nodos de IEEE 802.11ah	128
Figura 111. Gráfica logarítmica del SINR para 100 nodos de IEEE 802.11ah	128
Figura 112. Gráfica lineal del SINR para 150 nodos de IEEE 802.11ah	129

Figura 113. Gráfica logarítmica del SINR para 50 nodos de IEEE 802.11ah.....	129
Figura 114. Gráfica lineal del SINR para 200 nodos de IEEE 802.11ah .....	130
Figura 115. Gráfica logarítmica del SINR para 200 nodos de IEEE 802.11ah.....	130
Figura 116. Gráfica lineal del SINR para todas las variaciones de IEEE 802.11ah.....	131
Figura 117. Gráfica logarítmica del SINR para todas las variaciones de IEEE 802.11ah .....	131
Figura 118. Gráfica lineal del SINR para 100 nodos de IEEE 802.11ah .....	132
Figura 119. Gráfica lineal del SINR para 100 nodos de IEEE 802.11ah, variando el Payload .....	133

## ÍNDICE DE TABLAS

Tabla 1. Revisiones de IEEE 802.11. ....	9
Tabla 2. S1G MCSs para un canal de 1 MHz con $N_{ss} = 1$ .....	14
Tabla 3. S1G MCSs para un canal de 16 MHz con $N_{ss} = 4$ .....	14
Tabla 4. Campos que componen el PHY PPDU. ....	21
Tabla 5. Bandas de operación de LoRaWAN. ....	39
Tabla 6. Tabla de factores de propagación LoRa. ....	45
Tabla 7. Esquema de modulación de LoRa. ....	46
Tabla 8. Parámetros esenciales de la simulación de IEEE 802.11ah. ....	86
Tabla 9. Parámetros esenciales de la simulación de LoRa. ....	87
Tabla 10. Variaciones de la simulación de IEEE 802.11ah, en base al número de nodos. .....	89
Tabla 11. Iteraciones de la simulación de LoRa, en base al número de nodos. ....	90
Tabla 12. Registro de combinaciones realizadas. ....	90
Tabla 13. Iteraciones de la simulación de LoRa, en base al radio de distribución de los nodos.....	91
Tabla 14. Caso de estudio 1, transmisión de Lora empieza antes de IEEE 802.11ah y termina antes de IEEE 802.11ah.....	94
Tabla 15. Caso de estudio 2, transmisión de Lora empieza antes de IEEE 802.11ah y termina antes de IEEE 802.11ah.....	95
Tabla 16. Throughput y porcentaje de paquetes perdidos sin interferencia de LoRa. .	104
Tabla 17. Estructura del archivo RAWConfigFile. ....	104
Tabla 18. Configuraciones de parámetros RAW iniciales. ....	105
Tabla 19. Lista de nuevas configuraciones de RAW para las iteraciones de IEEE 802.11ah. ....	107
Tabla 20. Nuevos resultados obtenidos de las configuraciones de parámetros de RAW. .....	107

## RESUMEN

El presente trabajo describe el desarrollo de un modelo de simulación mediante el software de red open-source NS-3, con el fin de analizar el comportamiento del estándar IEEE 802.11ah cuando es interferido por el protocolo LoRa dentro de una red densa (masiva cantidad de estaciones). Para lo cual, se han seleccionado tres métricas principales a ser analizadas para determinar el impacto de la interferencia: Throughput (medida del rendimiento de la red), porcentaje total de paquetes perdidos y Signal-to-Interference + Noise Ratio (SINR).

Como punto de partida se realiza una investigación bibliográfica de ambas tecnologías en fuentes y estándares oficiales, para comprender el funcionamiento de sus capas PHY y MAC, así como también para definir los parámetros técnicos adecuados para su simulación.

A continuación, se modifican los módulos de IEEE 802.11ah y LoRa disponibles en GitHub (repositorio digital) para operar dentro de la banda de frecuencia correspondiente a Norteamérica (y Sudamérica) establecida por la FCC. Después de lo cual, se genera un modelo de simulación que tome en cuenta ambas tecnologías y se itera tantas veces sea necesario para cubrir las distintas variaciones de parámetros claves como: número de estaciones, radio de distribución máximo de los nodos, etc.

Para finalmente, con los datos recolectados de todas las iteraciones realizadas graficar las métricas antes mencionadas y analizarlas con el propósito de medir el impacto de esta interferencia inter-protocolo específica.

## **ABSTRACT**

This work describes the development of a simulation model using the open-source network software NS-3, to analyze the behavior of the IEEE 802.11ah standard when it is interfered by the LoRa protocol within a dense network (massive number of stations). For which reason, three main metrics have been selected to be analyzed to determine the impact of interference: Throughput (a measure of network performance), total percentage of lost packets and Signal-to-Interference + Noise Ratio (SINR).

As a starting point, a bibliographic investigation of both technologies is carried out in official sources and standards, to understand the operation of their PHY and MAC layers, as well as to define the appropriate technical parameters for the simulation.

The following step, the IEEE 802.11ah and LoRa modules available on GitHub (digital repository) are then modified to operate according the North American (and South American) frequency band set by the FCC. After which, a simulation model is generated that takes both technologies into account and is iterated as many times as necessary to cover the different variations of key parameters such as: number of stations, maximum node distribution radius, etc.

Finally, with the data collected from all the iterations accomplished, chart the metrics, and analyze them to measure the impact of this specific inter-protocol interference.

## **1. CAPÍTULO I: PLANTEAMIENTO DEL PROYECTO**

En este capítulo se detallarán las bases para el desarrollo del presente trabajo de titulación, siendo éstos: el tema, la problemática, los objetivos, el alcance y la justificación, con el fin de denotar la importancia de la realización de este trabajo.

### **1.1 ANTECEDENTES**

De acuerdo con reportes de Cisco, para 2030 habrán alrededor de 500 mil millones de dispositivos IoT (Internet of Things) conectados a internet (CISCO, 2020), lo que implica redes con múltiples tecnologías (LoRa, 802.11ah, etc.) trabajando en un mismo ecosistema. Las regulaciones en la banda de sub-GHz varían de acuerdo con la región, mientras en Europa está regulado a menos de 1% del ciclo de trabajo para evitar interferencias, en América no existe una regulación como tal en este aspecto (Castells-Rufas, Galin-Pons, & Carrabina, 2018), este hecho en conjunto con la masiva cantidad de dispositivos funcionando en redes densas, se convierte en un grave problema de desempeño cuando se trabaja con múltiples tecnologías LPWAN (Low Power Wireless Area Network).

Existe un precedente de un trabajo que comparaba las tecnologías IEEE 802.15.4 y 802.11ah (Ahmed, Rahman, & Hussain, A comparison of 802.11ah and 802.15.4 for IoT, 2016), por medio de métricas tales como: el tiempo de asociación, el Throughput y el retardo punto a punto, concluyendo que IEEE 802.11ah superaba a IEEE 802.15.4 en estos aspectos, además de en rango de cobertura, por otro lado, IEEE 802.15.4 demostró una mayor eficiencia en el consumo energético.

En la Escuela Politécnica Superior de Chimborazo se realizó un trabajo acerca de la evaluación del rendimiento del estándar IEEE 802.15.4 en entornos de interferencia (Tustón Torres, 2011), usando equipos reales para evaluar métricas de paquetes

intercambiados, rendimiento y espectros Wi-Fi, para proponer la implementación de un filtro selectivo microstrip (microcinta).

Un estudio similar se realizó al comprar IEEE 802.11ah y LoRa en NS-3, comparando los resultados de simulaciones de un solo protocolo a la vez, usando métricas de Throughput y pérdida de paquetes para determinar un porcentaje de congestión que favorecía a LoRa (Oukessou, Baslam, & Oukessou, 2018).

Cabe mencionar que, en el contexto local de la Universidad Técnica del Norte no existe constancia de trabajos similares que hagan referencia a las tecnologías IEEE 802.11ah o LoRa, una simulación en conjunto de estas, ni mucho de menos de un análisis; el presente proyecto supondrá el punto de partida para trabajos de coexistencia en la carrera de Telecomunicaciones.

## **1.2 DEFINICIÓN DEL PROBLEMA**

El trabajar con equipos reales es una fuerte limitante, en especial cuando se requiere estudiar el comportamiento de una red con gran cantidad de nodos, para suplir esta necesidad existen entornos de simulación especializados, como Omnet++ o Network Simulation 3 (NS-3). En este sentido ya ha sido implementado IEEE 802.11ah en NS-3 (Tian, y otros, 2018); En un trabajo similar se enriqueció el módulo de LoRa existente en NS-3 para estudiar el rendimiento de este protocolo bajo diferentes configuraciones de parámetros (Magrin, Capuzzo, & Zanella, 2020), hace unos cuantos años se presentó una comparación entre IEEE 802.11ah y 802.15.4 (Ahmed, Rahman, & Hussain, A comparison of 802.11ah and 802.15.4 for IoT, 2016), de manera similar existen estudios comparativos entre tecnologías que operan en la banda de sub-GHz, no obstante, ninguno de ellos aborda el problema de la interferencia entre los protocolos IEEE 802.11ah y LoRa, en gran parte debido a que el módulo actual de LoRa en el simulador de red no

cuenta con la capacidad para contabilizar la interferencia provocada por tecnologías diferentes a la propia.

En vista de esta problemática y las dificultades presentes, es necesario la modificación del módulo de LoRa en NS-3 para el desarrollo de una simulación en la banda de Sub-GHz (902 – 928 MHz) que permita evaluar el impacto de la interferencia entre IEEE 802.11ah y LoRa, a través del análisis de métricas claves como: la potencia de la señal, la tasa de bits, el throughput, el total de paquetes tanto enviados como recibidos y el SINR (Relación señal-interferencia más ruido). La contribución de la presente propuesta de investigación es el desarrollo de un modelo de simulación en conjunto para IEEE 802.11ah y LoRa, para evaluar el comportamiento de ambas tecnologías y que permita determinar el impacto de la interferencia inter-protocolo en el rendimiento de IEEE 802.11ah.

### **1.3 JUSTIFICACIÓN DEL PROBLEMA**

El futuro del Internet de las Cosas es concretar la visión de un mundo inteligente, donde cada vez más objetos se suman a comunicarse (Perwej, Omer, Sheta, Harb, & Adrees, 2019), ya no se trata solamente de computadoras o móviles, sino también de sensores, electrodomésticos, vehículos, maquinaria, etc. Desde su aparición en 2006 cada vez existen más trabajos de investigación relacionados a la aplicabilidad cada una de las múltiples tecnologías disponibles (Dachyar, Zagloel, & Saragih, 2019), sin embargo, un punto a considerar que no se ha tratado con la debida atención es la interferencia que se produce en redes donde se encuentran funcionando distintos protocolos, sumado al crecimiento exponencial de dispositivos IoT (Ryan & Watson, 2017) que llegarán a superar 4:1 a los humanos dentro de los próximos dos a tres años, conlleva a la necesidad de estudiar los impactos de esta problemática en redes densas.

El estándar IEEE 802.11ah combina las ventajas de Wi-Fi con una red de sensores de bajo consumo (LPWAN), para lograrlo trabaja en la banda sub-GHz, más específicamente entre 902 y 928 MHz (Meera & Rao, 2017); en la misma banda de frecuencia se encuentra operando otra de las tecnologías más prometedoras que ha venido revolucionando el terreno de la agricultura inteligente es LoRa (Marais, Malekian, & Abu-Mahfouz, 2017). Para sortear la barrera del hardware, existen entornos de simulación como Omnet++ (Varga & Hornig, 2008) o NS-3 (Panwar, 2017), siendo este último el predilecto a la hora de trabajar con modelos centrados en las 2 primeras capas del modelo OSI (Física y Enlace de Datos); A través de estos entornos es posible obtener métricas como: el total de paquetes (Tx, Rx y Loss), la potencia de la señal, la tasa de bits promedio, Throughput y el SINR (Tian, y otros, 2018).

La propuesta de investigación permitirá determinar el impacto de la interferencia entre los protocolos de las dos tecnologías mencionadas, configurando la capa física de los módulos LoRa y 802.11ah en NS-3 para que interactúen entre sí en un solo escenario de simulación, para extraer las métricas más importantes y analizarlas, a fin de determinar soluciones para la coexistencia de ambos protocolos, dando un paso más cerca de una red heterogénea.

## **1.4 OBJETIVOS**

### **1.4.1 OBJETIVO GENERAL**

Evaluar el comportamiento de las tecnologías IEEE 802.11ah y LoRa dentro de un entorno simulado en NS-3 para determinar el impacto de la interferencia inter-protocolo en el rendimiento de IEEE 802.11ah.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Investigar acerca de las tecnologías IEEE 802.11ah y LoRa, en bases bibliográficas y documentos oficiales de cada estándar, para comprender el funcionamiento y los parámetros técnicos de cada una.
- Modificar el modelo actual de LoRa en NS-3 en su capa física para que permita tomar en cuenta la interferencia debido a otras tecnologías.
- Desarrollar un solo escenario de simulación en NS-3 que contenga una gran cantidad de nodos IEEE 802.11ah y LoRa para simular una red densa.
- Analizar las métricas obtenidas de cada una de las iteraciones de simulación con la finalidad de determinar el impacto de la interferencia inter-protocolo en IEEE 802.11ah.

### **1.5 ALCANCE**

A través de la aplicación del modelo ágil se busca la evaluación del comportamiento de las tecnologías IEEE 802.11ah y LoRa dentro de un entorno simulado en NS-3 para determinar el impacto de la interferencia inter-protocolo en el rendimiento de IEEE 802.11ah, para lo cual se han planteado las siguientes fases para el cumplimiento del proyecto: la fase de requerimientos, que permitirá construir bases teóricas sólidas sobre el funcionamiento de las tecnologías y los parámetros técnicos de los protocolos, fundamentado en bases bibliográficas y documentos oficiales de cada estándar; en la fase de arquitectura y diseño se modificará el modelo actual de LoRa en NS-3, por lo que es necesario un estudio a fondo del entorno de simulación y como se encuentra conformada la arquitectura de los nodos, esto se lo realizará mediante la capacitación y práctica por parte del desarrollador de esta investigación; la fase de desarrollo consistirá en construir un solo escenario de simulación para una red densa, requiriendo para ello de una gran

cantidad de nodos de ambas tecnologías que operen en la misma banda y se encuentren ubicados dentro de un mismo radio determinado; finalmente en la fase de pruebas y realimentación, se analizarán las métricas obtenidas siendo necesario realizar múltiples iteraciones de la simulación variando parámetros claves como: el número de nodos, el radio de distribución y el tamaño de la carga útil (Payload).

Es importante señalar que se llevará a cabo una documentación en paralelo de todas las fases ejecutadas. Para desarrollo del proyecto se hará uso de repositorios bibliográficos, como bases de datos Scopus, IEEEExplore, Springer, entre otras fuentes de referencia confiables y avaladas por asociaciones científica, además de repositorios de código abierto como GitHub.

## **2. CAPÍTULO II: FUNDAMENTO TEÓRICO**

En este capítulo se examina las tecnologías involucradas dentro del proyecto, esto es IEEE 802.11ah y LoRa, para lo cual se ha considerado tanto su evolución como parámetros técnicos y de funcionamiento en las dos primeras capas del modelo OSI (Capa Física y MAC).

### **2.1 ESTÁNDAR 802.11AH**

En el año 1997 se define las características de una WLAN (Wireless Local Area Network) bajo el estándar IEEE 802.11 convirtiéndose en una de las tecnologías inalámbricas más usadas hasta la presente fecha (Sun, Cho, & Sunghyun , 2013). Ahora conocida como Legacy, en un inicio especificaba velocidades de transmisión de 1 y 2 Mbps en la Banda ISM (Industrial, Scientific and Medical) a través de una codificación DSSS (Direct Sequence Spread Spectrum) o FHSS (Frequency Hopping Spread Spectrum), con el paso de los años se han realizados revisiones al estándar para que cumpla las necesidades de los usuarios (Perahia, 2008).

La primera revisión se dio en 1999 bajo el nombre 802.11a, la cual opera en la banda de 5 GHz mediante 52 subportadoras OFDM (Orthogonal Frequency Division Multiplexing). Su velocidad teórica es de hasta 54 Mbps, mientras que su velocidad efectiva ronda los 20 Mbps. En cuanto a su canalización, cuenta con 12 canales no solapados, de los cuales 8 son para red inalámbrica y los 4 restantes para conexiones punto-a-punto. Las desventajas de trabajar en la banda de 5 GHz son las restricciones de los organismos de control y la necesidad de línea de vista. Durante el mismo año se ratificó otra revisión, a la que se llamó 802.11b, la principal diferencia con 802.11a es la banda de operación; trabajando en 2.4 GHz con una velocidad teórica de transmisión de hasta 11 Mbps, su velocidad efectiva varía con el protocolo de red usado, siendo de 5.9 Mbps para TCP y 7.1 Mbps para UDP aproximadamente, la diferencia entre sus

velocidades teórica y efectiva se debe al espacio ocupado por la codificación del protocolo de acceso al medio CSMA/CA.

Cuatro años más tarde (2003) se presentó 802.11g, misma que se define como la evolución inmediata al estándar 802.11b, a pesar de ser una nueva revisión todavía permite una compatibilidad con este último (misma frecuencia de operación), con los cambios realizados puede alcanzar velocidades similares a las de 802.11a (velocidad teórica de hasta 54 Mbps y efectiva de 22 Mbps). La mayoría de los equipos actuales trabajan bajo este estándar, con soporte para 802.11b, sin embargo, en redes con ambos protocolos la velocidad de transmisión se ve limitada.

Por otra parte, con el principal objetivo de proveer una solución a problemas derivados de la coexistencia entre redes 802.11a y sistemas militares (radares y satélites), se presentó en el mismo año la revisión 802.11h, que implementaba una gestión dinámica de la frecuencia y la potencia de transmisión.

Durante los años posteriores (2004 - 2005) se desarrollaron mejoras orientadas a robustecer la seguridad, con 802.11i y su compatibilidad a protocolos como 802.1x, TKIP, WPA y WPA2. Además, se presentaron mejoras en el sistema de control y servicios, como es el caso de 802.11e, responsable de introducir QoS (Quality of Service) mediante clases de tráfico y un sistema de coordinación (HCF) con dos tipos de acceso: EDCA (Enhanced Distributed Channel Access) y HCCA (Hybrid Coordination Function Controlled Access), el primero un sistema de control distribuido basado en prioridades de tráfico y el segundo un sistema de control centralizado con periodos controlados y no controlados.

Tuvo que pasar un tiempo para que se presentasen mejoras, la primera de estas nuevas revisiones es 802.11n y trae consigo la interoperabilidad debido a su capacidad de operar en las bandas de 2.4 GHz y 5 GHz. Además, cuenta con compatibilidad con todas

las anteriores revisiones, alcanza velocidades teóricas de más de 600 Mbps gracias a la tecnología MIMO (Multiple Input – Multiple Output) que le posibilita de utilizar varios canales a la vez para envío y recepción (múltiples antenas).

La segunda mejora es sobre el propio estándar 802.11n, a esta revisión se la denomina como Wi-Fi Gigabit (802.11ac) por la Wi-Fi Alliance, la principal diferencia con su antecesor es la velocidad de transmisión aumentada, trabaja solamente en la banda de 5 GHz y puede alcanzar velocidades teóricas de 1.3 Gbps, esto es posible gracias a la tecnología MIMO con hasta ocho antenas y una modulación 256 QAM (Quadrature Amplitude Modulation).

Finalmente, tras una larga evolución del protocolo y mejoras desarrolladas en cada una de estas etapas, se da paso a la revisión IEEE 802.11ah; la cual es un estándar de carácter emergente para Redes WLAN que se caracteriza por operar en bandas sub-GHz (inferiores a 1 GHz) sin licencia, factor que le permite contar con un alto rango de transmisión a la vez que lo diferencia de IEEE 802.11 (opera en 2,4 GHz y 5 GHz). Cabe mencionar que su capa física se basa en lo presentado por 802.11ac, pero con mejoras substanciales para obtener un mejor rendimiento. La Tabla 1 resume las revisiones expuestas, exaltando sus características más representativas, banda de operación, velocidad de transmisión y fecha de publicación.

Tabla 1. Revisiones de IEEE 802.11.

<b>Revisión</b>	<b>Detalle</b>	<b>Banda</b>	<b>Velocidad</b>	<b>Publicación</b>
802.11	Legacy	2.4 GHz	1 – 2 Mb/s	1997
802.11a	Banda de 5 GHz	5 GHz	54 Mb/s	1999
802.11b	Gran aceptación comercial	2.4 GHz	11 Mb/s	1999
802.11g	Revisión de 802.11b	2.4 GHz	54 Mb/s	2003
802.11h	Revisión de 802.11a	5 GHz	54 Mb/s	2003
802.11i	Mejora en la seguridad			2004

802.11e	Introduce QoS			2005
802.11n	Introduce MIMO	2.4 - 5 GHz	> 600 Mb/s	2009
802.11ac	Revisión 802.11n	5 GHz	> 1.3 Gb/s	2011
802.11ah	Banda sub-GHz	< 1 GHz		2011

Fuente: Autor.

### 2.1.1 ARQUITECTURA 802.11AH

El estándar IEEE 802.11ah ha sido creado para mejorar el rendimiento del sistema, razón por la cual se diseñó una nueva capa física o PHY, misma que se basa en IEEE 802.11ac. Además, IEEE 802.11ah cuenta con mejoras en la subcapa de Control de Acceso al Medio o MAC, estos cambios hacen de esta tecnología una opción atractiva para grandes redes de sensores.

La Figura 1 muestra la arquitectura básica de una red IEEE 802.11ah, la cual consta de nodos sensores o estaciones, un nodo central o Punto de Acceso y Puntos de Acceso intermedios o AP Relays. El rango de cobertura de este tipo de red es de hasta 1000m, donde cada nodo se comunica hacia el Punto de Acceso a través de un solo salto (single-hop), sin embargo, es posible extender el rango de cobertura usando AP Relays entre los nodos sensores y el AP Central. Los datos obtenidos de los sensores son enviados a Internet a través de una red de retorno o backhaul.

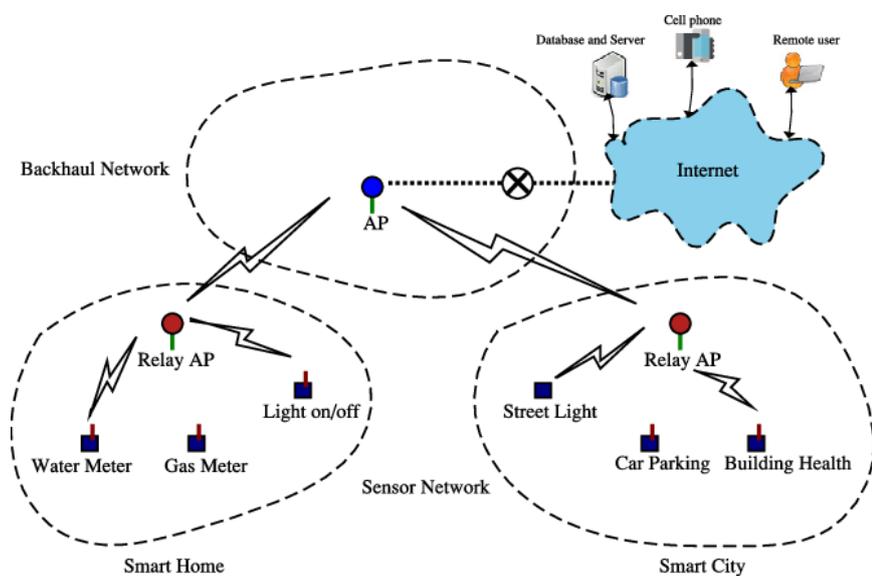


Figura 1. *Arquitectura de una red 802.11ah.*  
Fuente: (Ahmed, Rahman, & Hussain, *A comparison of 802.11ah and 802.15.4 for IoT*, 2016).

### **2.1.1.1 CAPA FÍSICA PHY 802.11AH**

IEEE 802.11ah opera sobre un conjunto de bandas de radio (todas por debajo de 1 GHz) que dependen de regulaciones del país, de modo que no se trabajará con las mismas frecuencias en Europa que en Estados Unidos o Japón. Los anchos de banda de canal que se han adoptado más fácilmente son los de 1 MHz y 2 MHz, aunque en algunos países también se permiten y utilizan configuraciones más amplias que utilizan 4, 8 y 16 MHz. La transmisión PHY utiliza una forma de onda basada en OFDM que consta de 32 o 64 tonos/subportadoras, con un espaciado de 31,25 kHz entre cada subportadora. Las modulaciones admitidas por el protocolo incluyen BPSK (By Phase Shift Keying), QPSK (Quadrature Phase Shift Keying), y QAM (Quadrature Amplitude Modulation). El desarrollo de IEEE 802.11ah tiene como base a IEEE 802.11ac, esto se ve reflejado en el uso de tecnologías como beamforming (conformación de haces para aumentar la intensidad de la señal con relación al ruido y posición del dispositivo final), MIMO (Adame, Bel, Bellata, Barcelo, & Oliver, 2014).

#### **2.1.1.1.1 CANALIZACIÓN**

Si bien se ha mencionado que IEEE 802.11ah se basa en IEEE 802.11ac, existe una importante diferencia entre las frecuencias de operación de cada uno, mientras IEEE 802.11ac define canales de frecuencia con anchos de banda de 20, 40, 80 y 160 MHz, su homólogo 802.11ah trabaja con canales de 1, 2, 4, 8 y 16 MHz de ancho de banda. La característica que si se mantiene en ambos estándares es la capacidad de combinar canales estrechos adyacentes para obtener un canal final más ancho, (IEEE 802.11 Working Group, 2016). En la Figura 2 están representados los anchos de banda de canal que utiliza IEEE 802.11ah. Los canales más cortos (1 MHz) se denominan **primary1**, dos de estos canales cortos adyacentes forman un canal con mayor ancho de banda (2 MHz)

**primary2**, este formato es usado para denominar a los dos canales principales disponibles de 1 y 2 MHz, no obstante, los canales más anchos son denominados **secondary4** y **secondary8** para 4 y 8 MHz de ancho de banda respectivamente (IEEE 802.11 Working Group, 2016).

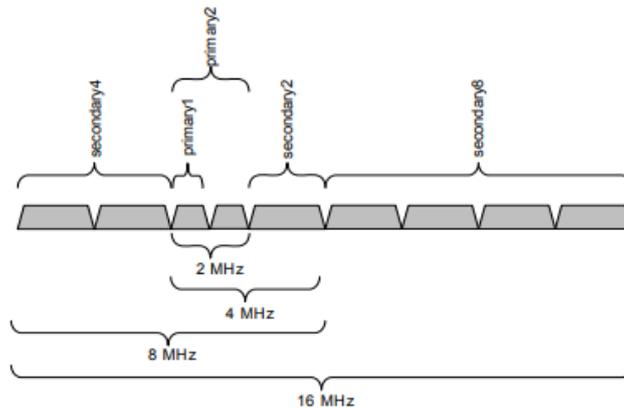


Figura 2. Canales según su ancho de banda para 802.11ah.  
Fuente: (IEEE 802.11 Working Group, 2016).

Las bandas sub-GHz varían de acuerdo con los organismos regulatorios de cada país o continente, tal como se muestra en la Figura 3, para Estados Unidos de acuerdo con la FCC (Federal Communications Commission) se dispone la banda entre 902 MHz y 928 MHz, es decir un total de 26 MHz; para Korea se pone a disposición la banda comprendida entre 917.5 MHz y 923.5 MHz, que es considerablemente menor a la de US; en Europa difiere al tener asignada una banda entre 863 MHz y 868 MHz; mientras que en China se ha establecido desde 755 MHz hasta 787 MHz; en Japón se usa al banda entre 916.5 MHz y 927.5 MHz (Keysight, 2017).

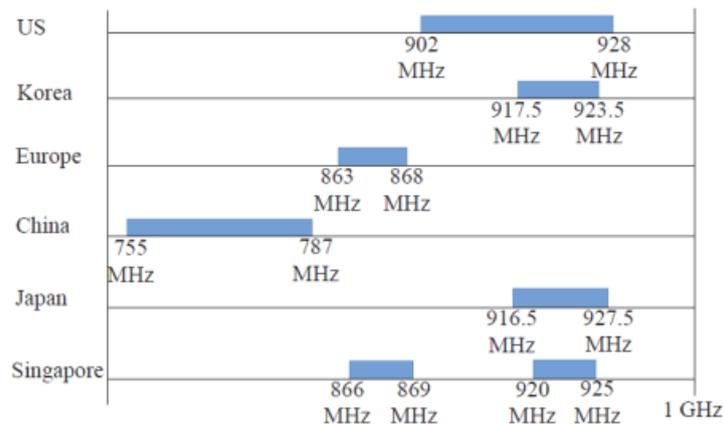


Figura 3. Asignación de canales para varios países.  
Fuente: (Keysight, 2017).

### 2.1.1.1.2 MODOS DE OPERACIÓN

El diseño de la capa PHY se puede clasificar siguiendo el criterio del modo de transmisión, para un canal con ancho de banda de 1 MHz (S1G\_1M) o para un canal igual o mayor a 2MHz.

El modo de operación con un ancho de banda de canal de 1 MHz se usa en aplicaciones que necesita de un rango extendido de operación, se sacrifica ancho de banda y velocidad de transmisión a cambio de rango. Es un modo adecuado para aplicaciones IoT que envía ráfagas de datos y no requiere de una velocidad de datos alta. El espacio de subportadora usado por este modo es 31.25 kHz y el número de subportadoras de datos por símbolo OFDM es 24, que es mucho menor al número de subportadoras para el canal de 2 MHz. La Tabla 2 muestra el MCS (Modulation and Coding Scheme) o Esquema de Modulación y Codificación para un canal de 1 MHz con un solo flujo espacial o spatial stream<sup>1</sup> (Nss), la velocidad de los datos dependerá del Intervalo de Guarda (GI) y tiene como objetivo prevenir la Interferencia Inter Símbolo (ISI).

<sup>1</sup> El parámetro Nss hace referencia a la cantidad de spatial streams o flujos espaciales, es decir, cada antena envía un flujo espacial, entre más antenas más flujos espaciales se podrán usar, posibilitando velocidades de transmisión sumamente altas.

Tabla 2. S1G MCSs para un canal de 1 MHz con Nss = 1.

MCS Idx	Mod	R	N_bpscs	N_sd	N_sp	N_cbps	N_dbps	N_es	Data_rate (kbps)	
									8 $\mu$ s GI	4 $\mu$ s GI
0	BPSK	1/2	1	24	2	24	12	1	300.0	333.3
1	QPSK	1/2	2	24	2	48	24	1	600.0	666.7
2	QPSK	3/4	2	24	2	48	36	1	900.0	1000.0
3	16-QAM	1/2	4	24	2	96	48	1	1200.0	1333.3
4	16-QAM	3/4	4	24	2	96	72	1	1800.0	2000.0
5	64-QAM	2/3	6	24	2	144	96	1	2400.0	2666.7
6	64-QAM	3/4	6	24	2	144	108	1	2700.0	3000.0
7	64-QAM	5/6	6	24	2	144	120	1	3000.0	3333.3
8	256-QAM	3/4	8	24	2	192	144	1	3600.0	4000.0
9	256-QAM	5/6	8	24	2	192	160	1	4000.0	4444.4
10	BPSK	1/2 with 2x repetition	1	24	2	24	6	1	150.0	166.7

Fuente: (IEEE 802.11 Working Group, 2016).

En cuanto al otro modo de operación, con anchos de banda desde 2 MHz, utiliza OFDM acompañado de un diseño con una longitud de símbolo diez veces mayor que en 802.11ac (una décima parte del reloj de 802.11ac). Junto con la tecnología MIMO es posible obtener velocidades de hasta casi 347 Mbps, como se detalla en el esquema mostrado en la Tabla 3.

Tabla 3. S1G MCSs para un canal de 16 MHz con Nss = 4.

MCS Idx	Mod	R	N_bpscs	N_sd	N_sp	N_cbps	N_dbps	N_es	Data_rate (kbps)	
									8 $\mu$ s GI	4 $\mu$ s GI
0	BPSK	1/2	1	468	16	1872	936	1	23400.0	26000.0
1	QPSK	1/2	2	468	16	3744	1872	1	46800.0	52000.0
2	QPSK	3/4	2	468	16	3744	2808	1	70200.0	78000.0
3	16-QAM	1/2	4	468	16	7488	3744	1	93600.0	104000.0
4	16-QAM	3/4	4	468	16	7488	5616	1	140400.0	156000.0
5	64-QAM	2/3	6	468	16	11232	7488	1	187200.0	208000.0
6	64-QAM	3/4	6	468	16	11232	8424	1	210600.0	234000.0
7	64-QAM	5/6	6	468	16	11232	9360	1	234000.0	260000.0
8	256-QAM	3/4	8	468	16	14976	11232	1	280800.0	312000.0
9	256-QAM	5/6	8	468	16	14976	12480	1	312000.0	346666.7

Fuente: (IEEE 802.11 Working Group, 2016).

Para comprender el MCS es necesario definir cada uno de los parámetros que constituyen las tablas antes presentadas, donde:

- MCS Idx: es el índice del esquema de modulación y codificación, van numerados en forma ascendente del 0 al 10 lo cual se refleja en un aumento

de velocidad de transmisión, cada índice maneja parámetros diferentes (modulación, ratio de codificación, entre otros).

- Mod: indica el tipo de modulación manejado por el índice de MCS, estos pueden ser BPSK (modulación por desplazamiento de fase), QPSK (modulación por desplazamiento de fase en cuadratura), QAM (modulación de amplitud en cuadratura).
- R: es la tasa de codificación, es decir, la proporción de bits transmitidos que sí llevan datos (no cuenta cabeceras o trailers).
- N\_bpscs: es el número de bits codificados por portadora en cada flujo espacial (spatial stream).
- N\_sd: cantidad de números de datos complejos por segmento de frecuencia.
- N\_sp: número de valores piloto por segmento de frecuencia.
- N\_cbps: representa el número de bits codificados por símbolo.
- N\_dbps: representa el número de bits de datos codificados por símbolo.

Conociendo los parámetros descritos es posible calcular la velocidad de transmisión para una determinada duración del intervalo de guarda o GI, este último puede ser de 8 microsegundos ( $\mu\text{s}$ ) o 4  $\mu\text{s}$ , dependiendo de, si se trata de un intervalo de guarda largo (normal) o corto respectivamente. Pero antes de calcular la velocidad de transmisión, se obtiene la duración del símbolo OFDM mediante la Ec. 1 (IEEE 802.11 Working Group, 2016).

$$T_{SYM} = T_{DFT} + T_{GI} \quad (\text{Ec. 1})$$

Donde,  $T_{DFT}$  representa el periodo de la DFT (Discrete Fourier Transform) o de la IDFT (Inverse Discrete Fourier Transform ) y viene definido como  $T_{DFT} = 1/\Delta_F = 32\mu\text{s}$  ; siendo  $\Delta_F$  el espaciado de frecuencia entre subportadoras e igual a 31.25 kHz.

Reemplazando los valores del periodo de la DFT y el Intervalo de Guarda Largo en la Ec. 1, se obtiene el tiempo de símbolo para un GI largo.

$$T_{SYML} = T_{DFT} + T_{GI_L} = 32\mu s + 8\mu s = 40\mu s$$

Se hace lo propio para el Intervalo de Guarda corto y se calcula la duración del símbolo OFDM.

$$T_{SYMS} = T_{DFT} + T_{GI_S} = 32\mu s + 4\mu s = 36\mu s$$

Ahora bien, para calcular la velocidad de transmisión obtenida de cada uno de los índices de MCS se deberán seguir los siguientes pasos:

1. Obtener el número de subportadoras de acuerdo con ancho de banda del canal (1, 2, 4, 8 o 16 MHz).
2. Obtener el número de bits por subportadora o **N\_bpscs**, según cada índice de MCS puede tomar un valor de 1, 2, 4, 6 u 8.
3. Calcular el número de bits codificados por símbolo o **N\_cbps**, multiplicando lo obtenido en los pasos 1 y 2, es decir,  **$N_{cbps} = N_{sd} * N_{bpscs}$** .
4. Calcular el número de bits de datos por símbolo o **N\_dbps**, multiplicando lo obtenido en el paso 3 por la tasa de codificación, es decir,  **$N_{dbps} = N_{cbps} * R$** .
5. Finalmente, para calcular la velocidad de transmisión del índice MCS determinado bastará con dividir el **N\_dbps** por la duración del símbolo OFDM para un GI normal o corto, tal como consta en la Ec. 2.

$$Data\_rate = \frac{N_{dbps}}{T_{SYM}} \quad (Ec. 2)$$

### 2.1.1.1.3 MODELOS DE PROPAGACIÓN

Los modelos de propagación se tratan de modelos matemáticos de carácter experimental, desarrollados para la caracterización de ondas de radio en función de la distancia, frecuencia u otras condiciones adicionales. En otras palabras, un modelo que sea capaz de anticipar el comportamiento de enlaces bajo condiciones idénticas, generalmente el valor que predicen es el Path Loss o atenuación de trayectoria dentro del área de cobertura efectiva del transmisor. En este sentido se puede aplicar varios modelos para IEEE 802.11ah tales como: Lee y Hata.

El modelo Lee, que fue diseñado para señales que operan alrededor de 900 MHz, además cuenta con factores para ajustar diferentes frecuencias, ganancias, altura de antenas, etc. De acuerdo con (Lazarakis, Dangakis, & Kostarakis, 1997), la Ec. 3 describe el comportamiento matemático de este modelo.

$$L(dB) = L_o + \gamma \log_{10}(d) - 10 \left( \log_{10} F_A - 2 \log_{10} \left( \frac{H_{ET}}{30} \right) \right) \quad (\text{Ec. 3})$$

Donde,  $L_o$  es el valor de atenuación referencial en 1 Km,  $\gamma$  es un parámetro ambiental de la pendiente de la curva de pérdida de trayectoria (medido en decibelios por década),  $d$  es la distancia en que se quiere calcular en Km,  $F_A$  es un factor de ajuste para la altura de la antena, ganancia y frecuencia, finalmente  $H_{ET}$  es la altura efectiva del terreno en metros.

El modelo Hata, usa parámetros similares a los del modelo Lee excepto por aquellos que permiten ajustar valores ( $F_A$ ), haciendo de este un modelo más simple y práctico (Hata, 1980). Sin embargo, este modelo puede ser aplicado para una variedad de frecuencias S1G o sub-GHz, la Ec. 4 define matemáticamente este modelo.

$$L(dB) = L_o + k_1 * \log_{10}(f_c) + k_2 * \log_{10}(h_{Tx}) \log_{10}(d) \quad (\text{Ec. 4})$$

Donde,  $L_o$  es el valor de atenuación referencial en 1 Km, tanto  $k_1$  como  $k_2$  son constantes,  $f_c$  es la frecuencia de la portadora en MHz,  $h_{Tx}$  es la altura a la que se encuentra el transmisor en metros y  $d$  es la distancia a la que se quiere calcular en metros.

Si bien estos modelos descritos funcionan adecuadamente para la banda de S1G, el grupo de trabajo de IEEE 802.11 ha desarrollado sus propios modelos de propagación para zonas urbanas, el primer modelo llamado “despliegue macro urbano” contempla una antena con altura de 15 metros sobre techo y su path loss se mide en decibelios, en base a (Erceg, 2004) se contempla la expresión matemática de la Ec. 5.

$$L(dB) = 8 + 37.6 * \log_{10}(d) \quad (\text{Ec. 5})$$

Al igual que los modelos anteriores  $d$  es la distancia a la que se quiere calcular en metros; debido a la simplicidad de la fórmula esta solo funciona para una frecuencia de portadora de 900MHz, en caso de trabajar en una frecuencia diferente se deberá usar un factor de corrección, la Ec. 6 define como calcular este factor.

$$\delta = 21 * \log_{10} \left( \frac{f}{900 \text{ MHz}} \right) \quad (\text{Ec. 6})$$

El otro modelo se conoce como “despliegue pico/hotzone urbano” y a diferencia de su análogo, la altura de la antena está ajustada a 2 metros al nivel del techo, es decir, en una zona más concurrida o “caliente”, la Ec. 6 representa este modelo matemático.

$$L(dB) = 23.3 + 37.6 * \log_{10}(d) \quad (\text{Ec. 7})$$

Tanto Ec. 5 como Ec. 7 comparten similitudes en sus modelos matemáticos, no obstante, se puede notar a simple vista que las pérdidas en el modelo de despliegue hotzone urbano serán mayores.

#### 2.1.1.1.4 FORMATO DEL PHY PPDU

Durante una transmisión, cuando al PSDU (PLCP Service Data Unit) es procesado y se le agrega el preámbulo PHY se origina lo que se conoce como PPDU (PLCP Protocol

Data Unit). Este preámbulo le será de vital utilidad al receptor, ya que será procesado para ayudar a la detección, demodulación y entrega del PSDU. En este sentido, es importante conocer el formato del PPDU que maneja IEEE 802.11ah.

De acuerdo con el estándar oficial se definen tres tipos de formatos de PPDU, nombrados como: S1G\_SHORT, S1G\_LONG, and S1G\_1M. Como paréntesis, dentro de los archivos del módulo IEEE 802.11ah para NS-3, se encuentran agregadas las definiciones de estos formatos de preámbulo, junto a los formatos del resto de tecnologías IEEE 802.11 soportadas (ver Figura 4).

```
enum WifiPreamble
{
    WIFI_PREAMBLE_LONG,
    WIFI_PREAMBLE_SHORT,
    WIFI_PREAMBLE_HT_MF,
    WIFI_PREAMBLE_HT_GF,
    WIFI_PREAMBLE_NONE,
    WIFI_PREAMBLE_S1G_SHORT,
    WIFI_PREAMBLE_S1G_LONG,
    WIFI_PREAMBLE_S1G_1M
};
```

Figura 4. Formatos de preámbulo para IEEE 802.11ah dentro del simulador.  
Fuente: Github (Tian, y otros, 2018)

La estructura general para S1G\_SHORT está definida en la Figura 5, este formato se utiliza para transmisiones de tipo SU (Single-User) con PPDU de 2 MHz, 4 MHz, 8 MHz y 16 MHz.

Tanto este como los otros formatos de PPDU mencionados anteriormente se componen de los siguientes campos:

- STF: Short Training Field, consta de 2 (S1G\_short) o 4 (S1G\_1M) símbolos OFDM (80µs o 160µs), se pueden utilizar para temporización aproximada, estimación de sincronización y desplazamiento de frecuencia.
- LTF1: First Long Training Field, consta de 2 (S1G\_short) o 4 (S1G\_1M) símbolos OFDM, utilizado para sincronización de tiempo fina, estimación de desplazamiento de frecuencia fina y estimación de canal single stream.

- SIG: Signal Field, consta de 2 (SIG\_short) o 6 (SIG\_1M) símbolos OFDM, es decodificado por el receptor para determinar los parámetros de transmisión, como MCS, NSS, etc.
- LTF2: Subsequent Long Training Fields, puede existir más de uno y consisten en símbolos OFDM de orden  $NLTF - 1$ , se utiliza para la estimación del canal MIMO, cuando  $NSS = 1$  este campo no existe.
- DATA: el campo de datos que contiene el Payload del usuario.

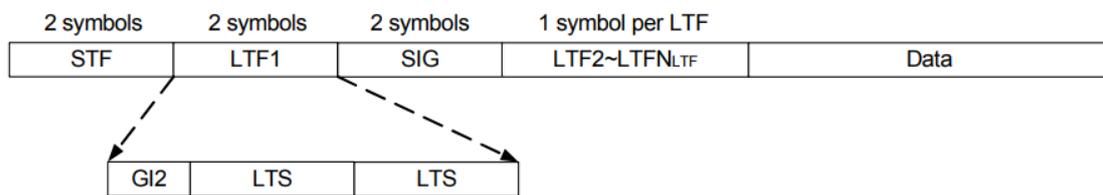


Figura 5. Formato S1G\_SHORT.  
Fuente: (IEEE 802.11 Working Group, 2016).

La estructura para un PDU del tipo S1G\_LONG se encuentra representada en la Figura 6, difiere de la estructura vista en la Figura 5 al contar con una porción omnidireccional y una porción “beamchangeable” o haz intercambiable. Por ende, este formato de trama puede ser usado tanto para transmisiones SU (Single-User) como para MU (Multiplie-User) de 2 MHz, 4 MHz, 8 MHz y 16 MHz.

Los campos que conforman este formato son los mismos que S1G\_SHORT, a excepción del campo D-STF, que corresponde al Short Training field para la porción de haz intercambiable.

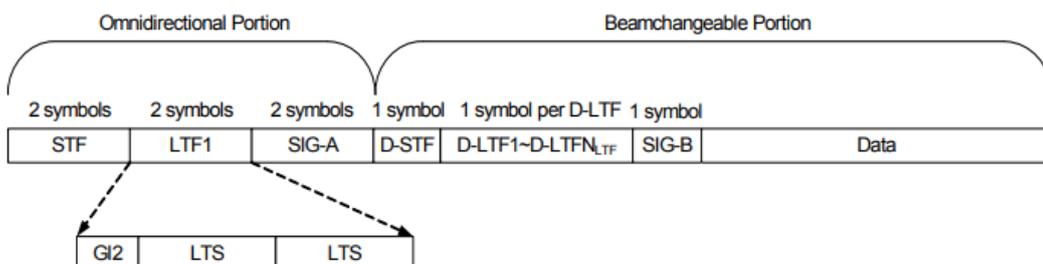


Figura 6. Formato S1G\_LONG.  
Fuente: (IEEE 802.11 Working Group, 2016).

El tercer y último formato de PPDU corresponde al S1G\_1M, que se define en la Figura 7. Este formato es utilizado para transmisiones SU (Single-User) de 1MHz únicamente.

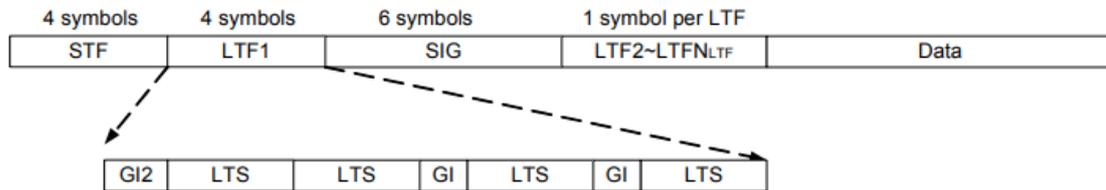


Figura 7. Formato S1G\_1M.  
Fuente: (IEEE 802.11 Working Group, 2016).

La definición de cada uno de los campos que conforman los distintos formatos de PPDU ya mencionados se encuentra en la Tabla 4, que ha sido extraída directamente del IEEE Std 802.11ah™-2016.

Tabla 4. Campos que componen el PHY PPDU.

Field	Description
STF	Short Training field
LTF	Long Training field
SIG	SIGNAL field
SIG-A	Signal A field
D-STF	Short Training field for the beam changeable portion
D-LTF	Long Training field for the beam changeable portion
SIG-B	Signal B field
Data	The Data field carries the PSDU(s)
GI	Guard interval
GI2	Double guard interval
LTS	Long training symbol

Fuente: (IEEE 802.11 Working Group, 2016).

### 2.1.1.2 CAPA MAC 802.11AH

La capa MAC está diseñada para maximizar el número de estaciones admitidas por la red, garantizando al mismo tiempo un consumo energético mínimo. IEEE 802.11ah define tres tipos diferentes de estaciones, cada una con diferentes procedimientos y horarios programados para acceder al canal, estas estaciones son: TIM, no-TIM y no

programadas. Las estaciones TIM necesitan escuchar los beacons enviados por el AP para poder enviar o recibir datos, la transmisión de información se realiza a través de una RAW (Restricted Access Window). Por otro lado, las estaciones no-TIM no necesitan escuchar los beacons para transmitir datos, en su lugar, durante el proceso de asociación negocian directamente con el AP para obtener un tiempo de transmisión asignado dentro de una PRAW (Periodic Restricted Access Window). Similar a esta última, las estaciones no programadas no necesitan escuchar beacons, en cambio, envían una trama de Poll hacia el AP solicitando acceso inmediato al canal, la respuesta del Punto de Acceso indicará un intervalo fuera de cualquier RAW para que pueda acceder al canal (Adame, Bel, Bellata, Barcelo, & Oliver, 2014).

#### **2.1.1.2.1 SOPORTE A GRAN CANTIDAD DE ESTACIONES**

Para una red 802.11, se identifica dos tipos de nodos: uno que hace de Punto de Acceso o AP (funciones de control) y uno que solamente hace de Estación o STA. Un AP tiene entre sus funciones la de asignar un AID (Association ID) a cada estación que se haya asociado con dicho dispositivo. Para un dispositivo 802.11 común y corriente, el número máximo de identificadores asignables es de 2007, mientras que para 802.11ah este número es mayor gracias a la estructura jerárquica de un AID, como se muestra en la Figura 8, es posible incluir múltiples estaciones en un solo bloque o sub-bloque. Un AID al estar compuesto por 13 bits permite expresar hasta  $2^{13} = 8191$  estaciones.

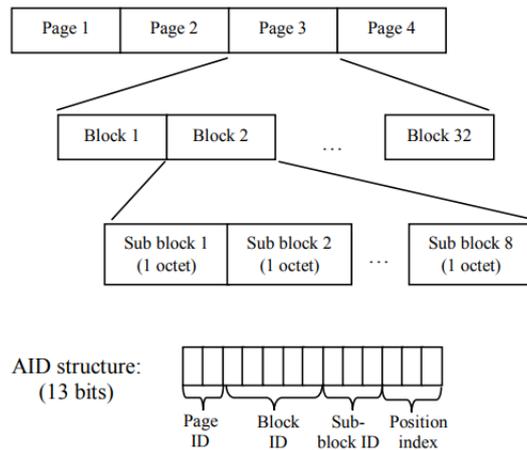


Figura 8. Estructura jerárquica de una AID.  
Fuente: (Yuan, Haiguang, Shoukang, & Zander, 2013).

Esta capacidad de granularidad que ofrece 802.11ah abre la posibilidad de agrupar estaciones de acuerdo con su patrón de tráfico, localización o necesidades con tal de optimizar la red (recurso inalámbrico).

#### 2.1.1.2.2 AHORRO DE ENERGÍA

IEEE 802.11ah incluye un modo de ahorro de energía para reducir el consumo de energía de los dispositivos finales. Este mecanismo saca provecho del comportamiento inherente de la red, desactivando el módulo RF (Radio Frequency) durante los períodos sin tráfico.

En el Modo activo, la estación mantiene encendidos sus componentes de RF (Radio Frecuencia) de manera continua, lo que le permite detectar señales entrantes, permaneciendo en un estado de “vigilia”.

Mientras que el modo de ahorro de energía le permite al dispositivo permanecer inactivo durante un periodo de tiempo **BSSMaxIdlePeriod**, pasado este periodo la estación quedará desasociada de la red. Para IEEE 802.11 Legacy, el campo de **BSSMaxIdlePeriod** es de 16 bits y maneja unidades de 1024 ms, dando como resultado

un tiempo máximo de inactividad de 18.64 horas<sup>2</sup>. Por otro lado, IEEE 802.11ah trabaja con un campo **BSSMaxIdlePeriod** de 14 bits, sin embargo, maneja diferentes periodos para diferentes aplicaciones (1, 10, 10<sup>3</sup>, 10<sup>4</sup>) permitiendo que un dispositivo pueda permanecer inactivo hasta por 5 años<sup>3</sup>. Una estación en este modo deberá “despertar” o pasar a modo activo de manera periódica para poder recibir la trama de beacon, esta le indicará si existen paquetes almacenados temporalmente con su destino, en caso de que esto sea afirmativo, la STA deberá transmitir una trama de control PS (Power Saving) hacia el AP, solicitando la entrega de los paquetes almacenados temporalmente. Una vez que estos paquetes sean recibidos, la STA volverá al modo de ahorro de energía.

En la Figura 9 se observa el comportamiento de una EL STA (Energy Limited Station), es decir una estación que trabaja en el modo de ahorro de energía, cada cambio o pico en la gráfica indica que la STA ha cambiado de estado en un momento de tiempo programado **ELMaxAwakeTimer**, para luego volver a un estado de reposo en un tiempo **ELRecoveryTimer**. En la parte inferior de la Figura 9 se encuentran las tramas enviadas, la de Power Saving se encuentra a la par del cambio de estado de la EL STA (de inactivo a activo), una vez enviado el ACK por parte de la EL STA se pasa al modo de ahorro de energía.

---

<sup>2</sup> Las 18.64 horas de tiempo de inactividad son obtenidas de:  $1.024 [s] * (2^{16} - 1) = 67107.84 [s] * 1[h] / 3600[s] = 18.64 [h]$ .

<sup>3</sup> Los 5 años de tiempo de inactividad son obtenidos de:  $10^4 [s] * (2^{14} - 1) = 163830000 [s] * 1[h] * 1[días] * 1[años] / 3600[s] * 24[h] * 365[días] = 5.19 [años]$ .

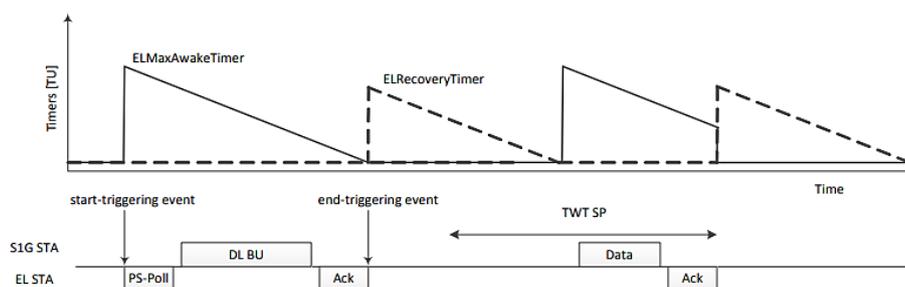


Figura 9. STA con energía limitada.  
Fuente: (IEEE 802.11 Working Group, 2016).

### 2.1.1.2.3 TIM Y SEGMENTACIÓN DE PÁGINA

Para reducir el tiempo que una estación compite por el canal, así como para aumentar su tiempo de inactividad (ahorro de energía), IEEE 802.11ah utiliza un esquema llamado TIM (Traffic Indication Map) y Segmentación de página. Por ende, la distribución jerárquica de estaciones en grupos (vista en la sección 2.1.1.2) se utiliza no solo con el fin de organizar sino también para programar la señalización y asignar los recursos de canal disponibles a los diferentes grupos TIM. El estándar restringe a un grupo de estaciones TIM en particular a competir simultáneamente por el mismo canal en un período específico. Estas estaciones solo necesitan competir entre ellas y escuchar sus beacons TIM. Este mecanismo de acceso permite que las estaciones TIM de un mismo grupo permanezcan en modo inactivo por el resto del tiempo (luego del periodo de contención), lo que se traduce en un ahorro considerable de energía.

El sistema de señalización usado es una extensión del sistema de señalización IEEE 802.11 Legacy; Además de utilizar beacons TIM para señalización a nivel de estaciones, también utiliza beacons DTIM (Delivery Traffic Indication Message) para señalización a nivel de grupo TIM. Estos últimos se utilizan para señalar qué grupos TIM tienen datos pendientes en el AP. Además, contiene toda la información sobre las propiedades de la Ventana de Acceso Restringido, como la duración de los segmentos, los mecanismos de sub-slots, etc. Por su parte, los beacons TIM pagan un solo grupo

TIM con al menos una estación que tiene datos pendientes en el AP (IEEE 802.11 Working Group, 2016).

Con este mecanismo, cualquier estación puede entrar en un estado de ahorro de energía durante todo el período de la Ventana de Acceso Restringido si no tiene un paquete para transmitir y se cumple al menos una de las dos condiciones siguientes:

- Identifica en el beacon DTIM que no hay tráfico de downlink (enlace descendente) dirigido a su grupo TIM.
- Identifica en el beacon DTIM que hay tráfico de downlink (enlace descendente) dirigido a su grupo TIM, pero no aparece explícitamente en su beacon TIM.

#### **2.1.1.2.4 MECANISMOS DE ACCESO AL CANAL**

El mecanismo usado para acceder al medio varía según el tipo de estación, en la sección 2.1.1.2 se introdujeron los conceptos de estaciones TIM, no-TIM y no programadas. Para una estación TIM (Traffic Indicator Map) se combina un sistema de asignación de período de tiempo desde un AP central con DCF<sup>4</sup> (Distributed Coordination Function). En la Figura 10 está representada la distribución Ventanas de Acceso Restringido, el periodo de tiempo entre TIMs consecutivos contiene una RAW conformada por un segmento de multicast (multidifusión), un segmento de downlink (enlace descendente), un segmento de uplink (enlace ascendente).

---

<sup>4</sup> DCF utiliza un algoritmo CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) con intercambio RTS/CTS opcional y acuse explícito de recibo.

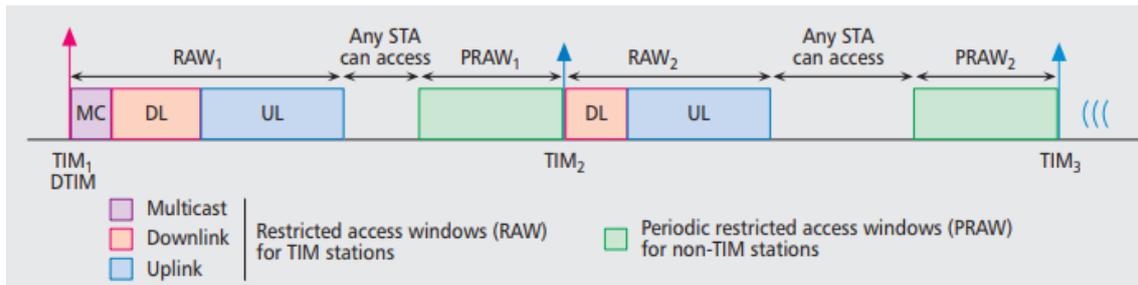


Figura 10. Distribución de RAW y PRAW entre beacons.  
Fuente: (Adame, Bel, Bellata, Barcelo, & Oliver, 2014).

Para un enlace de downlink (cuando un AP tiene un paquete para enviar a una estación), el beacon DTIM debe incluir en su mapa de bits el grupo TIM al que pertenece esa estación. Luego, el beacon TIM correspondiente incluye esa estación en su mapa de bits. Cada estación señalizada debe escuchar su beacon TIM para saber cuándo comenzar a competir por el medio. Esta contención se realizará utilizando DCF. Cuando expira el backoff de una estación, envía una trama PS-Poll para obtener sus datos correspondientes.

De igual manera, para un enlace de uplink, siempre que una estación quiera enviar un mensaje de enlace ascendente al AP, primero debe escuchar a su grupo TIM correspondiente para saber cuándo competir por el canal. La contención se realiza, como en la transmisión de downlink (a través del esquema DCF). Se pueden utilizar tanto los mecanismos básicos de acceso como de protocolo de enlace (RTS / CTS).

El segundo tipo de estación es la no-TIM, para estas el AP puede permitirles tanto solicitar tráfico de downlink almacenado en el búfer como transmitir tráfico de uplink en cualquier momento al despertarse. Sin embargo, al operar de una manera tan arbitraria, estas estaciones que no son TIM podrían incurrir en una gran cantidad de tráfico incontrolable, lo que probablemente degradaría el rendimiento de la red. Por ejemplo, si un gran número de estaciones se despierta al mismo tiempo, la disputa entre estas estaciones podría resultar en demoras excesivas en el acceso a los canales o incluso en colisiones.

Con el fin de controlar el tráfico de las estaciones no-TIM, el AP programa un tiempo definido para que cada STA despierte y sus intentos de acceso al canal se distribuyan. Para intercambiar esta información de tiempo de activación entre AP y estaciones, IEEE 802.11ah ha definido un IE (Information Element) denominado TWT IE (Target Wake Time o Tiempo de Activación Objetivo) que se intercambia en las tramas de asociación (solicitud/respuesta). Un TWT IE cuenta con cuatro campos principales: tipo de solicitud, tiempo de activación objetivo, duración mínima de activación y mantisa de intervalo de activación, estas se utilizan para determinar cuándo y con qué frecuencia una estación se despierta para realizar transmisiones de uplink/downlink. Los campos mencionados se pueden identificar en la Figura 11, que define el formato de un elemento TWT. Su tamaño puede ser de hasta 30 bytes, cuenta con un campo de ID [ElementID], un campo de longitud [Length] (puede variar desde 9 a 30 bytes), un campo de control [Control], uno para el tipo de solicitud [Request Type], tiempo de activación objetivo [TargetWakeTime], uno para la asignación de grupo [TWTGroupAssignment], duración mínima de activación [NominalMinimumTWTWakeDuration], mantisa de intervalo de activación [TWTWakeIntervalMantissa], canal TWT asignado [TWTChannel] y un campo opcional para paginado [NDPPaging].

	Element ID	Length	Control	Request Type	Target Wake Time	TWT Group Assignment	Nominal Minimum TWT Wake Duration	TWT Wake Interval Mantissa	TWT Channel	NDP Paging (optional)
Octets:	1	1	1	2	8 or 0	9 or 3 or 0	1	2	1	0 or 4

Figura 11. Formato de un elemento TWT.  
Fuente: (IEEE 802.11 Working Group, 2016).

Cuando existen paquetes almacenados en búfer para una estación no-TIM, el AP puede enviar a la STA una nueva trama de control llamada trama de paginación NDP (Null Data Packet o Paquetes de Datos Nulos) en su tiempo de activación objetivo, esta

trama contiene la información del estado de almacenamiento en búfer. Si la estación reconoce la existencia de paquetes almacenados en búfer después de recibir con éxito la trama de paginación NDP, puede solicitar la entrega de los paquetes almacenados en búfer transmitiendo una trama de PS-Poll. No obstante, si el AP no transmite la trama de paginación NDP a la hora de activación objetivo (TWT), la estación puede transmitir la trama de enlace ascendente si el canal está inactivo.

#### **2.1.1.2.5 VENTANA DE ACCESO RESTRINGIDO**

El estándar IEEE 802.11ah define un nuevo mecanismo libre de contención para acceder al canal, llamado RAW (Restricted Access Window). Con el fin de reducir colisiones y mejorar la eficiencia del canal, el AP se encarga de coordinar el acceso al canal de uplink o subida, solamente dispositivos determinados tendrán acceso exclusivo al medio compartido y estos se organizarán en diferentes grupos de acuerdo con su RAW asignada.

Un AP puede asignar una o más RAWs para un grupo de estaciones dentro de un intervalo de beacon y difundir esta información utilizando el elemento de información RPS (RAW Parameter Set) transportado en el beacon. El elemento RPS (ver Figura 13) especifica qué estaciones se asignan a cada RAW (ver Figura 10), la configuración de la RAW y su hora de inicio en relación con la hora de finalización del beacon (en caso de ser la primera ventana) o la RAW anterior. Tal como se observa en la Figura 12, una RAW se puede dividir en ranuras de tiempo denominadas RAW slots, y las estaciones asignadas a cada RAW se asignan uniformemente a estos slots mediante round robin (asignación por turnos). Si una estación pertenece a un grupo RAW, se le permite competir por el acceso al medio al comienzo de su ranura RAW asignada y no competirá por el acceso al medio dentro de ningún otro slot durante ese RAW. Adicionalmente, el

número de ranuras denominado NRAW, el formato de ranura y el **Slot Duration Count C** también se especifican en el elemento RPS (ver Figura 16).

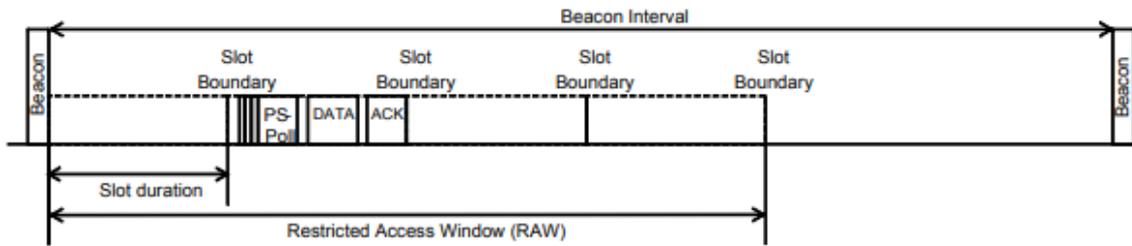


Figura 12. Ventana de Acceso Restringido (RAW).  
Fuente: (IEEE 802.11 Working Group, 2016).

Mediante la Ec. 8 es posible calcular la duración de una ranura de tiempo RAW. Para lo cual, será necesario conocer el parámetro **Slot Duration Count C** (ver Figura 16), mismo que depende del subcampo **Slot Definition Format Indication** (ver Figura 16). Si este último tiene asignado un bit 0, el subcampo **Slot Duration Count C** tendrá 8 bits ( $y=8$ ), por otro lado, si tiene asignado un bit 1, el subcampo **Slot Duration Count C** tendrá 11 bits ( $y=11$ ).

$$D_{slot} = 500\mu s + C_{slot} * 120\mu s \quad (\text{Ec. 8})$$

Donde,  $C_{slot}$  es el valor del parámetro **Slot Duration Count C**.

Así mismo, una vez obtenida la duración de una ranura de tiempo RAW se puede calcular la duración de la Ventana de Acceso Restringido, solamente se necesita del subcampo **Number of Slots** (ver Figura 16), este es un entero calculado como  $(14 - y)$  e indica la cantidad de RAW slots en una ventana. Finalmente, se deberá reemplazar los valores de la Ec. 8 y el **Number of Slots** en la Ec. 9.

$$D_{RAW} = D_{slot} * N_{RAW} \quad (\text{Ec. 9})$$

Un elemento RPS (RAW Parameter Set) contiene uno o más campos RAW Assignments, el formato de un elemento RPS se muestra en la Figura 13, se compone solamente por un identificador [Element ID], un campo de longitud [Length] y uno o

varios campos de asignación de RAW [RAW Assignments], de modo que la longitud de este elemento es variable.

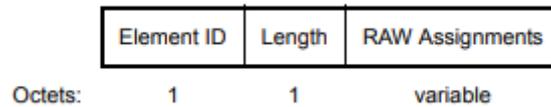


Figura 13. Formato de un elemento RPS.  
Fuente: (IEEE 802.11 Working Group, 2016).

De la trama de un elemento RPS (ver Figura 13) se puede expandir el subcampo **RAW Assignments**, obteniendo el formato que se puede observar en la Figura 14, compuesta por un subcampo de control [RAW Control], un subcampo de definición de ranura [RAW Slot Definition] (ampliado en la Figura 16), la hora de inicio de la RAW [RAW Start Time], el grupo RAW asignado [RAW Group], el canal RAW indicado [RAW Group] y un subcampo de parámetros de operaciones periódicas [Periodic Operation Parameters].

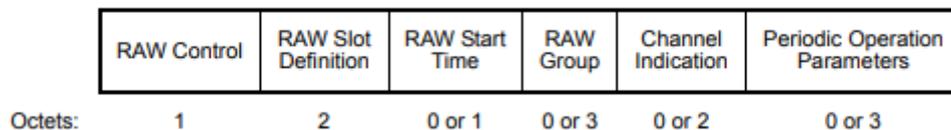


Figura 14. Formato del subcampo RAW Assignments.  
Fuente: (IEEE 802.11 Working Group, 2016).

A su vez, dentro del subcampo **RAW Control** (ver Figura 15) se encuentran aún más subcampos como el de tipo [Raw Type], de opciones de tipo de RAW [RAW Type Options], el indicador la hora de inicio [Start Time Indication], el indicador de grupo de RAW [RAW Group Indication], indicador de presencia de canal [Channel Indication Presenec] y de indicaciones periódicas de RAW [Periodic RAW Indication].

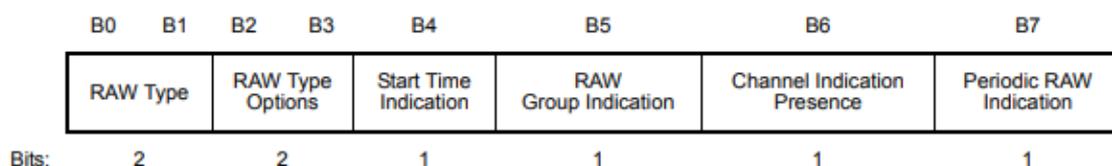


Figura 15. Formato del subcampo de control RAW.  
Fuente: (IEEE 802.11 Working Group, 2016).

Finalmente, al expandir el subcampo **RAW Slot Definition** (ver Figura 16), se encuentran los parámetros usados por las Ec. 8 y Ec. 9 para calcular la duración de la ranura y la ventana respectivamente. Dichos parámetros son: el indicador de formato para las ranuras de tiempo [Slot Definition Format Indication], el límite de ranura cruzada [Cross Slot Boundary], luego está el contador de duración de la ranura [Slot Duration Count] y el número de ranuras [Number of Slots] que dependerá directamente del subcampo anterior. Si el parámetro **Slot Definition Format Indication** tiene asignado un bit 1 indica que la STA tiene permitido cruzar el límite establecido en su ranura de tiempo asignada en caso de que aun necesite enviar o recibir paquetes. Si tiene un bit 0, no tiene permitido transmitir más allá desde su RAW slot establecido.

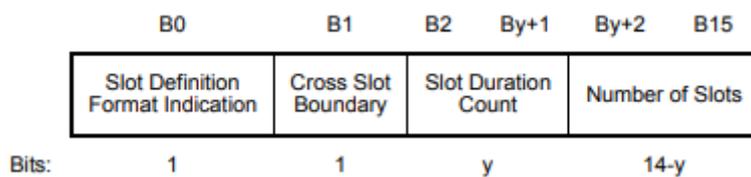


Figura 16. Formato del subcampo RAW Slot Definition.  
Fuente: (IEEE 802.11 Working Group, 2016).

#### 2.1.1.2.6 CABECERA REDUCIDA

Si bien las velocidades de transmisión que se pueden alcanzar con IEEE 802.11ah son bastante altas (en comparación a Legacy), el throughput puede llegar a ser bajo debido a la transmisión de paquetes cortos, propia de redes de sensores. Este overhead (sobrecarga de cabecera) afecta además a la eficiencia energética del dispositivo, es un desperdicio de energía enviar/recibir mayor cantidad de bits de cabecera que de datos. Con el fin de optimizar el protocolo y reducir la sobrecarga de cabecera se desarrollaron varios mecanismos novedosos dirigidos a la capa MAC.

El primer mecanismo, es contar con formatos de trama más compactos, así como una cabecera MAC reducida en comparación con IEEE 802.11 Legacy. Una cabecera MAC corta no cuenta con el campo de duración [Dur. ID], ni con la cuarta dirección

opcional [A4], tampoco mantiene el campo de QoS (Quality of Service) o el HT (High Troughput); tiene la opción de usar solamente dos direcciones en lugar de las 4 direcciones posibles de Legacy. Con estas consideraciones se llega a tener un 12% (14 Bytes de cabecera) de sobrecarga de cabecera, en lugar de 26% (36 Bytes de cabecera) de sobrecarga de cabecera que presenta IEEE 802.11 Legacy (ver figura 17). El estándar IEEE 802.11ah cambia los 2 bits que corresponden a la versión del protocolo dentro del campo **Frame Control (FC)** de 00 a 01, siendo esta la primera vez que un estándar realiza este cambio.

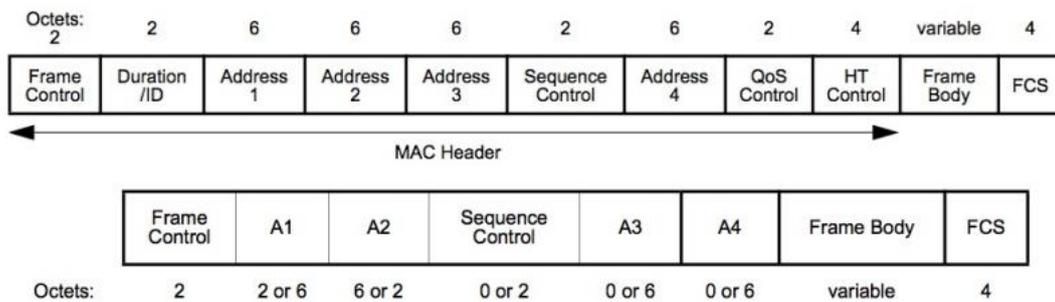


Figura 17. Comparación de cabeceras MAC 802.11ah y 802.11 legacy.  
Fuente: (Baños González, 2020).

Para reducir a un más la cabecera, IEEE 802.11ah reemplaza las direcciones de 6 bytes los dispositivos que no son APs, por direcciones de 2 bytes conocidos como SID (Short Identifier). Cada SID se encuentra compuesto por un AID (Association ID) de 13 bits y banderas. Estas banderas tienen como objetivo indicar la presencia de una tercera y cuarta dirección en casos especiales. En la Figura 18 se puede identificar cada uno de los campos que componen a un SID, como el AID [Association ID (AID)], el bit de la tercera dirección [A3 Present], el bit de la cuarta dirección [A4 Present] y un bit para indicar si el PDU de la capa MAC es dinámico [A-MSDU].

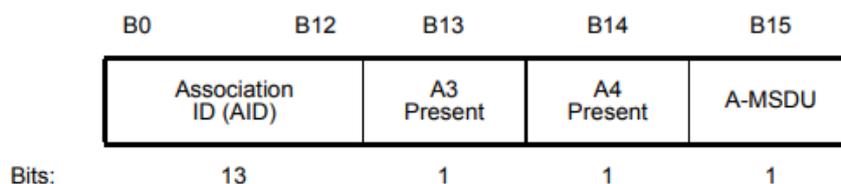


Figura 18. Formato de un SID.  
Fuente: (IEEE 802.11 Working Group, 2016).

El segundo mecanismo usado para reducir el overhead o sobrecarga de cabecera son los NULL Data Packets o Paquetes de Datos Nulos (NDP), la idea de estos paquetes es concentrar la información relevante de las tramas de control en la cabecera PHY, de este modo se evita la cabecera MAC y el payload, un ejemplo de uso de este mecanismo es el NDP ACK o NDP CTS (Garcia-Villegas & López-Aguilera, 2018).

## 2.2 ESTÁNDAR LORA

El protocolo LoRaWAN (Long Range Wide Area Network) se encuentra optimizado para dispositivos alimentados por una batería, estos pueden ser móviles o montados en una ubicación específica. Las redes LoRaWAN se ordenan en una topología en estrella-de-estrellas o star-of-stars, de modo que los Gateways o estaciones base actúen de Relay para pasar mensajes entre los dispositivos finales o motes y un Servidor de Red central. La función del Servidor de Red es la de enrutar los paquetes desde cada dispositivo de la red hacia su Servidor de Aplicaciones asociado. En cuanto a seguridad, maneja criptografía simétrica con claves de sesión derivadas de las claves raíz de cada dispositivo, esta información es almacenada y asegurada por un Servidor de Unión<sup>5</sup>.

La comunicación entre los dispositivos finales y los Gateways se realiza a través de diferentes canales de frecuencia y velocidades de transmisión, la elección de este último es en términos del rango de comunicación y la duración del mensaje. La velocidad de transmisión de LoRa puede ser desde 0.3 kbps hasta 50 kbps, con el fin de maximizar

---

<sup>5</sup> El Servidor de Unión o Join Server, maneja el flujo de unión de LoRaWAN, incluida la autenticación del Servidor de Aplicaciones y del Servidor de Red, así como la generación de claves de sesión.

la vida útil de la batería y de los dispositivos finales se aplica un esquema de velocidad de datos adaptativo o ADR (Adaptative Data Rate).

### 2.2.1 CLASES

El estándar LoRaWAN define un stack de protocolos acorde a la Figura 19, LoRa es la capa física utilizada para crear enlaces de larga distancia (de ahí vienen su nombre, Long Range), por su parte, la capa MAC tiene tres tipos de MAC options denominadas Clases. La Clase A está orientada a sensores que se alimentan por batería, la Clase B se dirige a actuadores que funcionan con batería y la Clase C para actuadores alimentados por una fuente principal.

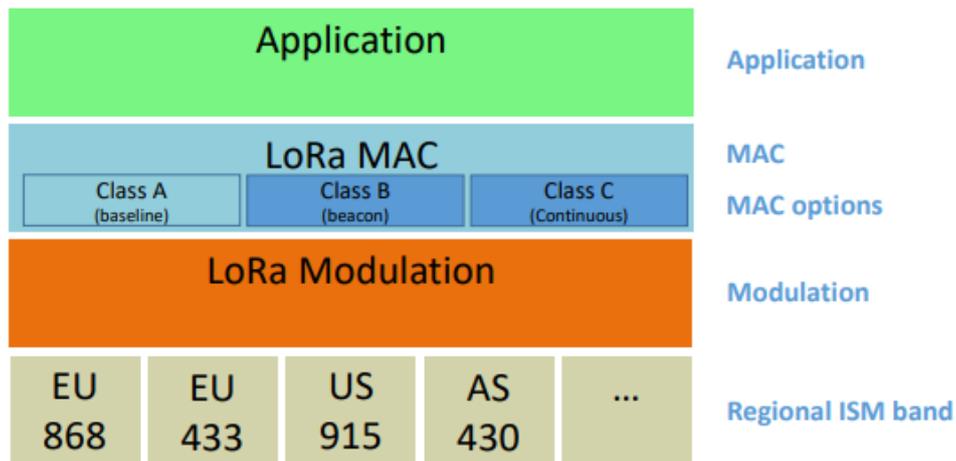


Figura 19. Clases de LoRaWAN.  
Fuente: (Committee LoRa Alliance Technical, 2017).

Los end-devices pertenecientes a la clase A permiten una comunicación bidireccional, donde cada transmisión para un enlace de subida está seguido por una ventana de recepción para dos enlaces de bajada más cortos (Casals Ibáñez, Mir Masnou, Vidal Ferré, & Gomez, 2017), tal como se observa en la Figura 20. Las ranuras de comunicación están organizadas de acuerdo con las necesidades del propio dispositivo final con una ligera variante de una base de tiempo aleatorio, es decir, maneja un protocolo de tipo ALOHA. Debido a este funcionamiento, la clase A es la que menos energía consume y es el modo por defecto de todos los dispositivos finales.



Figura 20. Ventanas de recepción para una clase A.  
Fuente: (Bouguera , Diouris, Chaillout, Jaouadi, & Andrieux, 2018).

Otros dispositivos más robustos pertenecerán a una clase B, estos contarán con ranuras de recepción programadas, aparte de la ventana de recepción aleatoria de la clase A se agrega una ventana de recepción extra programada en un tiempo determinado, para abrir dicha ventana el dispositivo recibe un Beacon de sincronización de tiempo desde el Gateway. En la Figura 21 se identifica el comportamiento de una clase B, donde, entre cada periodo de beacon (para sincronización) el Gateway inicia la transmisión enviado mensajes PNG (ping slots), de este modo el dispositivo final recibirá ventanas de recepción adicionales en tiempos programados.

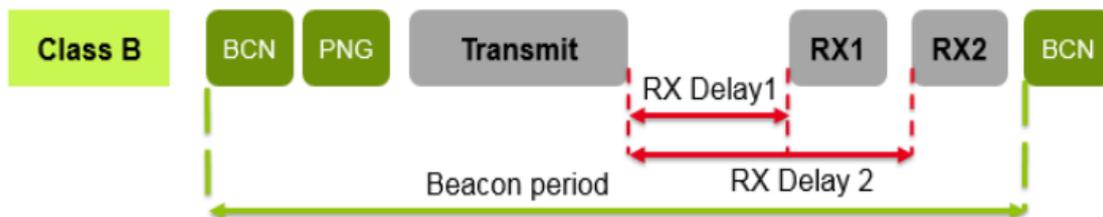


Figura 21. Ventana de recepción programada para una clase B.  
Fuente: (Bouguera , Diouris, Chaillout, Jaouadi, & Andrieux, 2018).

Un dispositivo con el máximo de ranuras de recepción será definido como uno de clase C, ya que prácticamente cuenta con una ventana de recepción continua (ver Figura 22), esta se cierra únicamente cuando el dispositivo va a transmitir, consecuentemente, estos dispositivos finales son aquellos que más consumo energético tienen, a cambio de una latencia mínima en la comunicación entre servidor y dispositivo final.

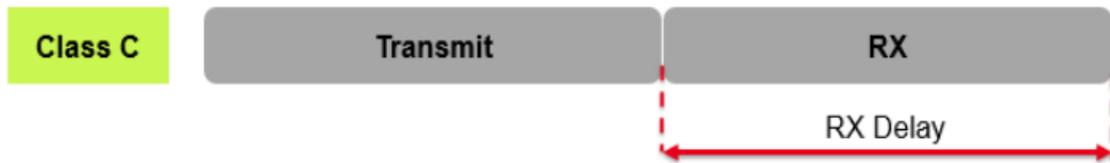


Figura 22. Ventana de recepción continua para una clase C.  
Fuente: (Bouguera , Diouris, Chaillout, Jaouadi, & Andrieux, 2018).

Como regla de oro, el estándar define que todo dispositivo LoRaWAN debe tener compatibilidad al menos con la clase A, aquellos que implementen más clases son denominados “dispositivos finales de alta clase”. Las clases B y C no son más que características adicionales de la capa MAC de clase A (Committee LoRa Alliance Technical, 2017).

### 2.2.2 ARQUITECTURA LORA

Si bien ya fueron mencionados algunos de los elementos que forman parte de una red LoRa, es necesario tener una definición precisa de cada uno de ellos y la función que cumplen, estos elementos son:

- **End-devices:** también conocidos como End-nodes, dispositivos finales o motes, su función es conectar los objetos (things) a la red LoRa, los datos recogidos son transmitidos al siguiente elemento, denominado Gateway, a través de una comunicación inalámbrica single-hop.
- **Gateway:** denominados también pasarelas o estaciones base, funcionan como un puente transparente, retransmitiendo de manera bidireccional las transmisiones de varios de end-devices y servidores de red.
- **Network Server:** una red LoRa puede contar con varios Network Servers (Servidores de Red), mismos que se conectan a varios Gateways a través de una conexión TCP/IP segura (alámbrica o inalámbrica). Su función es la de gestionar y configurar la red, así como recibir y procesar (eliminar duplicidad) la información proveniente de los end-devices.

- **Application Server:** el servidor de aplicaciones maneja la capa de aplicación LoRaWAN, incluyendo el descifrado y decodificación de datos de enlace ascendente, la cola de enlace descendente, así como también la codificación y cifrado de datos de enlace descendente (LoRa Alliance, 2015).

De acuerdo con la especificación LoRaWAN 1.1, se trata a los Servidores de Red, Aplicación y Unión como si siempre compartieran la misma ubicación (cubicados). El hecho de que estas funcionalidades se encuentren repartidas en múltiples nodos queda fuera de la especificación.

Los Gateways se encuentran conectados al Servidor de Red a través de una conexión IP estándar segura; mientras que por otro lado, se conectan al Gateway los dispositivos finales mediante un single-hop de LoRa (comunicación inalámbrica con modulación por desplazamiento de frecuencia FSK). En la Figura 23 se muestra la ubicación de cada uno de los elementos descritos anteriormente y que conforman la arquitectura de red LoRaWAN. A pesar de que la comunicación de la red es bidireccional, el tráfico predominante en la misma generalmente es tipo uplink (enlace ascendente), es decir entre un dispositivo final y un Servidor de Red.

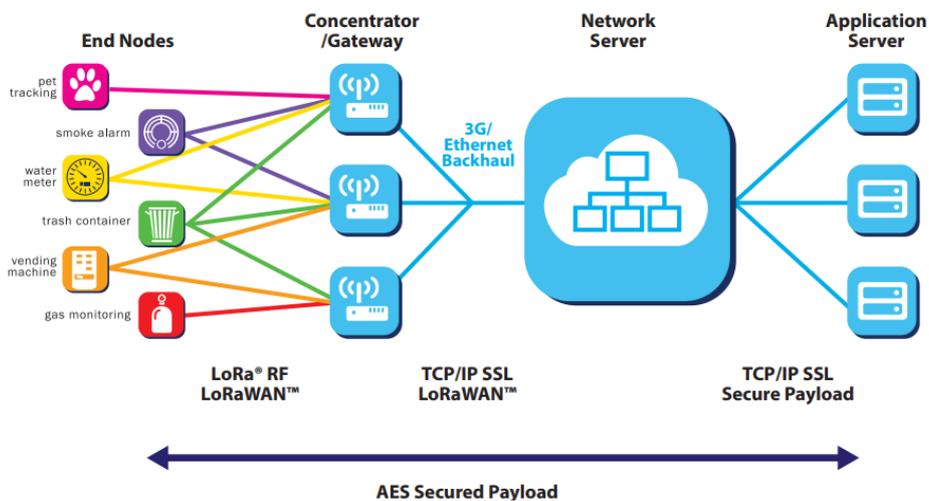


Figura 23. Arquitectura de una red LoRaWAN.  
Fuente: (LoRa Alliance, 2015).

### 2.2.2.1 CAPA FÍSICA LORA

La especificación LoRaWAN, en cuanto se refiere a bandas de frecuencia de operación, varía dependiendo de la región debido a lo dictado por las organizaciones regulatorias correspondientes, regiones como Europa o Norteamérica son aquellas que tienen más “claro” el panorama de utilización de LoRa. Es por esta razón que en la Tabla 5 contiene más información relacionada a las regiones europea y norteamericana.

Tabla 5. Bandas de operación de LoRaWAN.

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 +8	In definition by Technical Committee			
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kpbs				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

Fuente: (LoRa Alliance, 2015).

Para Europa, la ETSI (European Telecommunications Standards Institute) define 10 canales utilizables, de los cuales ocho son multi data rate con velocidades de transmisión desde 250 bps hasta 5.5 Kbps, un canal de alta velocidad de transmisión de hasta 11 kbps y un canal FSK de 50 Kbps. La potencia máxima permitida por la ETSI es de +14 dBm, a excepción de la banda G3 que permite hasta +27 dBm. La ETSI no toma en cuenta limitaciones de tiempo máximo de transmisión o tiempo de permanencia del canal (dwell) pero si limita el tiempo de trabajo (duty cycle).

En el caso de Norteamérica, el organismo encargado de estas regulaciones es la FCC (Federal Communications Commission). La frecuencia de trabajo difiere de la

utilizada en Europa, debido a que la banda ISM (Industrial, Scientific and Medical band) para esta región está comprendida entre 902 MHz y 928 MHz, dentro de la cual se definen 64 canales de uplink de 125 kHz de ancho de banda entre 902.3 MHz y 914.9 kHz con incrementos de 200 kHz, adicionalmente, 8 canales de uplink de 500 kHz entre 903 MHz y 914.9 MHz con incrementos de 1.6 MHz, finalmente, 8 canales de downlink de 500 kHz entre 923.3 MHz y 927.5 MHz (ver Figura 24). La región norteamericana permite un uso más alto de potencia de transmisión, llegando hasta +27 dBm en todos sus canales.

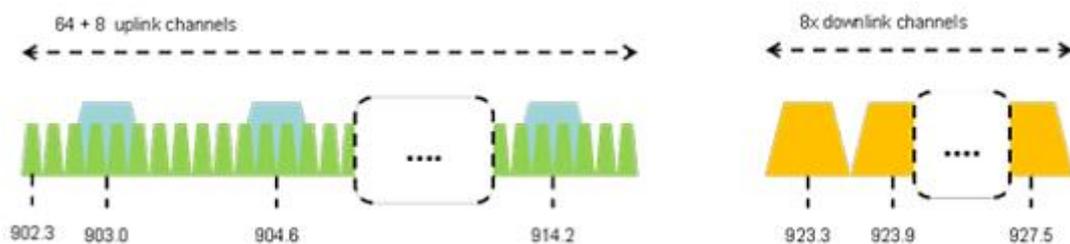


Figura 24. Disposición de canales en Norteamérica (FCC).  
Fuente: (LoRa Alliance, 2015).

Una diferencia clave que se ha considerado a la hora de la realización del presente trabajo es la limitante del ciclo de trabajo. Mientras que en Europa se limita al 1%, en Norte América no existe un límite para el ciclo de trabajo, únicamente se define una restricción de 400 milisegundos de dwell time (tiempo de permanencia) por canal, este hecho podría contribuir a una interferencia co-protocolo en redes densas.

#### 2.2.2.1.1 FORMATOS DE MENSAJES DE TRAMA FÍSICA

Los mensajes de enlace de subida o uplink son enviados por los dispositivos finales hacia el Servidor de Red o Network Server a través de uno o varios Gateways. Estos mensajes usan el modo de paquete de radio explícito de LoRa, en dicho modo se incluye un PHDR (una cabecera PHY de LoRa) y una cabecera de PHDR\_CRC (Código de Redundancia Cíclica), para proteger la integridad de la carga útil se cuenta con un campo CRC al final. Tanto el campo PHDR como el PHDR\_CRC son insertados por el

transceiver (Rx y Tx) del dispositivo final. La estructura PHY del enlace de subida y los campos mencionados se muestra en la Figura 25.

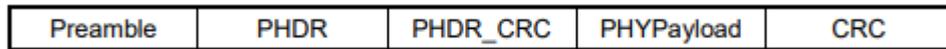


Figura 25. Estructura del uplink PHY.  
Fuente: (Committee LoRa Alliance Technical, 2017).

Por otro lado, los mensajes de enlace de bajada o downlink son enviados por el Network Server hacia un único dispositivo final a través de un solo Gateway (no se considera el caso de multicast). Los mensajes de downlink tienen una estructura similar a los de uplink, en la Figura 26 se muestra cómo se repiten los campos de PHDR y PHDR\_CRC. Como diferencia más notable, en el downlink no se realiza la comprobación de integridad del payload, es decir no cuenta con el campo CRC, esto con el objetivo de mantener el mensaje lo más corto posible, con un impacto mínimo en cualquier limitante del ciclo de trabajo de la banda ISM usada.



Figura 26. Estructura del downlink PHY.  
Fuente: (Committee LoRa Alliance Technical, 2017).

#### 2.2.2.1.2 VENTANA DE RECEPCIÓN

Seguido de cada transmisión de uplink, el dispositivo final deberá abrir dos ventanas cortas de recepción. Los tiempos de inicio de cada ventana de recepción son definidos usando el final de la transmisión como referencia. En la Figura 27 se aprecia la ranura de tiempo de transmisión, al final de esta se encuentran dos ventanas más cortas de recepción (RX1 y RX2).

La primera ventana de recepción RX1 utiliza una frecuencia y velocidad de transmisión que son funciones de las usadas en el uplink respectivamente. Antes de que la ventana esté disponible pasa un tiempo RECEIVE\_DELAY1 (20 microsegundos

aproximadamente) luego de finalizada la modulación de uplink. Por defecto la velocidad de transmisión de la primera ventana de recepción (RX1) es idéntica a la de último uplink, sin embargo, esto puede variar dependiendo de la región.

La segunda ventana de recepción RX2, usa una frecuencia y velocidad de transmisión fijas (configurables), la ventana se abre pasado un tiempo RECEIVE\_DELAY2 luego de finalizada la modulación uplink. Tanto la frecuencia como la velocidad de transmisión son configurables a través de comandos MAC.

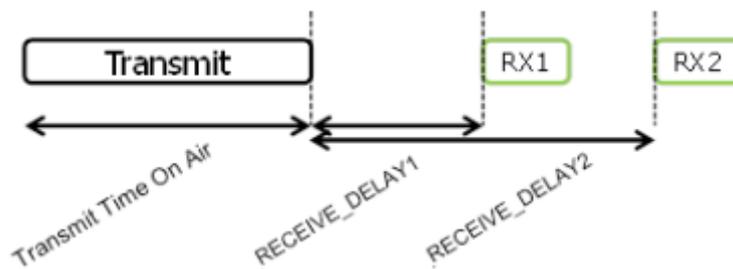


Figura 27. Ventanas de recepción.  
Fuente: (Committee LoRa Alliance Technical, 2017).

La duración de una ventana de recepción deberá ser suficiente para que un transceiver (dispositivo final) detecte un preámbulo de downlink [Preamble] (ver Figura 26). Si el preámbulo es detectado durante una ventana de recepción, el receptor de radio permanece activo hasta que la trama de downlink sea demodulada, en caso de que la dirección y el MIC (Código de Integridad de Mensaje) pertenezcan al dispositivo final no se abrirá la segunda ventana de recepción.

### 2.2.2.1.3 CHIRP SPREAD SPECTRUM

La técnica de modulación manejada por LoRa, se la conoce como CSS (Chirp Spread Spectrum) o también pulso chirp; la secuencia de chirp tiene una frecuencia más alta que la señal de información, produciendo un “esparcimiento” o “spreading” del ancho de banda de la señal que está siendo transmitida. De acuerdo con (Semtech

Corporation, 2015) esta técnica se caracteriza por ofrecer robustez a interferencias y la capacidad de comunicarse a largas distancias.

En la modulación de LoRa, el espectro ensanchando se logra generando una señal de chirp que varíe continuamente en frecuencia, el uso de esta técnica permite reducir la complejidad del diseño del receptor. La señal de datos deseada se procesa a una velocidad de datos más alta y se modula en la señal de chirp. La expresión matemática de la Ec.10 muestra como calcular la tasa de bits de la señal modulada en función de un término conocido como SF (Spreading Factor).

$$R_b = SF * \frac{1}{\left(\frac{2^{SF}}{BW}\right)} [bps] \quad (\text{Ec. 10})$$

El Spreading Factor se define como la cantidad de chips utilizados para representar un símbolo, entonces, este factor acompañado del ancho de banda permite obtener el tiempo de símbolo respectivo. Esta afirmación se encuentra representada en la expresión de la Ec. 11. Donde, SF puede tomar valores del 7 al 12 (ver Tabla 7) y BW es el ancho de banda Herzios.

$$T_s = \frac{2^{SF}}{BW} [s] \quad (\text{Ec. 11})$$

De la Ec. 11 es posible obtener el symbol rate [ $R_S$ ] (cantidad de símbolos por unidad de tiempo), simplemente con calcular el inverso del tiempo de símbolo,  $R_S = \frac{1}{T_s} = \frac{BW}{2^{SF}}$  [símbolos/s]. Este nuevo parámetro se puede reemplazar en la Ec. 12 de la definición del chip rate [ $R_C$ ], para demostrar una definición propia de la modulación de LoRa, la cual menciona que “un chip es enviado por segundo por Hz de banda ancha”.

$$R_C = R_S * 2^{SF} [chips/s] \quad (\text{Ec. 12})$$

$$R_C = \frac{BW}{2^{SF}} * 2^{SF} [chips/s]$$

$$R_C = BW [chips/s]$$

Una ventaja de este método es que las compensaciones de tiempo y frecuencia entre el transmisor y el receptor son equivalentes, lo que reduce en gran medida la complejidad del diseño del receptor. El ancho de banda de frecuencia de este chirp es equivalente al ancho de banda espectral de la señal. La señal de datos que lleva los datos desde un dispositivo final a una puerta de enlace se procesa a una velocidad de datos más alta y se modula en la señal portadora de chirp. La modulación LoRa también incluye un esquema de corrección de error variable que mejora la robustez de la señal transmitida. Por cada cuatro bits de información enviados, se envía un quinto bit de información de paridad (Semtech Corporation, 2015).

#### **2.2.2.1.4 FACTOR DE PROPAGACIÓN**

Como se señaló anteriormente, la ganancia de procesamiento LoRa se introduce en el canal de RF multiplicando la señal de datos con un spreading code. Al aumentar la tasa de chips, también se aumentan los componentes de frecuencia del espectro total de la señal. En otras palabras, la energía de la señal total ahora se distribuye en un rango más amplio de frecuencias, lo que permite al receptor discernir una señal con una SNT (Signal-to-Noise Rate) más baja (es decir, peor). En términos de LoRa, la cantidad de spreading code aplicado a la señal de datos original se denomina SF. La modulación LoRa tiene un total de seis factores de dispersión (SF7 a SF12). Cuanto mayor sea el SF utilizado, más lejos podrá viajar la señal y todavía ser recibida sin errores por el receptor RF.

Un ejemplo para visualizar la diferencia entre diferentes SFs se muestra en la Tabla 6. Donde, los valores presentados pertenecen a un tráfico UL (uplink), es por esta razón que solamente constan cuatro factores [SF7 a SF10]; mientras mayor es el SF la tasa de bits disminuye considerablemente de 547 bps (SF7) hasta 980 bps (SF10), como compensación el rango de cobertura se puede hasta cuadruplicar (de 2 a 8 Km). Finalmente, si bien el parámetro Time on Air (tiempo que se ocupa el canal, dwell time)

aumenta junto con el SF, se lo puede relacionar simplemente con el incremento del rango, es decir, mientras más lejos se encuentre el destino más tiempo le tomará a la señal llegar a él.

Tabla 6. Tabla de factores de propagación LoRa.

Spreading Factor (For UL at 125 KHz)	Bit Rate	Range (Depends on Terrain)	Time on Air for an 11-byte payload
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

Fuente: (Semtech, 2015).

Es importante destacar que el SF de la modulación LoRa son intrínsecamente ortogonales. Esto significa que las señales moduladas con diferentes factores de dispersión y transmitidas en el mismo canal de frecuencia al mismo tiempo no interfieren entre sí. En cambio, las señales en diferentes SF simplemente parecen ser ruido entre sí.

Las señales LoRa son robustas y muy resistentes a los mecanismos de interferencia dentro y fuera de banda. La modulación LoRa también ofrece inmunidad a las trayectorias múltiples y al desvanecimiento, lo que la hace ideal para su uso en entornos urbanos y suburbanos, donde dominan ambos mecanismos. Además, los cambios producidos por el efecto Doppler<sup>6</sup> provocan un pequeño cambio de frecuencia en el eje de tiempo de la señal de banda base. Esta tolerancia de desplazamiento de frecuencia mitiga el requisito de fuentes de reloj de referencia de tolerancia estricta y, por lo tanto, hace que LoRa sea ideal para comunicaciones de datos desde dispositivos móviles.

---

<sup>6</sup> El efecto Doppler produce cambios en la frecuencia, aumentando si el objeto móvil se acerca a la fuente y reduciendo si se aleja, la nueva frecuencia se calcula con:  $f' = f_0 \left( \frac{1}{1 + \frac{v}{c_0}} \right)$ , donde  $f_0$  es la frecuencia de la señal,  $v$  es la velocidad a la que se mueve el objeto y  $c_0$  es la velocidad de la onda en un medio estacionario.

En la Tabla 6, se encuentran organizados los parámetros del esquema de modulación LoRa, el término que más destaca es el SF, que como ya se lo ha mencionado antes, está relacionado directamente con la tasa de bits por segundo.

Tabla 7. Esquema de modulación de LoRa.

Data Rate (DR)	Spreading Factor (SF)	Channel Frequency	Uplink or Downlink	Bitrate (Bits/Sec)	Maximum User Payload Size (Bytes)
0	SF10	125 kHz	Uplink	980	11
1	SF9	125 kHz	Uplink	1,760	53
2	SF8	125 kHz	Uplink	3,125	125
3	SF7	125 kHz	Uplink	5,470	242
4	SF8	500 kHz	Uplink	12,500	242
5 – 7					
8	SF12	500 kHz	Downlink	980	53
9	SF11	500 kHz	Downlink	1,760	129
10	SF10	500 kHz	Downlink	3,125	242
11	SF9	500 kHz	Downlink	5,470	242
12	SF8	500 kHz	Downlink	12,500	242
13	SF8	500 kHz	Downlink	21,900	242

Fuente: (Semtech, 2015).

De la tabla presentada se pueden destacar algunas características propias de la modulación LoRa relacionadas con la capa PHY, el ancho de banda, su ortogonalidad y el SF:

- La capa física LoRa está diseñada para aplicaciones de bajo rendimiento, baja velocidad de datos y alto Link Budget (es decir de "largo alcance").
- Para un ancho de banda de canal fijo, cuanto mayor es el SF, mayor es la ganancia de procesamiento, lo que da como resultado un aumento de la sensibilidad y, por lo tanto, un aumento del Link Budget.
- La ortogonalidad entre factores de expansión permite la transmisión de múltiples señales LoRa que están en la misma frecuencia de canal y en el mismo intervalo de tiempo.

- Para un SF fijo, un ancho de banda más estrecho aumentará la sensibilidad a medida que se reduce el bit rate.
- LoRaWAN en Norteamérica utiliza canales de uplink de 125 kHz y canales de enlace uplink/downlink de 500 kHz.

### 2.2.2.2 CAPA MAC LORA

Todos los mensajes de uplink o downlink contienen un Payload PHY o carga útil (ver Figuras 25 y 26). La estructura de este Payload varia un poco (dependiendo del tipo), pero en general contiene una cabecera MAC [MHDR] con una longitud de un octeto, seguido de un Payload MAC [MACPayload] (su longitud máxima dependerá de la región) y un Código de Integridad de Mensajes [MIC] con una longitud de 4 octetos. En la Figura 28, se muestran los tipos de estructuras que puede tener el Payload PHY. Aquellos de solicitud de unión [Join-Request] o solicitud de reunión [Rejoin-Request] reemplazan al Payload MAC [MACPayload] por uno de su tipo específico, mientras que para la trama de aceptar la unión [Join-Accept] no se cuenta con un campo MIC porque este ya se encuentra encriptado con el Payload y no de manera separada.

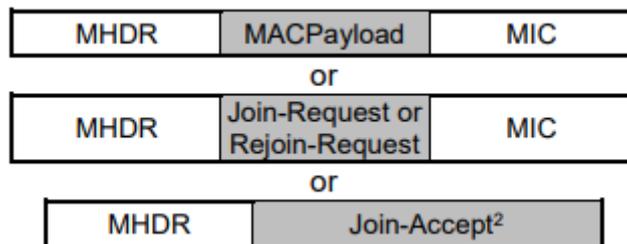


Figura 28. Estructura de un payload PHY.  
Fuente: (Committee LoRa Alliance Technical, 2017).

La estructura del payload MAC mostrada en la Figura 29 es producto de expandir el campo [MACPayload] de un Payload PHY (ver Figura 28). Contiene una Cabecera de Trama [FHDR], seguido de un campo de Puerto opcional [FPort] y un campo de Payload opcional [FRMPayload].

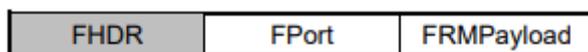


Figura 29. Estructura del payload MAC.  
Fuente: (Committee LoRa Alliance Technical, 2017).

### 2.2.2.2.1 TIPOS DE MENSAJES MAC

LoRaWAN distingue entre 8 tipos diferentes de mensajes MAC, es posible identificarlos por el subcampo [MType] dentro del campo [MHDR] del payload PHY (ver Figura 30).

MType	Description
000	Join-request
001	Join-accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	Rejoin-request
111	Proprietary

Figura 30. Tipos de mensajes MAC.  
Fuente: (Committee LoRa Alliance Technical, 2017).

Los mensajes Join-request y Join-accept son utilizados para el proceso de activación over-the-air y para roaming. Los mensajes de datos (DATA) son utilizados para transferir comandos MAC y datos de aplicación combinados en un solo mensajes, los de tipo “Confirmed” requieren de un acuse de recibo, mientras que los de tipo “Unconfirmed” no requieren de un ACK.

### 2.2.2.2.2 PAYLOAD MAC

Al expandir el subcampo de la Cabecera de Trama [FHDR] (ver Figura 29), obtenemos, una dirección corta de dispositivo [DevAddr] con longitud de cuatro octetos, un control de trama [FCtrl] de un octeto, un contador de trama [FCnt] de dos octetos y un campo de opciones de trama [FOpts] de quince octetos para transportar comandos MAC. En la Figura 31 se muestra el formato de cabecera FHDR con los tamaños de cada campo.

<b>Size (bytes)</b>	4	1	2	0..15
<b>FHDR</b>	DevAddr	FCtrl	FCnt	FOpts

Figura 31. Formato de cabecera de trama.  
Fuente: (Committee LoRa Alliance Technical, 2017).

El campo para el control de trama [FCtrl] varía ligeramente si se trata de un mensaje de downlink o uplink, lo que se mantiene en ambos tipos de mensajes es el subcampo ADR (Adaptive Data Rate), que permite adaptar y optimizar la velocidad de transmisión y la potencia de los dispositivos finales estáticos (utilizar ADR no es posible cuando la atenuación del canal de radio cambia de manera rápida y constante, aplicaciones móviles). En caso de que el Network Server no sea capaz de controlar la velocidad de transmisión, esta tarea recae en la capa de aplicación del dispositivo (siempre se tratará de minimizar el tiempo aire agregado debido a las condiciones de la red).

Si el bit de ADR de la Figura 32 y/o Figura 33 se encuentra establecido en 1, la red controlará la velocidad de transmisión y la potencia de transmisión del dispositivo a través de los comandos MAC; caso contrario, la red no hará ni siquiera el intento de controlar dichos parámetros.

<b>Bit#</b>	7	6	5	4	[3..0]
<b>FCtrl bits</b>	ADR	RFU	ACK	FPending	FOptsLen

Figura 32. Contenido del campo FCtrl para downlink.  
Fuente: (Committee LoRa Alliance Technical, 2017).

<b>Bit#</b>	7	6	5	4	[3..0]
<b>FCtrl bits</b>	ADR	ADRACKReq	ACK	ClassB	FOptsLen

Figura 33. Contenido del campo FCtrl para uplink.  
Fuente: (Committee LoRa Alliance Technical, 2017).

Incluso los dispositivos móviles tienen periodos estacionarios, durante estos periodos de tiempo los dispositivos finales deberían solicitar a la red la optimización de la velocidad de transmisión a través del bit de ADR de uplink.

Cuando se recibe un mensaje de datos de tipo “Confirmed” será necesario responder con una trama de datos que tenga el bit de ACK. Si el que envía el mensaje de datos es un dispositivo final, la red enviará un ACK utilizando una de las ventanas de recepción abiertas luego de la operación (ver Figura 27). Si quien envía los datos es el Gateway entonces, el dispositivo final transmitirá un ACK según su propia discreción. Para mantener este proceso lo más simple posible se tratará de transmitir un mensaje de ACK posiblemente vacío justo luego de la recepción del mensaje de datos.

Únicamente en los mensajes de downlink se encuentra el campo [FPending], su función es indicar que la red tiene más información pendiente a ser enviada, de modo que busca que el dispositivo final abra una ventana de recepción lo más pronto posible (enviando un mensaje de uplink).

Cada dispositivo cuenta con tres contadores [FCnt], estos llevan la información del número de tramas de datos enviadas como uplink hacia el Network Server [FCntUp] y enviados como downlink desde el Network Server hacia el dispositivo [FCntDown].

El subcampo [FOptsLen] u opciones de longitud de trama hace referencia a la longitud actual del campo de opciones de trama (incluyendo la trama). Si tiene un bit 0, entonces el campo [FOpts] se encuentra ausente, mientras que, si [FOptsLen] es diferente de 0, es indicativo que existen comandos MAC en el campo [FOpts] (encriptados usando el esquema de IEEE 802.15.4/2006 AES).

El subcampo [ClassB] en el mensaje de uplink indica al Network Server que el dispositivo ha cambiado a Clase B (explicado la sección 2.2.1) y se encuentra listo para recibir pings de downlink programados.

Finalmente, en la estructura del payload MAC (ver figura 29), el Campo de Puerto [FPort] indica que el campo de payload de trama no está vacío. En caso de que el bit de este campo sea 0, indicaría que el [FRMPayload] contiene únicamente comandos MAC.

En caso de que la trama de datos tenga un payload, el [FRMPayload] deberá estar encriptado previo a que el MIC (Message Integrity Code) sea calculado (encriptados usando el esquema de IEEE 802.15.4/2006 AES).

### **2.2.2.2.3 SEGURIDAD**

La seguridad de LoRaWAN está diseñada para ajustarse a los criterios generales de IoT: bajo consumo de energía, baja complejidad de implementación, bajo costo y alta escalabilidad. Dado que los dispositivos se implementan en el campo durante largos períodos de tiempo (años). Para mantenerse en la vanguardia en cuanto a seguridad se refiere, LoRaWAN utiliza algunas de las propiedades como son: autenticación mutua, protección de la integridad y confidencialidad (Gemalto, Actility, & Semtech, 2017).

La autenticación mutua se establece entre un dispositivo final LoRaWAN y la red LoRaWAN como parte del procedimiento de unión a la red (Join Server). Esto asegura que solo los dispositivos genuinos y autorizados se unirán a redes genuinas y auténticas.

Los mensajes de la capa MAC y de la capa Aplicación están autenticados en el origen, contando con protección de integridad (protegida contra reproducción y cifrada). Esta protección, combinada con la autenticación mutua, asegura que el tráfico de la red no se haya alterado, provenga de un dispositivo legítimo, no sea comprensible para los espías (eavesdropping) y que no haya sido capturado/reproducido por los atacantes.

Además, se aplica cifrado de extremo a extremo para los payloads de las aplicaciones intercambiadas entre los end-devices y los Application Servers. Al contar con esta protección en la red de aplicación no es necesario implementar una capa de seguridad adicional, como puede ser una VPN. Haciendo de este enfoque el adecuado para una LPWAN, debido a que las capas de seguridad superiores agregan un considerable consumo de energía, complejidad y costo adicionales.

El esquema comprendido en la Figura 34 permite comprender como LoRaWAN maneja las llaves de sesión. Cuando un dispositivo se une a la red (mediante el Join Server), se generan dos llaves: una de sesión de aplicación denominada AppSKey (privada) y una de sesión de red llamada NwkSKey (compartida con la red).

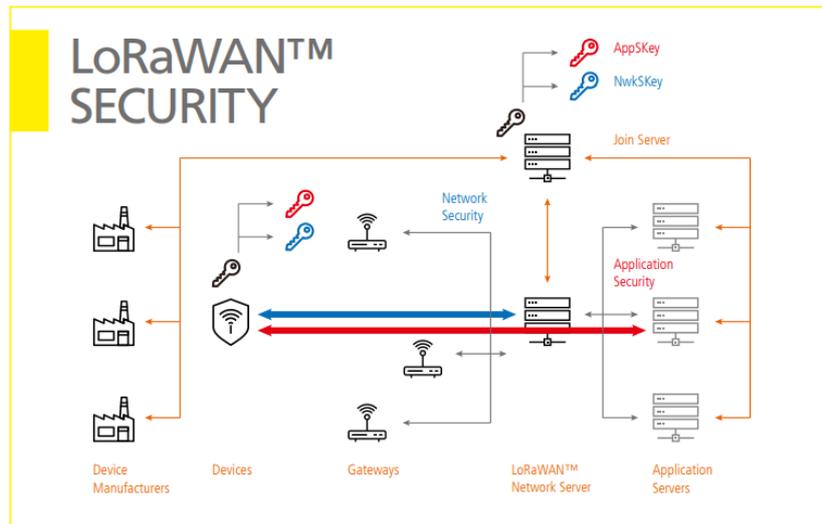


Figura 34. Esquema de seguridad de LoRa.  
Fuente: (Gemalto, Actility, & Semtech, 2017).

La llave de sesión de red [NwkSKey] se utiliza para interactuar entre el end-device y el Network Server. Con esta llave es posible validar la integridad de cada mensaje a través de un mecanismo conocido como MIC (Message Integrity Code). Se puede comparar a este mecanismo con un checksum, con la excepción de que el MIC previene la manipulación mal intencionada del mensaje usando AES-CMAC<sup>7</sup>. Esta validación también permite relacionar la dirección no única [DevAddr] a la dirección global única para dispositivo final [DevEUI] y para el Join Server [AppEUI].

La llave de sesión de aplicación tiene como fin encriptar y desencriptar el payload, este se encontrará completamente encriptado entre el end-device y el Application Server, permitiendo que solamente quien maneja la llave sea capaz de leer el contenido de los mensajes enviados o recibidos.

<sup>7</sup> AES-CMAC viene de Advanced Encryption Standard - Cipher-based Message Authentication Code, provee integridad y autenticidad al mensaje.

La aplicación de las dos llaves mencionadas se indica gráficamente en la Figura 35, donde claramente se puede observar que el payload es encriptado por la AppSKey (llave de sesión de aplicación), mientras que el MIC y la NwkSKey son calculados para evitar la manipulación del paquete.

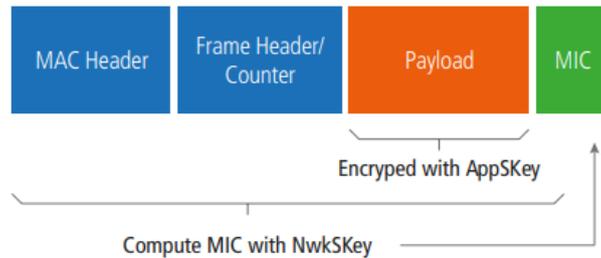


Figura 35. Uso de las llaves AppSKey y NwkSKey.  
Fuente: (Gemalto, Actility, & Semtech, 2017).

#### 2.2.2.2.4 COMANDOS MAC

Estos comandos de administración de red son intercambiados entre el Network Server y la capa MAC de un dispositivo final, los comandos MAC son completamente invisibles para el servidor de aplicación o la aplicación en sí.

Una sola trama de datos puede contener cualquier secuencia de comandos MAC, ya sea en el campo de [FOpts] o enviados en una trama de datos aparte en el campo [FRMPayload] con el campo [FPort] con valor de 0.

La estructura de un comando MAC consta de un Identificador de Comando o CID de un octeto seguido de una secuencia de comandos específicos.

CID	Command	Transmitted by		Short Description
		End-device	Gateway	
0x01	<b>ResetInd</b>	x		Used by an ABP device to indicate a reset to the network and negotiate protocol version
0x01	<b>ResetConf</b>		x	Acknowledges ResetInd command
0x02	<b>LinkCheckReq</b>	x		Used by an end-device to validate its connectivity to a network.
0x02	<b>LinkCheckAns</b>		x	Answer to LinkCheckReq command. Contains the received signal power estimation indicating to the end-device the quality of reception (link margin).
0x03	<b>LinkADRReq</b>		x	Requests the end-device to change data rate, transmit power, repetition rate or channel.
0x03	<b>LinkADRAns</b>	x		Acknowledges the LinkADRReq.
0x04	<b>DutyCycleReq</b>		x	Sets the maximum aggregated transmit duty-cycle of a device
0x04	<b>DutyCycleAns</b>	x		Acknowledges a DutyCycleReq command
0x05	<b>RXParamSetupReq</b>		x	Sets the reception slots parameters
0x05	<b>RXParamSetupAns</b>	x		Acknowledges a RXParamSetupReq command
0x06	<b>DevStatusReq</b>		x	Requests the status of the end-device
0x06	<b>DevStatusAns</b>	x		Returns the status of the end-device, namely its battery level and its demodulation margin
0x07	<b>NewChannelReq</b>		x	Creates or modifies the definition of a radio channel
0x07	<b>NewChannelAns</b>	x		Acknowledges a NewChannelReq command
0x08	<b>RXTimingSetupReq</b>		x	Sets the timing of the of the reception slots
0x08	<b>RXTimingSetupAns</b>	x		Acknowledges RXTimingSetupReq command
0x09	<b>TxParamSetupReq</b>		x	Used by the Network Server to set the maximum allowed dwell time and Max EIRP of end-device, based on local regulations
0x09	<b>TxParamSetupAns</b>	x		Acknowledges TxParamSetupReq command
0x0A	<b>DIChannelReq</b>		x	Modifies the definition of a downlink RX1 radio channel by shifting the downlink frequency from the uplink frequencies (i.e. creating an asymmetric channel)
0x0A	<b>DIChannelAns</b>	x		Acknowledges DIChannelReq command
0x0B	<b>RekeyInd</b>	x		Used by an OTA device to signal a security context update (rekeying)
0x0B	<b>RekeyConf</b>		x	Acknowledges RekeyInd command
0x0C	<b>ADRParamSetupReq</b>		x	Used by the Network Server to set the ADR_ACK_LIMIT and ADR_ACK_DELAY parameters of an end-device
0x0C	<b>ADRParamSetupAns</b>	x		Acknowledges ADRParamSetupReq command
0x0D	<b>DeviceTimeReq</b>	x		Used by an end-device to request the current date and time
0x0D	<b>DeviceTimeAns</b>		x	Sent by the network, answer to the DeviceTimeReq request
0x0E	<b>ForceRejoinReq</b>		x	Sent by the network, ask the device to Rejoin immediately with optional periodic retries
0x0F	<b>RejoinParamSetupReq</b>		x	Used by the network to set periodic device Rejoin messages
0x0F	<b>RejoinParamSetupAns</b>	x		Acknowledges RejoinParamSetupReq
0x80 to 0xFF	Proprietary	x	x	Reserved for proprietary network command extensions

Figura 36. Lista de comandos MAC.

Fuente: (Committee LoRa Alliance Technical, 2017).

Un dispositivo final deberá responder solamente una vez a cualquier comando MAC, en caso de que esta respuesta se pierda la red tendrá que volver a enviar el comando de nuevo. Además, cuando un comando MAC es iniciado por el dispositivo final, la red realiza su mejor esfuerzo para enviar su respuesta o ACK en una de las ventanas RX1/RX2 inmediatamente después de la petición (ver Figura 27).

### 3. CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL PROYECTO

En este capítulo se desarrollará el proyecto como tal, abordando el estado del arte de los módulos de IEEE 802.11ah y LoRa, actualmente disponibles para el simulador de red NS-3. Estos módulos se irán modificando conforme avanza el capítulo, hasta obtener un solo escenario de simulación con ambas tecnologías implementadas.

#### 3.1 ARQUITECTURA DE LOS MÓDULOS EN NS-3

Una de las limitantes del presente proyecto es la disponibilidad de hardware, contar con un gran número de end-devices que trabajen bajo los estándares IEEE 802.11ah y LoRa es sumamente complicado, no obstante, hoy en día existen mecanismos que permiten dar una solución alterna, el mecanismo escogido para realizar este trabajo es uso del simulador NS-3.

NS3 es el software de simulación de red escogido debido a su orientación hacia la investigación y el uso educativo, además, trabaja bajo una licencia GPL (General Public License) en un entorno Unix (Linux, Ubuntu, etc.) y está siendo constantemente desarrollado por una comunidad de usuarios que comparten sus aportes en múltiples repositorios virtuales, como es el caso de GitHub<sup>8</sup>.

La lógica detrás del simulador es la de combinar varios objetos de C++ con cada clase para modelar un aspecto de una red, de esta manera NS-3 puede simular redes complejas de manera detallada y realista. Las clases que modelan conceptos o sistemas relacionados se agrupan en módulos, por ejemplo, el módulo WiFi contiene varias clases que modelan componentes de un sistema WiFi, como: APs (Access Points), ED (end-devices), la capa WiFi MAC y un canal inalámbrico. Estas clases están interconectadas y cuando se combinan con otros módulos que modelan la funcionalidad de core, la

---

<sup>8</sup> GitHub es el principal repositorio usado para alojar proyectos y permite compartir el código fuente entre los miembros de su comunidad.

movilidad del dispositivo, la propagación, etc., se pueden usar para simular una red completa que implementa el estándar WiFi. (Magrin D. , 2016).

### 3.1.1 MÓDULO IEEE 802.11AH

El módulo de IEEE 802.11ah que se utilizará corresponde a un aporte de la comunidad de NS-3, por ende, no se encuentra en la documentación oficial como tal, en cambio, ha sido desarrollado por terceros sobre la versión 3.23 del simulador, tomando como base el modelo ya existente de IEEE 802.11n. Es así como el módulo de IEEE 802.11ah tiene su funcionamiento basado en cuatro clases principales: **WifiChannel**, **WifiPhy**, **MacLowLayer** y **MacHighLayer**.

Es importante recalcar que el módulo de IEEE 802.11ah ha sido desarrollado de forma tal que se encuentra junto con las anteriores revisiones de IEEE 802.11, es decir, no se instala como un módulo extra sino como una actualización del paquete NS-3 completo.

#### 3.1.1.1 MODELO PARA LA CAPA PHY IEEE 802.11AH

Las dos primeras clases principales que componen el módulo están dedicadas a modelar la capa PHY del protocolo. La clase **WifiChannel** (se la puede encontrar como `wifi-channel.cc` en el directorio `model`) se define como una aproximación del medio físico sobre el cual se produce la comunicación, incluyendo otros fenómenos como un modelo de pérdida de propagación (caracteriza la pérdida de intensidad de la señal durante la transmisión a través del medio) y un modelo de retardo (describe el retardo de la transmisión entre dos nodos). Por otro lado, la clase **WifiPhyLayer** (se la puede encontrar como `wifi-phy.cc` en la carpeta `model`) es la parte del módulo que define el formato del PPDU (PCLP Protocol Data Unit), además envía y recibe información a través del

**WifiChannel.** La clase **WifiPhyLayer** trabaja además con las clases **WifiPhy/YansWifiPhy**<sup>9</sup>, **InterferenceHelper** y **ErrorRateModel**.

Los distintos parámetros de MCS, que se observaron con detalle en la sección 2.1.1.1, se encuentran definidos en la clase **YansWifiPhy** (corresponde al fichero `yans-wifi-phy.cc` dentro de la ruta `/src/wifi/model/`). En la Figura 37 se encuentra un fragmento del extenso grupo de definiciones para cada ancho de banda de canal (1 a 16 MHz).

```
1018     if (GetChannelWidth () == 2)
1019     {
1020         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate300KbpsBW1MHz ());
1021         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate600KbpsBW1MHz ());
1022         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate900KbpsBW1MHz ());
1023         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate1_2MbpsBW1MHz ());
1024         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate1_8MbpsBW1MHz ());
1025         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate2_4MbpsBW1MHz ());
1026         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate2_7MbpsBW1MHz ());
1027         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate3MbpsBW1MHz ());
1028         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate3_6MbpsBW1MHz ());
1029         m_deviceRateSet.push_back (WifiPhy::GetOfdmRate4MbpsBW1MHz ());
```

Figura 37. Fragmento de la definición de MCS.  
Fuente: Github (Tian, y otros, 2018).

Otra de las clases con las que trabaja **WifiPhyLayer** es la **InterferenceHelper**, que como su nombre lo indica, es un objeto que permite calcular la interferencia, al tratarse de una clase que se usa en todas las versiones implementadas de IEEE 802.11 en NS-3, esta cuenta con múltiples variables a calcular, entre ellas: el SNR, la potencia del ruido interferente, el PER (Packet Error Rate) para PCLP, entre otras. En la Figura 38 se encuentra la función con la que se calcula el SNR, se puede observar que incluso se utiliza la constante de Boltzmann para el ruido térmico.

---

<sup>9</sup> YANS son las siglas para “Yet Another Network Simulator”, que es un simulador de red discontinuado de la época de NS-2, algunas de sus aportaciones pasaron a formar parte de NS-3.

```

236 double
237 InterferenceHelper::CalculateSnr (double signal, double noiseInterference, WifiMode mode) const
238 {
239     //thermal noise at 290K in J/s = W
240     static const double BOLTZMANN = 1.3803e-23;
241     //Nt is the power of thermal noise in W
242     double Nt = BOLTZMANN * 290.0 * mode.GetBandwidth ();
243     //receiver noise Floor (W) which accounts for thermal noise and non-idealities of the receiver
244     double noiseFloor = m_noiseFigure * Nt;
245     double noise = noiseFloor + noiseInterference;
246     double snr = signal / noise;
247     return snr;

```

Figura 38. Cálculo del SNR, extracto de las funciones de interferencia.  
Fuente: Github (*Tian, y otros, 2018*).

### 3.1.1.2 MODELO PARA LA CAPA MAC IEEE 802.11AH

Las dos clases principales restantes permiten modelar la capa MAC. La primera clase **MacLowLayer** implementa características como: mensajes RTS-CTS-DATA-ACK, la Función de Distribución Coordinada, el EDCA, parte de la funcionalidad de RAW, encolamiento, fragmentación y retransmisión. Por su parte, la clase **MacHighLayer** se encarga de administrar las funciones de generación de beacons, los mensajes de probe, asociación y parte de la funcionalidad de la RAW.

La funcionalidad de RAW al ser propia de IEEE 802.11ah, fue una adición completamente nueva a la funcionalidad de las clases **ApWifiMAC** (identificada como `ap-wifi-mac.cc` en `/src/wifi/model`) y **StaWifiMAC** (`sta-wifi-mac.cc` en `/src/wifi/model`) ya existentes dentro del simulador. Una muestra de esta adición a la clase **ApWifiMAC** se encuentra en la Figura 39, esta muestra un bucle dentro de la función usada para asignar cada RAW correspondiente.

```

601     for (uint8_t raw_index=0; raw_index < RAW_number; raw_index++)
602     {
603         RPS::RawAssignment ass = m_rpssset.rpssset.at(toTim->GetRawAssignmentObj(raw_index));
604         currentRAW_start += (500 + slotDurationCount * 120) * slotNum;
605         slotDurationCount = ass.GetSlotDurationCount();
606         slotNum = ass.GetSlotNum();
607         Time slotDuration = MicroSeconds(500 + slotDurationCount * 120);
608         lastRawDurationus += slotDuration * slotNum;
609         if (ass.GetRawGroupAIDStart() <= aid && aid <= ass.GetRawGroupAIDEnd()) {
610             uint16_t statRawSlot = (aid & 0x03ff) % slotNum;
611             Time start = MicroSeconds((500 + slotDurationCount * 120) * statRawSlot + currentRAW_start);

```

Figura 39. Extracto de la función para asignar una RAW.  
Fuente: Github (*Tian, y otros, 2018*).

### 3.1.2 MÓDULO LORA

Este módulo originalmente no forma parte del paquete NS-3, por lo que es necesario instalarlo como un módulo extra. Comprende dos modelos principales: uno para la capa LoRa PHY que necesita representar los chips LoRa y el comportamiento de las transmisiones LoRa, y otro para la capa LoRaWAN MAC, que necesita comportarse de acuerdo con las especificaciones oficiales (ver Capítulo II).

Para representar estos dos modelos, el módulo presenta dos clases base genéricas **LoraPhy** y **LorawanMac**. Estas clases se amplían luego con clases que modelan las peculiaridades de los dos dispositivos de red inalámbrica soportados: el end-device (ED) y el Gateway (GW), en este sentido, las capas PHY se pueden modelar mediante el uso de las clases **EndDeviceLoraPhy** y **GatewayLoraPhy**. Por otro lado, objetos de la clase **EndDeviceLorawanMac**, **ClassAEndDeviceLorawanMac** y **GatewayLorawanMac** se utilizan para representar la capa MAC.

#### 3.1.2.1 MODELO PARA LA CAPA PHY LORA

Al modelar el enlace se toman en cuenta tres componentes para determinar el rendimiento de la transmisión LoRa, estos son: las especificaciones sobre la sensibilidad del dispositivo obtenidos directamente de sus datasheets, un modelo para contabilizar la interferencia entre diferentes transmisiones LoRa y una serie de supuestos sobre este modelo de interferencia.

Para comprender el modelo de la capa PHY es necesario introducir primero la clase **LoraChannel** (lora-channel.cc dentro del directorio model), esta se utiliza para interconectar las capas **LoraPHY** de todos los dispositivos LoRa que desean comunicarse. Durante el proceso de comunicación, la clase **LoraChannel** mantiene una lista de las capas PHY conectadas y las notifica sobre transmisiones entrantes, las capas PHY conectadas al canal exponen un método público **StartReceive** que le permite al

canal empezar la recepción a una PHY específica. Por consiguiente, la capa PHY notifica al objeto llamado **LoraInterferenceHelper** para realizar un seguimiento de todas las señales entrantes (deseadas o interferencia). Una vez que se han notificado entre clases, se comprueba si el PHY cumple con los siguientes requisitos para continuar con la recepción:

1. El receptor debe estar inactivo (en estado STANDBY) cuando la función **StartReceive** ha sido llamada.
2. La potencia de la señal recibida deberá ser mayor al umbral de sensibilidad.
3. El receptor deberá estar escuchando en la frecuencia correcta.
4. El receptor deberá estar escuchando por el SF correcto.

Con respecto al requisito 2, la clase **LoraPHY** modela la capa física de un end-device y un Gateway, basándose en el comportamiento de los chips SX1272 y SX1301 respectivamente. Los parámetros de sensibilidad del end-device se encuentran definidos en la clase **EndDeviceLoraPhy** (archivo end-device-lora-phy.cc en el directorio model). En la Figura 40 y 41 se puede notar que se definen 6 valores para el umbral de sensibilidad, uno para cada SF (SF7 – SF12).

```
81 // Downlink sensitivity (from SX1272 datasheet)
82 // {SF7, SF8, SF9, SF10, SF11, SF12}
83 // These sensitivities are for a bandwidth of 125000 Hz
84 const double EndDeviceLoraPhy::sensitivity[6] =
85 {-124, -127, -130, -133, -135, -137};
```

Figura 40. Parámetros de sensibilidad para end-device SX1272.  
Fuente: Github (Magrin, Capuzzo, & Zanella, 2020).

En el caso del Gateway, sus valores de sensibilidad son diferentes (hacen referencia al chip SX1301) y son definidos por la clase **GatewayLoraPhy** (gateway-lora-phy.cc). Observando con detenimiento la Figura 41, se puede notar que el Gateway tiene una mayor sensibilidad que los end-devices, siendo capaz de captar señales hasta los -142.5 dBm.

```

133 // Uplink sensitivity (Source: SX1301 datasheet)
134 // {SF7, SF8, SF9, SF10, SF11, SF12}
135 const double GatewayLoraPhy::sensitivity[6] =
136 {-130.0, -132.5, -135.0, -137.5, -140.0, -142.5};

```

Figura 41. Parámetros de sensibilidad para Gateway SX1301.  
Fuente: Github (Magrin, Capuzzo, & Zanella, 2020).

Continuando con el proceso de comunicación y luego de que la capa PHY haya asegurado la señal entrante, se programa una llamada a la función **EndReceive**. Cuando la recepción concluye, la función **EndReceive** llama al método **IsDestroyedByInterference** que forma parte del objeto **LoraInterferenceHelper** y permite determinar si la información transmitida se pierde debido a la interferencia o no. La función **IsDestroyedByInterference** compara la potencia de recepción de la señal deseada con la energía de interferencia de las señales que se superponen en un determinado SF, luego compara el valor obtenido de SIR (Signal-to-interference ratio) con la matriz de colisión (Goursaud & Gorce, 2015).

Por ejemplo, si la señal deseada usa un SF7 y se solapa con dos señales que usan un SF10, entonces, la energía de la señal deseada (calculada como el producto de la potencia de recepción y la duración de la señal) se compara con la energía sumada de las dos señales interferentes (calculada como el producto de la potencia de la interferente en el receptor y el tiempo de solapamiento). La relación entre la energía de la señal deseada y la interferente para cada SF se compara con la matriz de la Figura 42, donde cada fila corresponde al SF de la señal deseada y las columnas al SF de la interferente. Si el SIR es mayor al umbral de la matriz, el paquete se recibe correctamente y es reenviado a la capa MAC.

```

128 // LoRa Collision Matrix (Goursaud)
129 // Values are inverted w.r.t. the paper since here we interpret this as an
130 // _isolation_matrix instead of a cochannel_rejection_matrix like in
131 // Goursaud's paper.
132 std::vector<std::vector<double>> LoraInterferenceHelper::collisionSnirGoursaud= {
133     // SF7 SF8 SF9 SF10 SF11 SF12
134     {6, -16, -18, -19, -19, -20}, // SF7
135     {-24, 6, -20, -22, -22, -22}, // SF8
136     {-27, -27, 6, -23, -25, -25}, // SF9
137     {-30, -30, -30, 6, -26, -28}, // SF10
138     {-33, -33, -33, -33, 6, -29}, // SF11
139     {-36, -36, -36, -36, -36, 6} // SF12
140 };

```

Figura 42. Matriz de Colisión para determinar la interferencia en la transmisión.  
Fuente: Github (Magrin, Capuzzo, & Zanella, 2020).

### 3.1.2.2 MODELO PARA LA CAPA MAC LORA

Con el objetivo de implementar el estándar LoRaWAN, han sido desarrolladas múltiples clases para manejar headers, comandos MAC, canales lógicos y cálculos del ciclo de trabajo. La estructura de paquetes definida por el estándar se implementa en el simulador a través de 2 clases: **LorawanMacHeader** (extensión de la clase Header) y **LoraFrameHeader** (incluye los comandos MAC mediante las clases MacCommand y LoraDeviceAddress).

Los comandos MAC están implementados extendiendo la clase **MacCommand** que viene por defecto en el simulador. Cada clase hijo es usada para definir un conjunto de variables y métodos. Por otro lado, la clase **LoraDeviceAddress** se utiliza para representar la dirección de un end-device de LoRa.

El módulo de LoRa para NS-3 fue desarrollado tomando en cuenta la operación en bandas sin licencia que están sujetas a restricciones del ciclo de trabajo (región europea), en este sentido, se creó una serie de objetos para mantener un seguimiento del tiempo de transmisión disponible y el límite de transmisión en la capa MAC (en caso de que las capas superiores no tengan en cuenta estas limitaciones). Para ello, un objeto **LogicalLoraChannelHelper** es asignado a cada instancia de la clase **LorawanMac** para cumplir con el seguimiento de todos los canales lógicos.

En cuanto a las limitaciones del ciclo de trabajo, un objeto registra todas las transmisiones realizadas en cada canal para determinar cuándo se podrá volver a

transmitir de acuerdo con las regulaciones. Es decir, si una transmisión tiene una duración  $t_{air}$  dentro de un canal con un ciclo de trabajo es expresado como  $dc$ , entonces, el tiempo que el dispositivo necesitará permanecer fuera de trabajo ( $t_{off}$ ) se calculará como la Ec. 13.

$$t_{off} = \frac{t_{air}}{dc} - t_{air} \quad (\text{Ec. 13})$$

### 3.2 MODIFICACIÓN DE LAS CLASES DENTRO DEL SIMULADOR

Antes de pasar a modificar las clases dentro del simulador, primero es necesario conocer cómo se encuentran organizadas, tanto jerárquica como lógicamente las clases de ambos módulos del simulador.

Para el caso de IEEE 802.11ah, en su diagrama de clases (ver Figura 43) se tiene como clase principal o “main” a la llamada **s1g-test-tim-raw**, entre sus parámetros define variables relacionadas con los grupos de RAW/RPS, así como números y tiempos de asociación para cada estación, por otro lado, sus métodos permiten la des/asociación, configurar los parámetros de RAW, enviar estadísticas e incluso llenar el caché ARP de los dispositivos finales simulados. Si bien la clase principal cuenta con ciertos parámetros propios de la simulación, existe una clase completamente dedicada a los parámetros de una configuración por defecto, esta se llama **Configuration**. Entre los parámetros aquí definidos está el tiempo de simulación, el tamaño del payload y el radio máximo de aparición de las estaciones (representado por la variable rho), esta clase además provee a las demás de información delicada como son las rutas de ficheros y directorios de los cuales obtener o almacenar datos.

De manera adicional, la clase main se ayuda de las clases **Statistics** y **SimulationEventManager**, la primera encargándose de contabilizar la información de la simulación en una escala mayor, como es el tiempo total de simulación, cuando se asocia

cada STA, la cantidad de nodos o su posición. Por su parte, **SimulationEventManager** supervisa los eventos (creación de nodos, asociaciones, transmisiones, entre otros) y envía esta información a otras clases (SimpleTCPCliente) o como **output**<sup>10</sup>, para extraer la información de la simulación hacia un visualizador (ahVisualizar) o en un formato de texto plano se usan los parámetros definidos (hostname, port, filename).

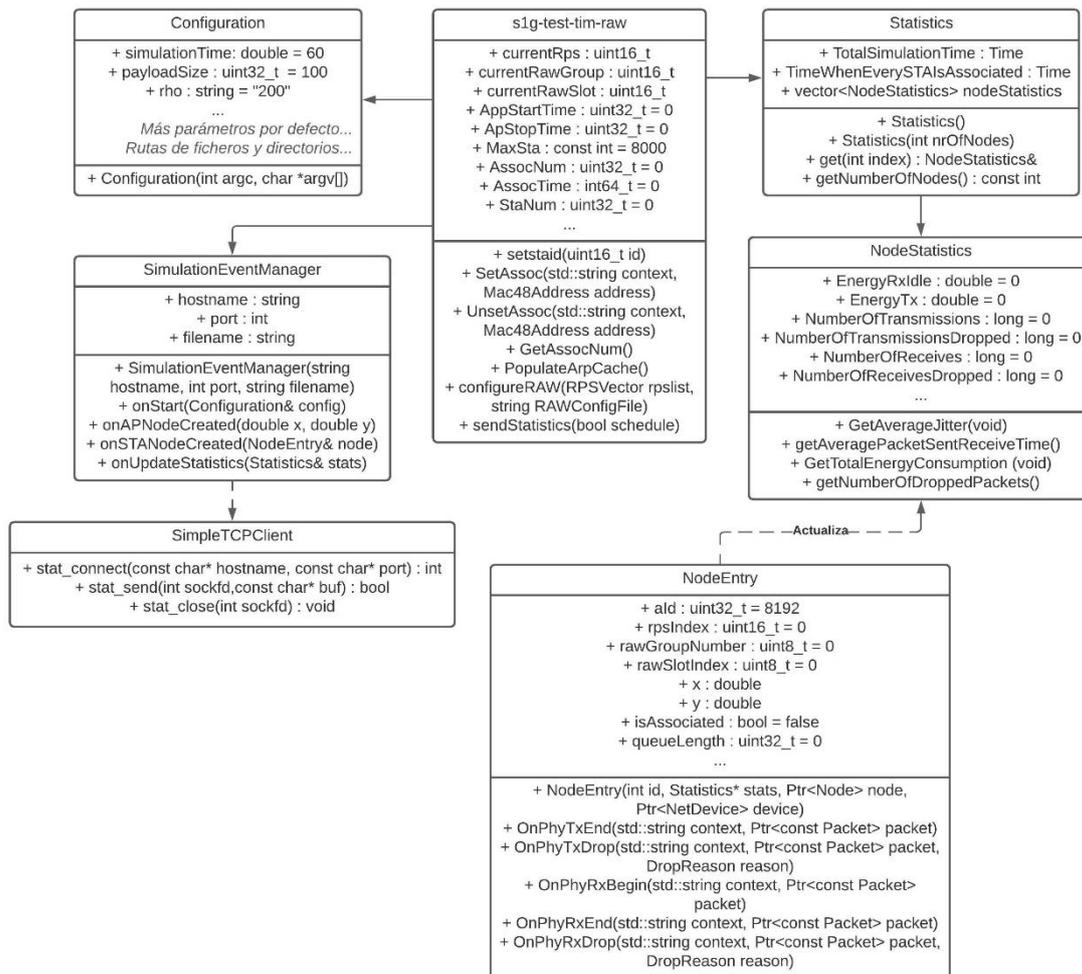


Figura 43. Diagrama de clases módulo IEEE 802.11ah en NS-3.  
Fuente: Autor.

La clase **Statistics** trabaja a su vez directamente con otra clase identificada como **NodeStatistics**, está última contabiliza información de la simulación en una escala menor, por cada nodo. Permitiendo obtener datos como la potencia/energía durante la transmisión/recepción, el número de transmisiones/recepciones, e incluso la cantidad de

<sup>10</sup> Output se refiere a la salida del sistema o resultados obtenidos por el mismo; por ejemplo, una tabla de datos.

pérdidas durante la comunicación. Para obtener esta gran cantidad de información estadística, la clase NodeEntry envía, a través de métodos específicos, actualizaciones de estado durante los procesos de Tx y Rx (transmisión y recepción).

Por otra parte, el diagrama para el módulo de LoRaWAN (ver Figura 44) tiene claras diferencias con el de la Figura 43, de las que se pueden resaltar dos: no existe una clase exclusiva de configuración (cada clase define los parámetros que necesite) y la contabilización de estadísticas se realizan por separado entre end-devices y Gateways. En este sentido, la clase principal es **complete-network-example**, siendo esta la que define los parámetros de una simulación por defecto (número de dispositivos finales, número de Gateways, radio de distribución de equipos, tiempo de simulación, entre otros). La clase main se apoya directamente de las clases **end-device-status** y **gateway-status**, ambas cumplen con funciones análogas, como establecer/obtener (set/get) una dirección IP o MAC, sin embargo, **end-device-status** cuenta con funciones exclusivas relacionadas con la primera y segunda ventana de recepción.

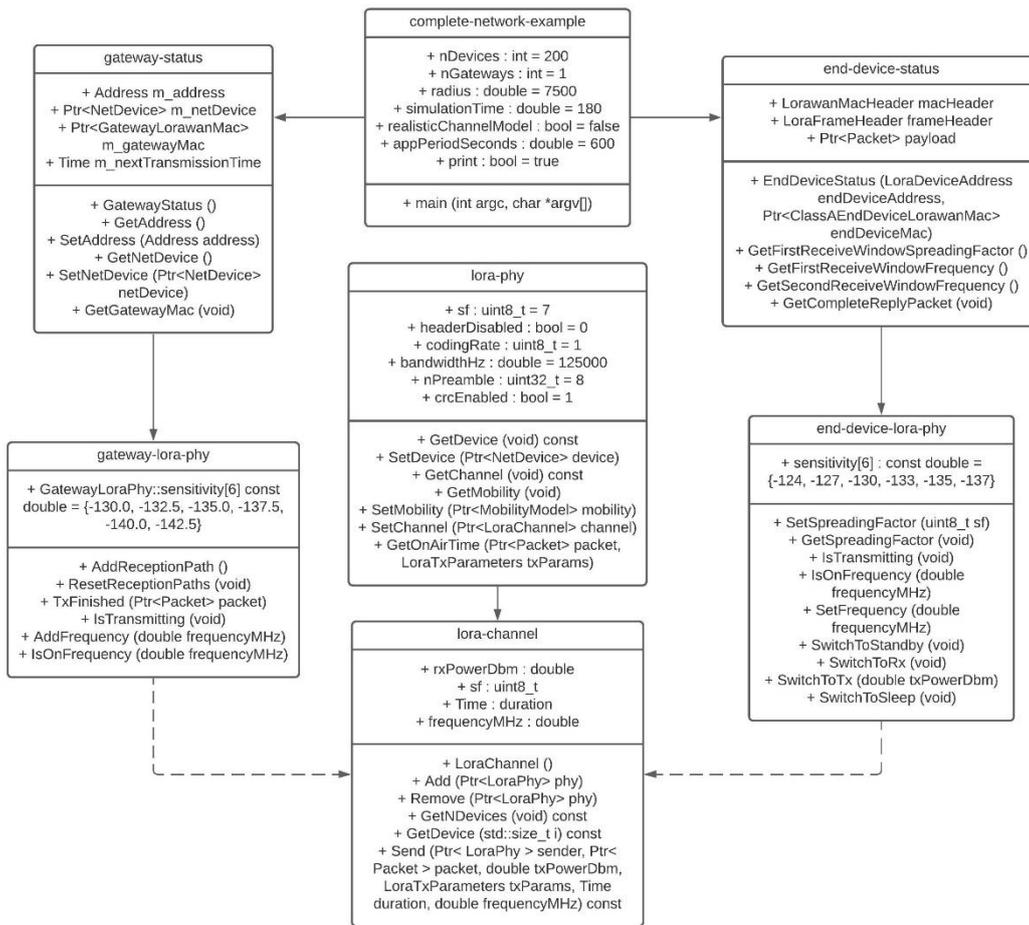


Figura 44. Diagrama de clases módulo LoRaWAN en NS-3.  
Fuente: Autor.

Las clases que más se relacionan con la abstracción de la capa física son **gateway-lora-phy** y **end-device-lora-phy**, como muestra, entre sus parámetros definen la sensibilidad que manejará cada SF (SF7 – SF12). La clase **gateway-lora-phy** se encarga de establecer el path físico de comunicación, agregar las frecuencias de funcionamiento y validar que las comunicaciones estén en las frecuencias de operación. Por su parte, **end-device-lora-phy** establece además de la frecuencia de trabajo, el Spreading Factor a utilizar y maneja los cambios de estado durante la simulación (ACTIVE, SLEEP).

La clase **lora-channel** con ayuda de **lora-phy** simulan el medio de transmisión mediante parámetros que permite identificar el SF, el canal de frecuencia o el ancho de banda, dichos criterios son obtenidos a mediante funciones específicas. Mientras que **lora-phy** puede obtener información clave como lo es el tiempo en el aire de la señal,

**lora-channel** define la función para enviar el “paquete” (señal) desde una instancia de end-device hacia el Gateway. En la Figura 44, se pueden identificar los parámetros necesarios para usar la función **Send**, siendo estos, la instancia de lora-phy que envía, la instancia del “paquete” (señal), la potencia de transmisión, la duración de la transmisión, la frecuencia y un conjunto de parámetros de transmisión adicionales (SF, canal, ganancia, entre otros.).

La funcionalidad de las clases en la Figura 43 puede ser más fácilmente comprendida con el diagrama de secuencia de la Figura 45, en el centro se encuentra ubicada la clase principal, está inicia y detiene la simulación con sus métodos asociados (::Run y ::Stop). En su lado izquierdo, se registran los eventos dentro de la simulación y se colabora con el cliente TCP para realizar la comunicación. Mientras que, en el lado derecho se establecen parámetros de configuración y se contabilizan los datos asociados a cada evento y cada nodo.

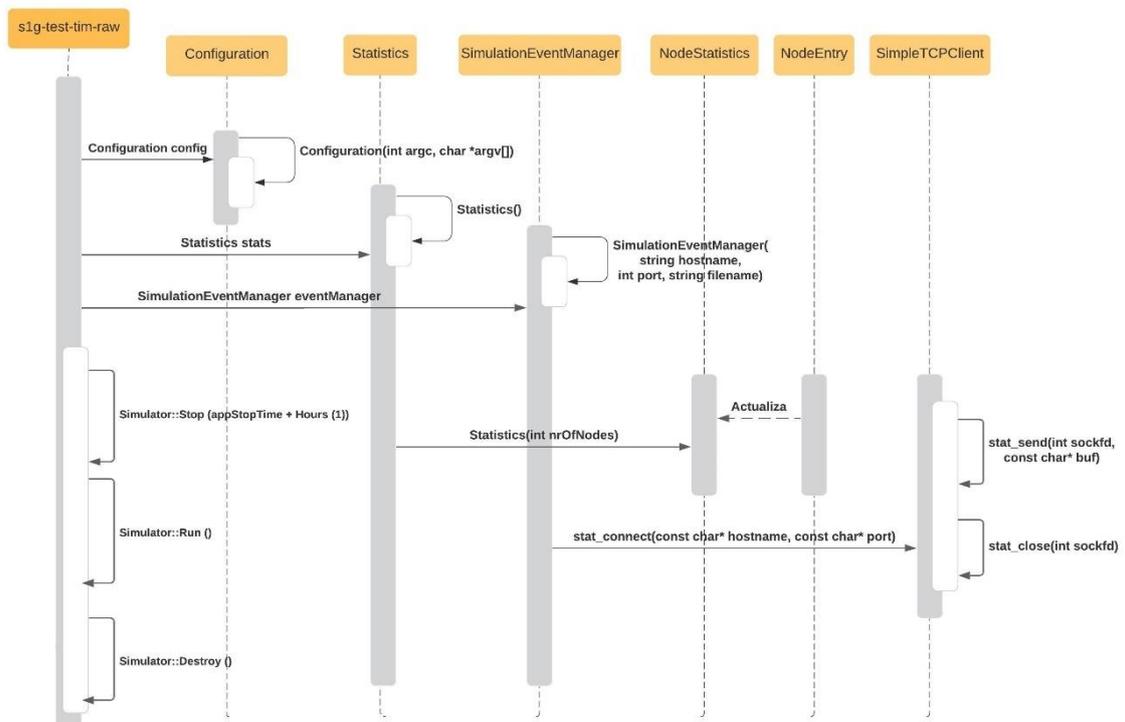


Figura 45. Diagrama de secuencia módulo IEEE 802.1ah en NS-3.  
Fuente: Autor.

El funcionamiento de la simulación de LoRaWAN de acuerdo con la Figura 46 tiene su inicio en la clase **complete-network-example**, esta clase contiene toda la configuración necesaria para simular una red con una gran cantidad de end-devices (200 por defecto) y un solo Gateway. Durante la simulación, las clases status contabilizan las estadísticas que obtienen de las instancias de **gateway-lora-phy** y **end-device-lora-phy**, mientras que la clase **lora-channel** maneja las transmisiones con ayuda de la clase **lora-phy**.

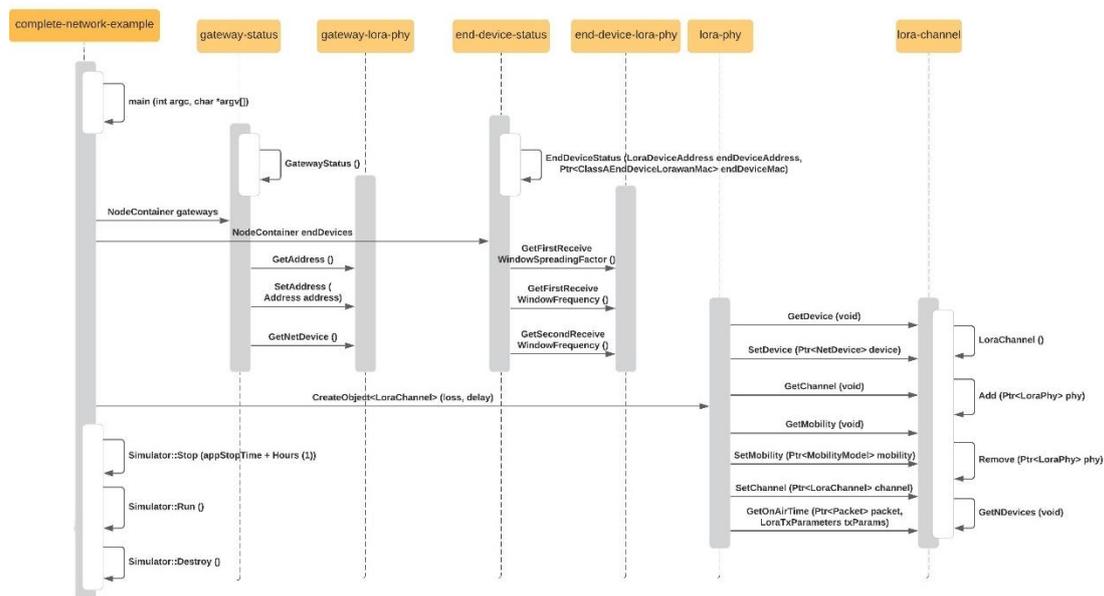


Figura 46. Diagrama de secuencia módulo LoRaWAN en NS-3.  
Fuente: Autor.

Una vez comprendido como se relacionan las distintas clases para cada módulo se procede a realizar el primer cambio significativo. El módulo de LoRaWAN utilizado para el desarrollo del presente proyecto simula por defecto en la banda de frecuencia que corresponde a Europa (868 MHz). De modo que es necesario identificar la clase encargada de asignar las frecuencias de trabajo y editarla para tener compatibilidad con la región de Norteamérica.

Siguiendo las indicaciones del diagrama de clases se busca la configuración del canal de operación dentro de la clase principal **complete-network-example**. Estas clases se identifican como “helpers” y permiten realizar configuraciones a las capas PHY y

MAC, más específicamente, la clase LoraPhyHelper (ver Figura 47) es aquella que definirá las bandas y sub-bandas de operación.

```
/* *****  
 * Create the helpers *  
***** */  
  
// Create the LoraPhyHelper  
LoraPhyHelper phyHelper = LoraPhyHelper ();  
phyHelper.SetChannel (channel);  
  
// Create the LorawanMacHelper  
LorawanMacHelper macHelper = LorawanMacHelper ();
```

Figura 47. Creación de las clases helper para las capas PHY y MAC.  
Fuente: Github (Magrin, Capuzzo, & Zanella, 2020).

De acuerdo con el árbol de ficheros para el módulo de LoRaWAN, el helper buscado se encuentra en /src/lorawan/helper/lora-phy-helper.cc. La modificación se realiza en el método **LoraPhyHelper::Set**, se declara el vector que almacenará los Receive Paths (trayectorias de recepción) y en las líneas subsecuentes se determinan sus frecuencias respectivas que escuchará el Gateway.

```
std::vector<double> frequencies;  
frequencies.push_back (902.3);  
frequencies.push_back (902.5);  
frequencies.push_back (902.7);
```

Figura 48. Cambio de las frecuencias de los Receive Paths.  
Fuente: Autor.

El siguiente helper modificado pertenece a la capa MAC, se encuentra en /src/lorawan/helper/lorawan-mac-helper.cc. Este helper se encargará de sobrescribir algunas de las configuraciones por defecto de las otras clases, esto incluye la frecuencia de operación y sus respectivas sub-bandas, el módulo al haber sido diseñado para simular una operación en la banda europea no cuenta con los métodos necesarios para trabajar en otra banda, de modo que las modificaciones que se realizarán a continuación están dirigidas a crear un método propio para la región norteamericana.

Primero, dentro del fichero **lorawan-mac-helper.cc** se modificó el constructor para que la región por defecto sea US (norteamericana) en lugar de EU (europea) como

estaba en un inicio. Esto permitirá que el objeto de la clase **LorawanMacHelper** en el resto del código configure a los dispositivos finales y Gateway en la región deseada (ver Figura 49).

```
//Región por defecto US
LorawanMacHelper::LorawanMacHelper () : m_region (LorawanMacHelper::US)
{
}
```

Figura 49. Cambio de la región por defecto en el constructor.  
Fuente: Autor.

Como continuación directa del paso anterior, se añaden dos nuevas opciones a la definición de end-device y de Gateway (ver Figura 50). Los cuales indican que al tener la clase **LorawanMacHelper** configurada con la región US se procede a llamar a la función **ConfigureForUSRegion()**, misma que será creada en el siguiente paso. El mismo procedimiento se ha realizado para el caso del Gateway.

```
//Case para la región US - end device //Case para la región US - gateway
case LorawanMacHelper::US: {           case LorawanMacHelper::US: {
    ConfigureForUSRegion (edMac);       ConfigureForUSRegion (gwMac);
    break;                               break;
}                                       }
```

Figura 50. Configuración de end-device y Gateway en la región US.  
Fuente: Autor.

La función **ConfigureForUSRegion()** se crea siguiendo la misma estructura de la función **ConfigureForEuRegion()**, cambiado solamente la frecuencia de la segunda ventana de recepción a 923.30 MHz (primer canal de downlink disponible) para los end-devices (ver Figura 51). Para la función que configura los Gateways no se deberá definir la ventana de recepción, sino las trayectorias de recepción, tal y como se lo hizo en la Figura 48.

```
////////////////////////////////////
// Second receive window parameters //
////////////////////////////////////
edMac->SetSecondReceiveWindowFrequency (923.30);
```

Figura 51. Cambio de la frecuencia de la segunda ventana de recepción.  
Fuente: Autor.

La última función que se debe agregar es **ApplyCommonUSConfigurations**, de igual manera, basada en la clase **ApplyCommoEuConfigurations** que sirvió de guía para

el desarrollo de esta. Dentro de la sección de sub-bandas (ver Figura 52) se encuentra una instancia de la clase **LogicalLoraChanelHelper**, que define las frecuencias de cada sub-banda siguiendo la estructura de la función **AddSubBand** (frecuencia inicial, frecuencia final, ciclo de trabajo y máxima potencia de transmisión en dBm). Cada sub-banda se conformará de 8 canales de subida de 125 kHz con un ciclo de trabajo por defecto de 1% (1 -> 100%, entonces, 0.01 -> 1%).

```

////////////////////
// SubBands //
////////////////////
LogicalLoraChannelHelper channelHelper;
channelHelper.AddSubBand (902.2, 903.7, 0.01, 14);
channelHelper.AddSubBand (903.9, 905.3, 0.01, 14);
channelHelper.AddSubBand (905.5, 906.9, 0.01, 27);

```

Figura 52. Cambio en la asignación de sub-bandas.  
Fuente: Autor.

Para finalizar, en la sección de canales por defecto se definen los canales que usará la clase **LogicalLoraChanelHelper**, esta configuración debe ser la misma que los Receive Paths (ver Figura 48) en la clase **LoraPhyHelper**. Cada una de las sub-bandas (ver Figura 53) fueron asignadas en base a un artículo acerca de frecuencias, canales y sub-bandas de LoRaWAN en USA para dispositivos IoT (Barany, 2019).

```

////////////////////
// Default channels //
////////////////////
Ptr<LogicalLoraChannel> lc1 = CreateObject<LogicalLoraChannel> (902.3, 0, 5);
Ptr<LogicalLoraChannel> lc2 = CreateObject<LogicalLoraChannel> (902.5, 0, 5);
Ptr<LogicalLoraChannel> lc3 = CreateObject<LogicalLoraChannel> (902.7, 0, 5);
channelHelper.AddChannel (lc1);
channelHelper.AddChannel (lc2);
channelHelper.AddChannel (lc3);

```

Figura 53. Cambio en los canales lógicos.  
Fuente: Autor.

### 3.3 DESARROLLO DE UN SOLO MODELO DE SIMULACIÓN

Una vez que ambos módulos (IEEE 802.11ah y LoRa) se encuentran funcionando dentro del simulador en la banda de frecuencia asignada para la región US por la FCC (902 MHz – 928 MHz), el siguiente paso es preparar el módulo de IEEE 802.11ah para que importe los datos correspondientes a la simulación de LoRa. Para ello, primero se

agregó el código respectivo al módulo de LoRa de modo que cada iteración de simulación genere como resultado un fichero de texto plano con el tiempo inicial y final de la transmisión, además de la potencia de la señal. Dicho código se aplica en el fichero **lora-interference-helper.cc** (aprovechando que maneja todas las transmisiones como eventos) y se muestra en la Figura 54.

```
// Imprimir informacion de interferencia
ofstream interfInfo;
string rutaInfo = "./Resultados/interfLora_templ.txt";
interfInfo.open(rutaInfo.c_str(), ios::out | ios::app);
interfInfo << event->GetStartTime () << " " << event->GetEndTime () << std::endl;
interfInfo.close();
```

Figura 54. Bloque de código para exportar datos de la simulación de LoRa.  
Fuente: Autor.

Estos datos se almacenan en el directorio **Resultados** bajo el nombre **interfLora**, en formato de texto plano. Por facilidad se ha delimitado cada parámetro por un espacio en blanco, esto se puede observar en la Figura 55, donde se encuentran las 10 primeras entradas del fichero. Es importante mencionar que los eventos de tiempo se manejan en el orden de los nanosegundos (manteniendo el formato que también maneja el módulo IEEE 802.11ah). Otros de los datos extraídos de la simulación de LoRa son las coordenadas de la estación que está utilizando el canal en ese instante, la potencia de transmisión en dBm y la frecuencia de la sub-banda utilizada.

```
1 +1095430000.0ns +1167360000.0ns 293.125 55.5308 14 902.3
2 +4906260000.0ns +4978200000.0ns 150.638 263.036 14 902.7
3 +6983240000.0ns +7055170000.0ns 241.235 182.704 14 902.3
4 +22805100000.0ns +22877000000.0ns 333.455 241.808 14 902.5
5 +25297400000.0ns +25369300000.0ns 259.684 122.174 14 902.3
6 +25601600000.0ns +25673500000.0ns 186.719 138.256 14 902.7
7 +26004500000.0ns +26076500000.0ns 210.92 209.17 14 902.5
8 +32915200000.0ns +32987200000.0ns 339.995 276.193 14 902.3
9 +36271300000.0ns +36343200000.0ns 49.6094 120.268 14 902.3
10 +36292100000.0ns +36364000000.0ns 264.294 288.272 14 902.5
```

Figura 55. Datos exportados de la simulación LoRa.  
Fuente: Autor.

Los datos obtenidos de cada simulación de LoRa son luego importados a la simulación de IEEE 802.11ah desde el fichero **yans-wifi-phy.cc**. La sección idónea para agregar el código respectivo ha sido la función **EndReceive** de la clase **YansWifiPhy**,

que es la que se encarga de validar y terminar la recepción de la señal IEEE 802.11ah recibida.

Para determinar si alguna de las transmisiones de LoRa simuladas puede actuar como interferencia hacia las transmisiones de IEEE 802.11ah, se identifica en primer lugar si la transmisión actual de IEEE 802.11ah (señal deseada) se solapla con alguna de las transmisiones de LoRa (señal interferente) en el fichero importado (interfLora.txt).

Con el fin de entender de mejor manera la lógica utilizada para determinar el solapamiento de las señales, se han dividido las distintas posibilidades en casos. El primero de estos casos corresponde un escenario en donde la señal deseada ha sido transmitida en su totalidad (enviada y recibida) antes de que la señal interferente haya empezado a transmitirse (ver Figura 56), es decir que no existe solapamiento.

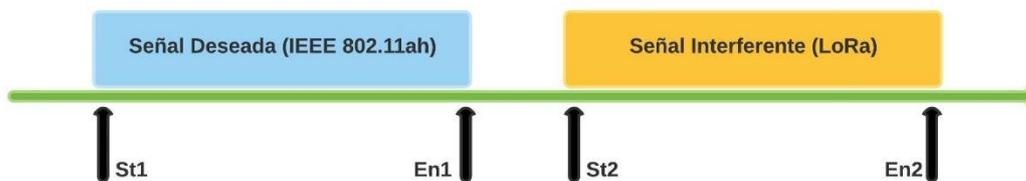


Figura 56. Caso 1: la señal deseada se transmite antes que la interferente, no hay solapamiento.  
Fuente: Autor.

El caso 2 es análogo al caso 1 (tampoco existe solapamiento), con la diferencia de que la señal interferente ha sido transmitida en su totalidad antes que la señal deseada (ver Figura 57). Para el resto de los casos los tiempos de inicio (St) y de final (En) de la señal deseada serán denominados St1 y En1, respectivamente, mientras que los de la señal interferente serán St2 y En2. De igual manera, la transmisión de la señal deseada se tratará como evento 1 y la señal interferente corresponderá a evento 2.

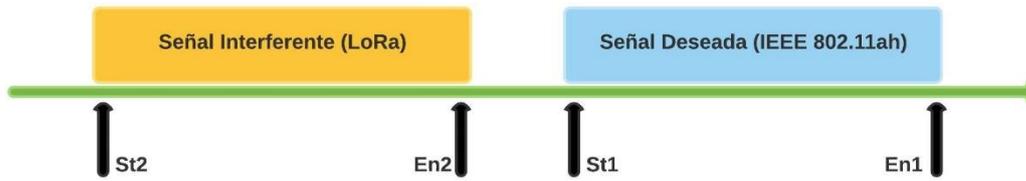


Figura 57. Caso 2: la señal interferente se transmite antes que la deseada, no hay solapamiento.  
Fuente: Autor.

En el caso 3 si existe solapamiento. Debido a que el evento 2 es transmitido durante el evento 1 y se cumple la condición  $St1 < St2$ , de modo que la duración del solapamiento se calcula como  $En2 - St2$  (solamente si  $En2 < En1$ ), en otras palabras, es la duración completa del evento 2. Gráficamente este escenario se encuentra representado por la Figura 58.



Figura 58. Caso 3: la señal interferente se transmite durante la señal deseada, sí hay solapamiento.  
Fuente: Autor.

Partiendo de la condición anterior ( $St1 < St2$ ) se establece el caso 4 (ver Figura 59), en este escenario no se cumple  $En2 < En1$ , haciendo que el tiempo de solapamiento se obtenga como  $En1 - St2$ .

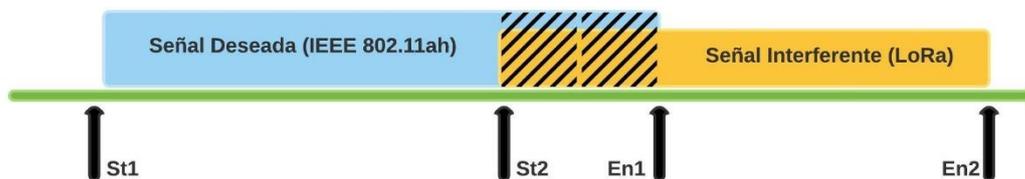


Figura 59. Caso 4: la señal interferente se transmite antes que finalice la señal deseada, sí hay solapamiento.  
Fuente: Autor.

De forma inversa a lo que se había observado en un caso anterior (caso 3), en el caso 5, el evento 1 se transmite durante el evento 2, por lo cual la duración del solapamiento se obtiene de  $En1 - St1$ , tal como de muestra en la Figura 60.

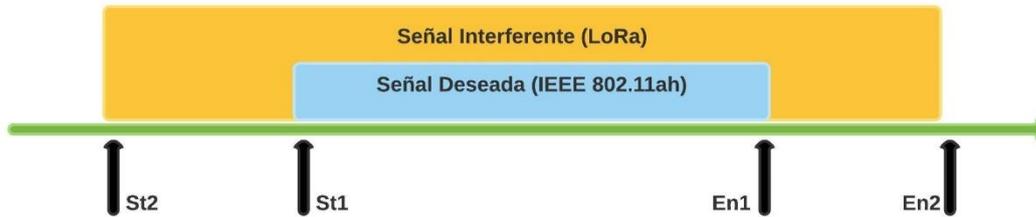


Figura 60. Caso 5: la señal deseada se transmite durante la señal interferente, sí hay solapamiento.  
Fuente: Autor.

Finalmente, en la Figura 61 se encuentra el caso 6, este corresponde a un escenario donde el evento 1 empieza a transmitirse durante el evento 2 y termina después de este. Para calcular la duración del solapamiento se usa  $En2 - St1$ .

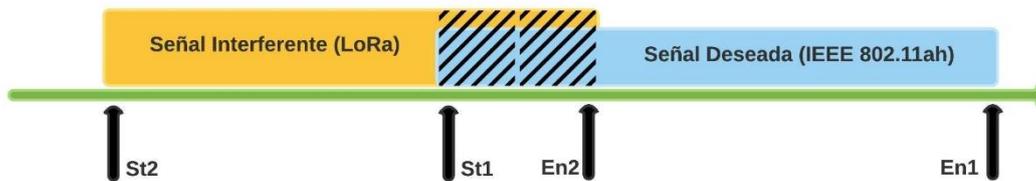


Figura 61. Caso 6: la señal deseada se transmite antes que finalice la señal interferente, sí hay solapamiento.  
Fuente: Autor.

La lógica de programación correspondiente al proceso de identificar y calcular el solapamiento de una señal IEEE 802.11ah con una de LoRa se encuentra explicado a detalle en el Procedimiento 1. Esto incluye tanto las variables necesarias para su ejecución como los condicionales y output.

---

**Procedimiento 1.** Identificar y calcular el solapamiento.

---

**Input:** st1 = tiempo inicial señal deseada; en1 = tiempo final señal deseada

**Input:** st2 = tiempo inicial señal interferente; en2 = tiempo final señal interferente

1: banderah = false

2: **if** en1 <= st2

3: Termina el proceso de solapamiento

```

4: else if en2 <= st1
3:   overlap_1 = 0
5:   banderah = false
6: else if st1 < st2
7:   if en2 < en1
8:     overlap_1 = en2 - st2
9:     banderah = true
10:    contBand++
11:  else
12:    overlap_1 = en1 - st2
13:    banderah = true
14:    contBand++
15: else
16:  if en1 < en2
17:    overlap_1 = en1 - st1
18:    banderah = true
19:    contBand++
20:  else
21:    overlap_1 = en2 - st1
22:    banderah = true
23:    contBand++
24: return overlap_1

```

---

En caso de que, sí se haya detectado un solapamiento entre las señales deseada e interferente, se comienza a contabilizar la energía de la interferencia de LoRa sobre IEEE 802.11ah. Para hacerlo se toman los datos, todas las columnas del archivo de interferencias LoRa (interfInfo.txt), dichos datos corresponden a: tiempo de inicio de la transmisión, tiempo final de la transmisión, coordenadas “X” e “Y” de la estación LoRa que transmitió la señal, la potencia en dBm con la que lo hizo y la frecuencia de la sub-banda.

Mediante un objeto de la clase **MobilityModel** se obtiene la posición exacta de la estación IEEE 802.11ah que está sufriendo interferencia, con la posición de ambos nodos (IEEE 802.11ah y LoRa) se calcula la distancia que existe entre ellos con la ecuación de la distancia entre dos puntos<sup>11</sup>. La distancia entre ambas estaciones será necesario para calcular el Path Loss<sup>12</sup>, el modelo matemático utilizado para calcularlo es el Log Distance Propagation Loss Model, dado que permite predecir el Path Loss de una señal dentro de un área densamente poblada. La Ec.14 corresponde al modelo descrito, donde:  $n_{PL}$  es el exponente de Path Loss, tiene un valor de 3 que corresponde a una zona de radio urbana;  $d$  es la distancia entre las estaciones;  $d_{ref}$  es una distancia referencial de 1m;  $\delta$  es el factor de corrección que permite usar el modelo con otras frecuencias que no sean exactamente 900MHz, se encuentra más a detalle en la Ec. 6 en el Capítulo II.

$$Path Loss (dB) = 10 * n_{PL} * \log_{10} \left( \frac{d}{d_{ref}} \right) + \delta \quad (\text{Ec. 14})$$

Al valor de Path Loss obtenido de la Ec. 14 se le adiciona la pérdida de referencia 30 dB en la distancia de referencia  $d_{ref}$  para una frecuencia alrededor de los 900 MHz (Abdul Rahman & Abdul Aziz, 2014). El resultado final será el coeficiente de atenuación; bastará con sumar (o restar) la potencia de transmisión de LoRa con el coeficiente de atenuación para obtener la potencia real que detectará el receptor de la estación IEEE 802.11ah.

La potencia obtenida del paso anterior es convertida de dBm a Watts ( $P(W) = 1W \cdot 10(P(dBm) / 10) / 1000$ ), para luego calcular la energía de esta interferencia a través del producto de la potencia (Watts) y la duración del solapamiento (segundos). Tomando en cuenta que una misma señal deseada (IEEE 802.11ah) puede sufrir

---

<sup>11</sup> La distancia entre dos puntos se calcula como:  $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ .

<sup>12</sup> El Path Loss o Pérdida de Trayectoria, se entiende como la atenuación en la potencia de la señal al propagarse por el espacio.

interferencia de múltiples señales LoRa, se realiza una sumatoria con la energía resultante de todos los solapamientos competentes.

Para finalizar el proceso, se obtiene la energía de la señal deseada y se calcula el SINR<sup>13</sup> (Signal-to-interference-plus-noise ratio) utilizando el ruido térmico y la energía interferente acumulada. Para obtener la energía del ruido térmico durante la transmisión de la señal deseada se utilizó la Ec. 15, donde  $T$  corresponde a la temperatura absoluta del medio en grados Kelvin (200° K),  $K$  hace referencia a la constante de Boltzman ( $1.38 \times 10^{-23} \text{ J/}^\circ\text{K}$ ) y  $AB$  es el ancho de banda de la señal deseada (1 MHz).

$$\mathbf{Ruido\ Térmico\ (dBm) = 10 * \log_{10}(1000 * K * T * AB)} \quad \mathbf{(Ec. 15)}$$

El valor obtenido pasa a ser evaluado mediante un condicional. En caso de que el SINR sea mayor a 16 dB, la señal deseada seguirá el proceso de desencapsulación respectivo. Caso contrario, si el SNIR es menor al umbral, será un indicativo de que la señal ha sido destruida por la interferencia, por consiguiente, se procederá a terminar la recepción de la señal llamando al método **NotifyRxEnd** y a descartar la señal con el método **NotifyRxDrop**, ambos pertenecientes a la clase **WifiPhy**.

El umbral de 16 dB ha sido escogido luego del análisis pertinente, partiendo de lo definido en la sección 23.3.17.4 del estándar IEEE 802.11ah (IEEE 802.11 Working Group, 2016), esta sección deja en claro que el receptor deberá proveer un PER (Packet Error Rate) máximo de 10% para que la señal pueda ser utilizada adecuadamente. Teniendo un PER de referencia es posible relacionarlo con el BER (Bit Error Rate), mediante la Ec. 16 se puede calcular el PER con el BER y la longitud del payload.

$$\mathbf{PER = 1 - (1 - BER)^L} \quad \mathbf{(Ec. 16)}$$

---

<sup>13</sup> El SNIR indica la relación señal-interferencia más ruido, sirve como un medidor de la calidad de la red, se obtiene en dB con la siguiente fórmula:  $10 * \log_{10}(\frac{Señal}{Ruido + Interferencia})$ .

De acuerdo con la Ec. 15, un BER de  $10^{-3}$  es necesario para que se cumpla con un PER de 10%, finalmente, se establece la relación entre BER y SNR en función del esquema de modulación, el esquema definido para la simulación de IEEE 802.11ah es el MCS 2, mismo que opera con modulación QPSK (ver Tabla 2). Con toda esta información obtenida se compararon entre distintos trabajos de investigación que evaluaran las curvas de SNR vs BER. De acuerdo con la Figura 62, el umbral de SNR debería ser de 10 dB para cumplir con un BER de  $10^{-3}$  trabajando con una modulación QPSK (Hervas, Alsina-Pagès, Pijoan, & Salvador, 2014).

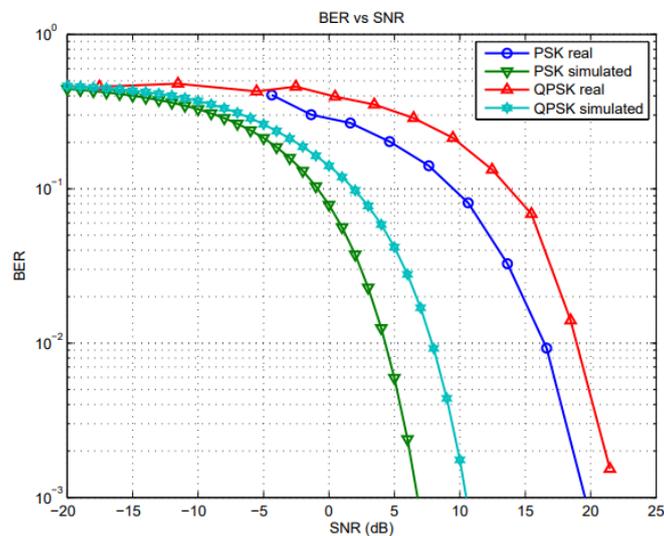


Figura 62. Relación entre BER y SNR.  
Fuente: (Hervas, Alsina-Pagès, Pijoan, & Salvador, 2014).

Por otro lado, un trabajo de investigación acerca de simulación de modelos de enlaces inalámbricos (Chen, 2010) propone las curvas de la Figura 63, los valores que toman las curvas con un BER  $10^{-3}$  son muy similares a los de la Figura 62.

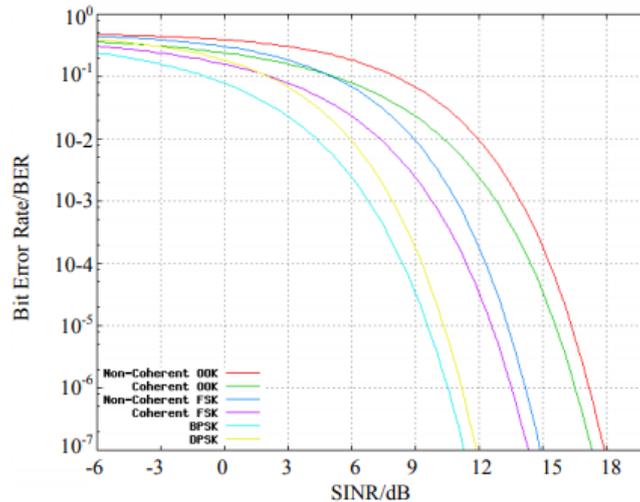


Figura 63. Relación en BER y SINR.  
Fuente: (Chen, 2010).

A pesar de que existe una relación clara entre el PER máximo determinado por el Estándar IEEE 802.11ah y el SINR, las curvas de las Figuras 62 y 63 pertenecen a tonos puros, por lo que es necesario tomar en cuenta cada una de las subportadoras que componen el símbolo OFDM utilizado por IEEE 802.11ah, tal y como lo hace NS-3. El simulador de red, a través de la clase denominada **SpectrumWifiPhy** implementa una mejora spectrum-aware (consciente del espectro de frecuencia) para las tecnologías 802.11. En la Figura 64 se muestra la relación de herencia entre la clase **WifiPhy** (modela la capa física para las tecnologías 802.11) y **OfdmaSpectrumWifiPhy** (permite probar OFMD).

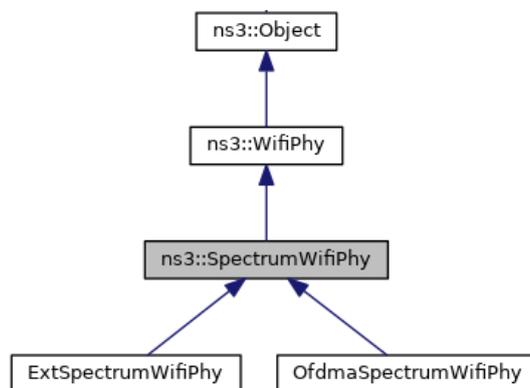


Figura 64. Diagrama de herencia para la clase SpectrumWifiPhy.  
Fuente: (NS-3, 2021).

Siendo necesario un análisis más profundo de la relación que existe entre el PER y el SNR, se ha optado por utilizar el paquete WLAN de Matlab®, dentro del cual consta un ejemplo para medir el PER de una comunicación IEEE 802.11ah en un enlace extremo a extremo con fading y ruido Gaussiano. La forma en que este código trabaja es transmitiendo múltiples paquetes para cada uno de los valores de SNR especificados, cada valor de SNR se llamará punto SNR y seguirá el siguiente proceso:

1. Se transmiten varios paquetes de datos a través de un canal TGah<sup>14</sup> 1x1 con AWGN<sup>15</sup>.
2. Cada paquete se sincroniza en el tiempo dado un retraso conocido.
3. Los campos S1G-LTF1 y S1G-LTF2N (ver Capítulo II) se demodulan y se realiza la estimación del canal.
4. El campo de datos S1G se extrae de la forma de onda recibida sincronizada y se demodula OFDM.
5. Las subportadoras portadoras de datos se ecualizan utilizando las estimaciones del canal.
6. La PSDU (PCLP Service Data Unit) se recupera utilizando las subportadoras de datos ecualizadas, la estimación de la varianza de ruido y la CSI (Channel State Information).
7. La PSDU recuperada de cada paquete se compara con las transmitidas para determinar el número de errores de paquetes y, por tanto, la tasa de errores de paquetes.

Los primeros resultados se obtuvieron con una configuración de parámetros basada en la Tabla 8, los puntos de SNR escogidos fueron entre 5 y 45 dB con espacios

---

<sup>14</sup> Hace referencia al Task Group que desarrolló el estándar IEEE 802.11ah.

<sup>15</sup> AWGN son las siglas en inglés para el Ruido Blanco Gaussiano Aditivo, utilizado para imitar el ruido que se puede encontrar en la naturaleza.

de 10 dB entre cada uno. Recordando la sección 23.3.17.4 del estándar IEEE 802.11ah (IEEE 802.11 Working Group, 2016), el valor máximo de PER será de 0.1 (1 es igual al 100%). De acuerdo con la Figura 65, un PER de 0.1 ( $10^{-1}$ ) se encontraría entre 15 y 20 dB.

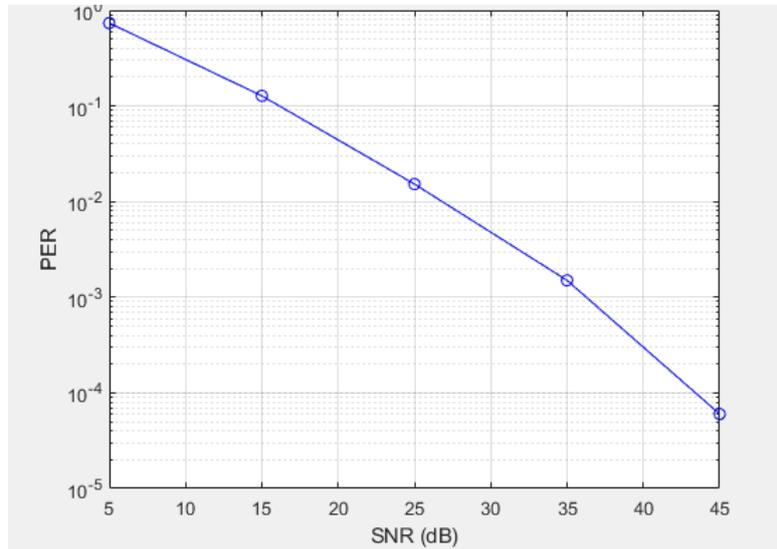


Figura 65. Simulación del PER para IEEE 802.11ah con SNR entre 5 y 45 dB.  
Fuente: Autor.

Conociendo un rango más preciso se vuelve a ejecutar la simulación del PER para IEEE 802.11ah en Matlab®, pero esta vez en un rango de 15 a 20 dB con espacios de 1 dB entre cada uno. Los resultados de la Figura 66 indican que un PER de 0.1 o del 10% se consigue con un SNR de 16 dB. Este valor es el escogido para servir como umbral en la simulación y determinar si alguna de las subportadoras de IEEE 802.11ah es destruida por la interferencia.

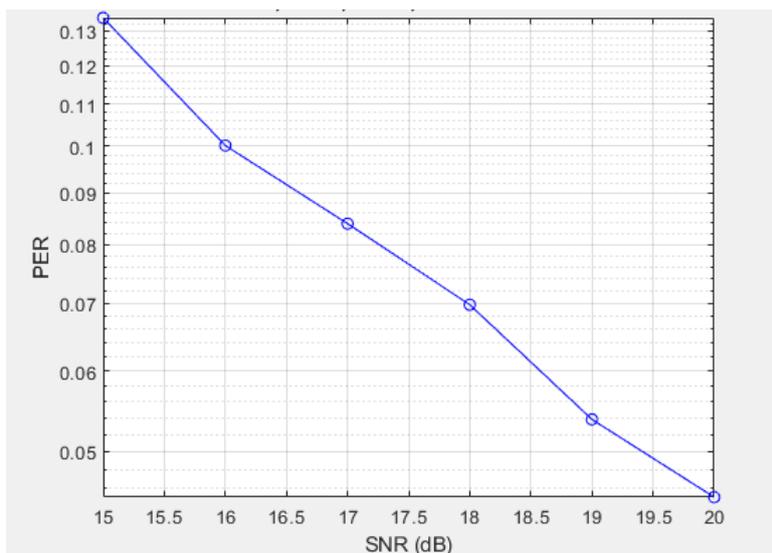


Figura 66. Simulación del PER para IEEE 802.11ah con SNR entre 15 y 20 dB.  
Fuente: Autor.

Una vez identificado el umbral adecuado se da paso al proceso final, que consta de calcular la potencia de recepción real de la señal interferente, así como la obtención del SINR (con todos sus componentes) y determinar si el PPDU sobrevive a la interferencia, se encuentran comprendidos en el Procedimiento 2.

---

**Procedimiento 2.** Obtener el SINR y determinar si pasa el umbral

---

**Input:** positionRx = posición “x” e “y” de la STA IEEE 802.11ah

**Input:** banderah = booleano que indica un solapamiento

**Input:** power = potencia de transmisión de la señal interferente

**Input** signalPowerW\_1 = potencia de transmisión de la señal deseada

**Input:** overallp\_1 = tiempo de solapamiento obtenido

**Input:** posición en “x” e “y” de la STA que origina la señal interferente

m\_referenceLoss = 30, m\_exponent = 3,

m\_referenceDistance = 1, m\_frequency = 900e6

1: **if** banderah == true

2: distancia = Ecuación de la distancia entre dos puntos (Pie de página 11)

3: pathLossDB = Path Loss obtenido de la Ec. 14

4: rxc = pathLossdB – m\_referenceLoss

5: rxPowerdBm\_lora = power + rxc

6: interfPowerW\_lora = rxPowerdBm\_lora transformado de dBm a Watts

```

7:  interfEnergy_lora = interfPowerW_lora * overalp_1
8:  cumulativeInterfEnergy_lora += interfEnergy_lora
9:  Extrae los resultados un archivo de texto plano
10: signalEnergy_1 = duration * signalPowerW_1
11: ruidoTermico = energía del ruido térmico obtenido de la Ec. 15
12: sinr_lora = (signalEnergy_1/ cumulativeInterfEnergy_lora + ruidoTermico)
13: if contBand != 0
14:  Extrae los resultados en un archivo de texto plano
15:    if sinr_lora < 10
16:      Termina la recepción actual
17:      Dropea el PPDU
18:      Cambia el estado de la STA de RxEnd a Error
19:    else
20:      Continúa el proceso de Recepción
21: return sinr

```

---

### 3.4 MODELO ÚNICO PARA CONTROLAR LAS ITERACIONES

Debido a la incompatibilidad entre los nodos de LoRa e IEEE 802.11ah dentro del simulador NS-3, se tomó la decisión de simular cada tecnología individualmente. Sin embargo, esto no quiere decir que las modificaciones realizadas a ambos módulos no hayan sido con el fin de contar con un solo proceso unificado. En otras palabras, primero se ejecuta la simulación de LoRa, luego los datos obtenidos producto de la iteración son utilizados por la simulación de IEEE 802.11ah para contabilizar una interferencia inter-protocolo.

La simulación de la tecnología IEEE 802.11ah se realiza de acuerdo con los parámetros de la Tabla 8, cabe mencionar que no todos estos valores se mantendrán estáticos a excepción de aquellos que son solicitados como input por el ejecutable. Bajo los parámetros PHY se encuentra especificado el modelo de propagación “macro urbano”, mismo que se encuentra explicado más a detalle en el Capítulo II. Por otro lado, en los

parámetros MAC se encuentran expresadas las duraciones de los elementos que componen el PPDU. Es importante mencionar que, considerando el MCS 2 escogido, la longitud del Payload de 100 bytes seleccionada para las simulaciones, no es el máximo disponible (ampliado en la Sección 4.2.1).

Tabla 8. Parámetros esenciales de la simulación de IEEE 802.11ah.

<b>Parámetros PHY</b>	<b>Valor</b>
Potencia TX (dBm)	0
Ganancia TX/RX (dB)	0
Noise Figure (dB)	6.8
Método de codificación	BCC
Modulación	QPSK
Modelo de Propagación	Macro urbano
Error rate model	YansErrorRate
<b>Parámetros MAC</b>	<b>Valor</b>
Duración del DIFS ( $\mu$ s)	264
Duración del SIFS ( $\mu$ s)	160
Duración Preámbulo PPDU ( $\mu$ s)	160, 320
Duración Header PPDU ( $\mu$ s)	80, 240
Intervalo de beacon ( $\mu$ s)	102400
Tamaño de la cola Tx (paquetes)	10
Intervalo de transmisión (s)	1
Distribución de las STAs	Aleatoria
<b>Parámetros de rendimiento</b>	<b>Valor</b>
Modo Wi-Fi	MCS 2
BW (MHz)	2
PER máximo (%)	10
Longitud del Payload (bytes)	100
Radio de la topología (m)	200
Número de nodos	50 - 300

En el caso de la simulación de LoRa se cuenta de igual manera con un set de parámetros específicos. En la Tabla 9, bajo los parámetros MAC se encuentra la banda y

sub-bandas de operación, mismas que fueron agregadas al modificar el módulo de LoRa en el Capítulo III. Bajo los parámetros de rendimiento, aparte del ancho de banda estático se especifican los rangos de valores que puede tomar el SF, el Payload, el radio o el número de nodos.

Tabla 9. Parámetros esenciales de la simulación de LoRa.

<b>Parámetros PHY</b>	<b>Valor</b>
Potencia TX (dBm)	14
Ganancia TX/RX (dB)	0
Noise Figure (dB)	6.8
Modulación	LoRa
Banda de Operación (MHz)	902 - 928
Sub-bandas (MHz)	902.2 - 903.7 903.9 – 905.3 905.5 – 906.9
Modelo de Propagación	Log Distance
<b>Parámetros MAC</b>	<b>Valor</b>
Longitud preámbulo (bytes)	10
Intervalo de transmisión (s)	1
Distribución de las STAs	Aleatoria
<b>Parámetros de rendimiento</b>	<b>Valor</b>
SF	7 - 12
BW (kHz)	125
Longitud del Payload (bytes)	10 - 30
Radio de la topología (m)	200 - 800
Número de nodos	10 - 8000

Para ejecutar todo el proceso anteriormente descrito de una manera automatizada, se generó un archivo de tipo ejecutable (ver Figura 67) en Ubuntu, mismo que permite ingresar los parámetros de cada iteración (IEEE 802.11ah y LoRa), ejecuta en orden ambas simulaciones, para finalmente recoger y organizar los datos obtenidos en un fichero debidamente identificado.

```

1 #!/bin/bash
2 # SCRIPT PARA AUTOMATIZAR LA SIMULACIÓN
3
4 #####
5 # Receptar parámetros por teclado.
6 cd ~
7 echo "Bienvenido al asistente de Simulación"
8
9 echo "Por favor ingrese los parámetros solicitados para IEEE 802.11ah: "
10 echo -e "Número de nodos: "
11 read nodos_ah
12 echo -e "Radio en metros: "
13 read radio_ah
14 echo -e "Tamaño del Payload en bytes: "
15 read payload_ah
16 echo -e "Tiempo en segundos: "
17 read tiempo_ah

```

Figura 67. Ejecutable para controlar las iteraciones.  
Fuente: Autor.

Tal como se observa en la Figura 68, lo primero que se solicita al utilizar el ejecutable son: el número de nodos, el radio de distribución de los nodos (el proceso de asignar una posición a cada nodo es aleatoria), la longitud del Payload y el tiempo límite que durará la simulación. Los parámetros mencionados se ingresan por teclado de manera individual para cada tecnología, ver Figura 69.

```

mateo@mateo:~$ sudo ./mainScript.sh
[sudo] contraseña para mateo:
Bienvenido al asistente de Simulación
Por favor ingrese los parámetros solicitados para IEEE 802.11ah:
Número de nodos:
100
Radio en metros:
200
Tamaño del Payload en bytes:
100
Tiempo en segundos:
30
Cargando...

Por favor ingrese los parámetros solicitados para LoRa:
Número de nodos:
200
Radio en metros:
200
Tiempo en segundos:
180
Configurando parámetros...
./mainScript.sh: línea 44: ig-test.txt: orden no encontrada
Waf: Entering directory `/home/mateo/LORAWAN/ns-3/build'

```

Figura 68. Demostración del ejecutable.  
Fuente: Autor.

Finalizado el proceso, se obtiene como resultado un archivo en texto plano con datos claves para su posterior análisis, cada iteración generará un archivo con nombre diferente, representado por los parámetros utilizados.

```

*****
Simulación de IEEE 802.11ah terminada correctamente
Sus resultados han sido guardados en la ruta /home/mateo/Iteraciones con el siguiente nombre:
AH_nodos100-radio200-payload100-tiempo30-LORAnodos200-radio200-tiempo180.txt
mateo@mateo:~$

```

Figura 69. Fichero resultado de la iteración.  
Fuente: Autor.

### 3.5 RECOLECCIÓN DE DATOS

Antes de realizar las iteraciones se definió primero las configuraciones de cada simulación (LoRa e IEEE 802.11ah), de manera que en cada iteración se varíe un determinado parámetro o grupo de parámetros. En este sentido se construyó la Tabla 10, misma que corresponde a la simulación de IEEE 802.11ah y contiene variaciones del parámetro “nodos”, mientras el resto de los parámetros se mantiene estático.

El motivo por el cual se ha escogido iterar en relación con la cantidad de nodos se encuentra fundamentado en los objetivos del presente trabajo, ya que se busca analizar el comportamiento de una red densa, es decir, una red compuesta por un masivo número de estaciones o nodos.

Tabla 10. Variaciones de la simulación de IEEE 802.11ah, en base al número de nodos.

<b>IEEE 802.11AH</b>				
<b>ID</b>	<b>Nodos</b>	<b>Radio (m)</b>	<b>Payload (bytes)</b>	<b>Tiempo (s)</b>
1	50	200	100	180
2	100	200	100	180
3	150	200	100	180
4	200	200	100	180

Fuente: Autor.

Tal y como fue explicado en las secciones anteriores, la simulación de IEEE 802.11ah necesita de datos extraídos de la simulación de LoRa, de modo que para cada una de las iteraciones de la simulación de IEEE 802.11ah se ha planteado también un grupo de variaciones para los parámetros de la simulación de LoRa.

De esta manera se ha estructurado la Tabla 8, en la cual constan 10 valores distintos en la cantidad de nodos, a la par que se conserva el mismo radio y tiempo de simulación que se platean en la Tabla 11.

Cabe mencionar que el tamaño del Payload no ha sido tomado en cuenta en la Tabla 8 debido al propio funcionamiento del módulo de LoRa, puesto que la longitud de la carga útil en cada transmisión es seleccionada de manera aleatoria entre 10 y 30 bytes.

Tabla 11. Iteraciones de la simulación de LoRa, en base al número de nodos.

<b>LORA</b>			
<b>ID</b>	<b>Nodos</b>	<b>Radio (m)</b>	<b>Tiempo (s)</b>
1	10	200	180
2	100	200	180
3	300	200	180
4	500	200	180
5	700	200	180
6	1000	200	180
7	2000	200	180
8	4000	200	180
9	6000	200	180
10	8000	200	180

Fuente: Autor.

Lo que se busca es utilizar de la mano la Tabla 12 y la Tabla 13, de modo que cada configuración de IEEE 802.11ah (4) se itere tantas veces como variaciones de LoRa (10) se han propuesto, lo que da como resultado un total de 40 simulaciones distintas.

Para llevar un registro y verificar que cada iteración se haya realizado como lo planeado se diseñó la Tabla 12, en donde, el número de registro corresponde al número de variación actual (1-4), la columna 11AH hace referencia a la configuración escogida de IEEE 802.11ah (ver Tabla 7) y lo propio con la columna LORA (ver Tabla 8).

Tabla 12. Registro de combinaciones realizadas.

<b>Registro 11AH LORA</b>		
1	1	1
	1	2
	1	3
	1	...
	1	10

Fuente: Autor.

Otro de los parámetros clave que se tomó en cuenta para la iteración de las simulaciones es el radio, el valor que se asigne a cualquiera de las simulaciones determinará el radio máximo de distribución (aleatoria) de los nodos, es decir, hasta que distancia se ubicarán los nodos dentro de la simulación. Se escogió iterar el radio de la

simulación de LoRa entre 50 y 4000 metros, mientras se mantiene el tiempo de simulación (180s) y la cantidad de nodos (1000), tal como se puede observar en la Tabla 13.

Tabla 13. Iteraciones de la simulación de LoRa, en base al radio de distribución de los nodos.

<b>LORA</b>			
<b>ID</b>	<b>Nodos</b>	<b>Radio (m)</b>	<b>Tiempo (s)</b>
1	1000	50	180
2	1000	100	180
3	1000	200	180
4	1000	400	180
5	1000	600	180
6	1000	800	180
7	1000	1000	180
8	1000	2000	180
9	1000	3000	180
10	1000	4000	180

Fuente: Autor.

Esta última variación de parámetros busca comprobar principalmente la relación entre el SINR y la distancia entre los nodos LoRa e IEEE 802.11ah, tomando en consideración que las fórmulas utilizadas para calcular el SINR utilizan la distancia entre estaciones.

#### **4. CAPÍTULO IV: RESULTADOS**

Este capítulo cubre las simulaciones realizadas con los módulos modificados de IEEE 802.11ah y LoRa, descritos a detalle en el Capítulo 3 y analiza los resultados obtenidos de los mismos. Las métricas escogidas para evaluar el desempeño de la red IEEE 802.11ah frente a la interferencia proveniente de LoRa son: throughput, total de paquetes perdidos y SINR (Signal-to-Interference-plus-Noise Ratio), mismas que son consideradas como Key Performance Indicator (KPI) dentro del estándar de calidad ANSI C63.27-2017 American National Standard for Evaluation of Wireless Coexistence. En esta norma se proporciona un proceso de evaluación y métodos de prueba de apoyo, para cuantificar la capacidad de un dispositivo inalámbrico para coexistir con otros servicios inalámbricos.

Además, con la finalidad de comprender a detalle el fenómeno de la interferencia de LoRa sobre IEEE 802.11ah se ha dedicado un espacio para el análisis teórico-matemático de este, a través del cálculo de la duración del preámbulo, cabecera y Payload del PPDU de capa física.

##### **4.1 COMPROBACIÓN TEÓRICO-MATEMÁTICA**

Para comprobar el comportamiento de las señales IEEE 802.11ah y LoRa dentro de un ambiente donde una interfiera con la otra se ha seleccionado una iteración simulación de 100 nodos de IEEE 802.11ah y 10 nodos de LoRa, con el objetivo de extraer los tiempos de cada elemento de la PPDU (PLCP Protocol Data Unit) y determinar en qué sección del PPDU se ha producido la interferencia (preámbulo, cabecera, datos). Adicionalmente, se comprueba de manera analítica la potencia espectral que ocupa cada uno a través del PSD (Power Spectral Density) de ambas señales.

Desde la simulación de IEEE 802.11ah se han extraído datos relacionados con las duraciones de los campos que componen el formato del PHY PPDU (ver Figura 5), se ha podido identificar dos tipos de configuraciones con tiempos distintos. El primero de este

conjunto de datos extraídos se encuentran en la Figura 70, corresponde a un tipo de preámbulo corto (ver Figura 5), con una duración del preámbulo de 160  $\mu$ s, un tiempo de inicio de la cabecera PLCP determinado y el tiempo de inicio del Payload (la duración del resto de campos es cero).

```

PayloadMode: OfdmRate650KbpsBW2MHzPreamble: 0
DuracionPreamble: +160000.0ns
PLCPHeaderStart: +2542568456.0ns
PCLPHtTrainingSymbolStart (Ts):+2542648456.0ns
PLCPHtTrainingSymbolDuraion (Ts): +0.0ns
PLCPSlgAStart: +2542648456.0ns
PLCPSlgADuration: +0.0ns
PLCPSlgTrainingSymbolsStart: +2542648456.0ns
PLCPSlgTrainingSymbolDuration : +0.0ns
PLCPSigBStart: +2542648456.0ns
PLCPSigBDuration: +0.0ns
PLCPPayloadStart: +2542648456.0ns

```

Figura 70. Conjunto de tiempos de duración del PDU para un preámbulo SHORT.  
Fuente: Autor.

Con los datos del tiempo de inicio del Header y el tiempo de inicio del Payload se obtiene la duración del Header, que es de 80  $\mu$ s, luego de reemplazar los valores respectivos en la Ec. 18. Sumando las duraciones del preámbulo y la cabecera se obtiene un total de 240  $\mu$ s, esta información será útil más adelante.

$$\text{duraciónHeader} = \text{tiempoInicioPayload} - \text{tiempoInicioHeader} \quad (\text{Ec. 18})$$

$$\text{duraciónHeader} = 2542648.456 \mu\text{s} - 2542568.456 \mu\text{s} = 80 \mu\text{s}$$

De los datos extraídos de la simulación de IEEE 802.11ah se identificó el formato SIG\_1M, en este sentido la duración del preámbulo es de 320  $\mu$ s (ver Figura 71).

```

PayloadMode: OfdmRate300KbpsBW1MHzPreamble: 7
DuracionPreamble: +320000.0ns
PLCPHeaderStart: +2544488014.0ns
PCLPHtTrainingSymbolStart (Ts):+2544728014.0ns
PLCPHtTrainingSymbolDuraion (Ts): +0.0ns
PLCPSlgAStart: +2544728014.0ns
PLCPSlgADuration: +0.0ns
PLCPSlgTrainingSymbolsStart: +2544728014.0ns
PLCPSlgTrainingSymbolDuration : +0.0ns
PLCPSigBStart: +2544728014.0ns
PLCPSigBDuration: +0.0ns
PLCPPayloadStart: +2544728014.0ns

```

Figura 71. Conjunto de tiempos de duración del PDU para un preámbulo SIG\_1M.  
Fuente: Autor.

Por otro lado, la duración de la cabecera del PPDU se puede calcular mediante la Ec. 18, obteniendo que entre todos los campos de la cabecera se tiene una duración de 240  $\mu$ s.

$$duraciónHeader = 2544728.014 \mu s - 2544488.014 \mu s = 240 \mu s$$

De los datos generados por la simulación de IEEE 802.11ah se ha seleccionado de manera aleatoria uno de los eventos de interferencia. En la Tabla 14 se recopilan los tiempos de inicio y de fin de los PDDU de IEEE 802.11ah y LoRa para el evento escogido, así como también la duración total de cada PPDU, obtenido a partir del cálculo de los tiempos ya mencionados.

Tabla 14. Caso de estudio 1, transmisión de Lora empieza antes de IEEE 802.11ah y termina antes de IEEE 802.11ah.

	<b>Tiempo Inicio (<math>\mu</math>s)</b>	<b>Tiempo Fin (<math>\mu</math>s)</b>	<b>Duración (<math>\mu</math>s)</b>
<b>IEEE 802.11ah</b>	39282802.170	39285122.170	2320
<b>LoRa</b>	39212700.000	39284600.000	71900

Fuente: Autor.

Para analizar más a profundidad este evento de interferencia es necesario mencionar que el formato del preámbulo del PPDU para IEEE 802.11ah corresponde a SIG\_1M (duración de preámbulo 320  $\mu$ s y duración de cabecera 240  $\mu$ s), de modo que al comparar los tiempos que constan en la Tabla 14, se ha determinado que corresponde al caso de solapamiento 6 (ver Figura 61). En este caso específico de solapamiento (ver Figura 72), la señal de LoRa interfiere a la señal de IEEE 802.11ah en una duración total de 522.170  $\mu$ s, afectado completamente el preámbulo y parte de la cabecera del PPDU. Al haber sido afectado el preámbulo, se compromete la capacidad de sincronizar el transmisor con el receptor durante la comunicación.

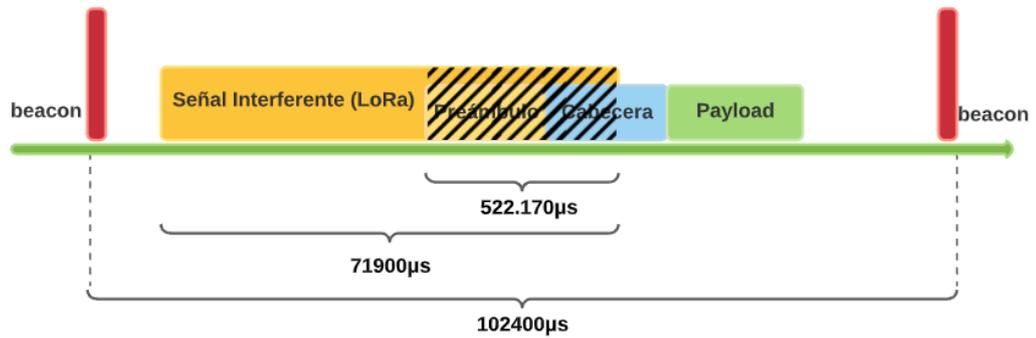


Figura 72. Suma de los tiempos de cabecera y preámbulo del PDU.  
Fuente: Autor.

Para contar con una referencia, se ha tomado otro evento dentro de los mismos datos obtenidos de la simulación de IEEE 802.11ah. Los tiempos de inicio, fin y duración, tanto de la señal interferente como de la señal deseada se encuentran organizados en la Tabla 15.

Tabla 15. Caso de estudio 2, transmisión de Lora empieza antes de IEEE 802.11ah y termina antes de IEEE 802.11ah.

	<b>Tiempo Inicio (µs)</b>	<b>Tiempo Fin (µs)</b>	<b>Duración (µs)</b>
<b>IEEE 802.11ah</b>	39270170.687	39271170.687	2320
<b>LoRa</b>	39212700.000	39284600.000	71900

Fuente: Autor.

Utilizando los datos de la Tabla 15, se identificó que el evento en cuestión correspondería al caso de solapamiento 5 (ver Figura 60), donde la señal deseada es transmitida durante la señal interferente, es decir, la PDU de IEEE 802.11ah se ve completamente afectada por la interferencia de LoRa (ver Figura 73) en la totalidad de sus campos (preámbulo, cabecera y Payload).

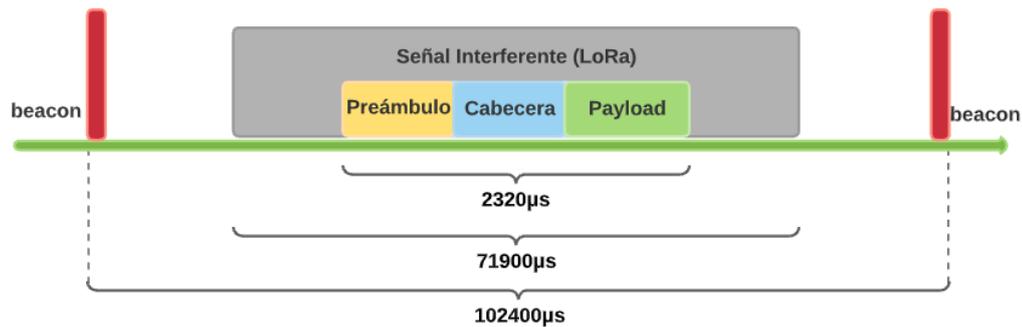


Figura 73. Suma de los tiempos de cabecera y preámbulo del PDU.  
Fuente: Autor.

Debido a la utilización de OFDM por parte de la tecnología IEEE 802.11ah, es necesario realizar un análisis adicional, ya que, si bien se ha determinado la manera en la que la señal de LoRa puede interferir a la señal de IEEE 802.11ah, no ha sido tomado en cuenta (por el momento) el comportamiento de las subportadoras de OFDM. Con el fin de analizar que sucede durante la interferencia de estas tecnologías en el dominio de la frecuencia, se ha utilizado una gráfica del PSD (Power Spectral Density).

Antes de analizar el PSD de la interferencia de ambas tecnologías, se debe obtener las IQ samples (muestras en fase y cuadratura) de la LoRa y de IEEE 802.11ah. Para el caso de LoRa se ha utilizado un código disponible en el File Exchange<sup>16</sup> de Matlab®, dicho código permite emular la transmisión y recepción de LoRa (Al Homssi, Dakic, & Maselli, 2021). Los parámetros utilizados en la generación de los IQ samples de LoRa corresponden a los definidos en la Tabla 9. Una vez obtenidas las muestras se utilizó la función **obw** de Matlab®, para graficar en el dominio de la frecuencia el 99% del ancho de banda ocupado por la señal. El resultado es la Figura 74, donde se puede identificar que la distribución de la potencia de la señal de LoRa se concentra en un ancho de banda aproximado de 130.484 kHz.

<sup>16</sup> File Exchange es un foro/comunidad que permite a los usuarios de Matlab® obtener e intercambiar código open-source.

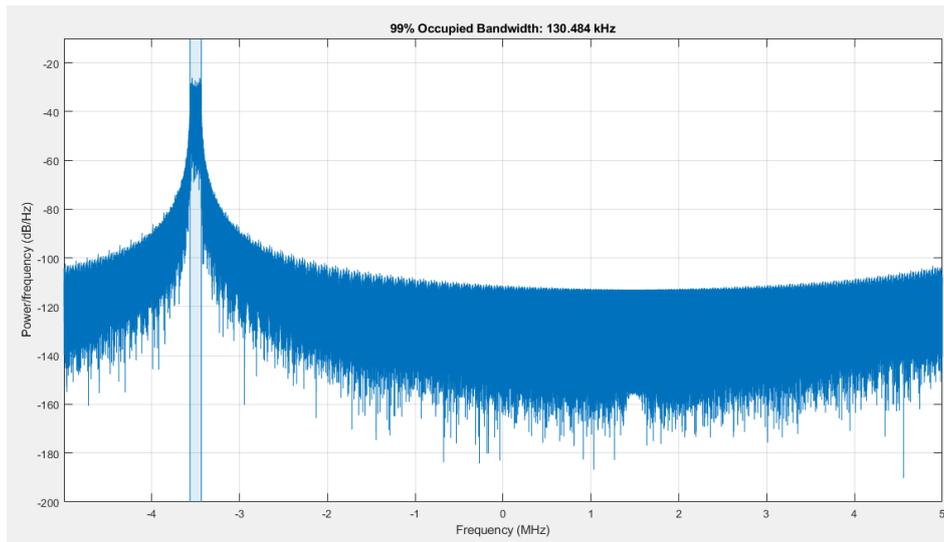


Figura 74. PSD de la señal de LoRa.  
Fuente: Autor.

Para generar las muestras de la señal de IEEE 802.11ah se utilizó la herramienta Matlab® WLAN Toolbox, esta utilidad de Matlab® permite generar señales de la familia de estándares IEEE 802.11 be/az/ax/ac/ad/ah/j/p/n/g/a/b. Los parámetros ingresados en la herramienta corresponden a los de la Tabla 8, de manera que la forma de onda generada sea lo más apegada a la simulación posible. El resultado se aprecia en la Figura 75, donde se puede identificar que las 52 subportadoras de un símbolo OFDM ocupan 1.758 MHz de ancho de banda, la parte restante a cada lado corresponde a las bandas de guarda.

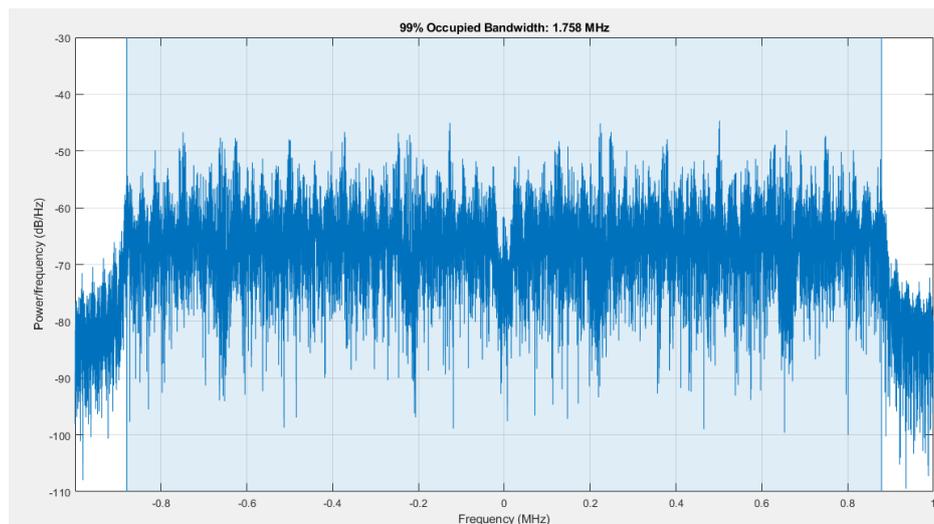


Figura 75. PSD de la señal IEEE 802.11ah.  
Fuente: Autor.

Con ambas señales generadas se ha utilizado la función **periodogram**, que como su nombre lo indica, realiza una estimación de la densidad espectral de una señal  $x_n$  muestreada a una frecuencia  $f_s$ . El periodograma está definido según la Ec. 19, donde  $\Delta t$  es el intervalo de muestreo (Fulop & Fitz, 2006).

$$\hat{P}(f) = \frac{\Delta t}{N} \left| \sum_{n=0}^{N-1} x_n e^{-j2\pi f \Delta t n} \right|^2, \quad -1/2\Delta t < f \leq 1/2\Delta t, \quad (\text{Ec. 19})$$

En la Figura 76 se encuentran graficados la PSD de IEEE 802.11ah (color azul) y LoRa (color rojo), la interferencia entre ambas tecnologías se puede identificar únicamente alrededor de -0.7 MHz. Tomando en cuenta el ancho de banda de LoRa (ver Figura 74) y el espacio entre cada subportadora para IEEE 802.11ah (31.25 kHz), es posible asumir que la interferencia afectaría como máximo a 4 subportadoras, entre estas subportadoras afectadas podría encontrarse al menos una de tipo piloto (ayuda a la sincronización de portadora), esto dependería de la frecuencia del sub-canal de LoRa utilizado. En el resto de frecuencias existe una clara superioridad de IEEE 802.11ah por sobre LoRa en cuanto a la densidad de potencia (dB/Hz).

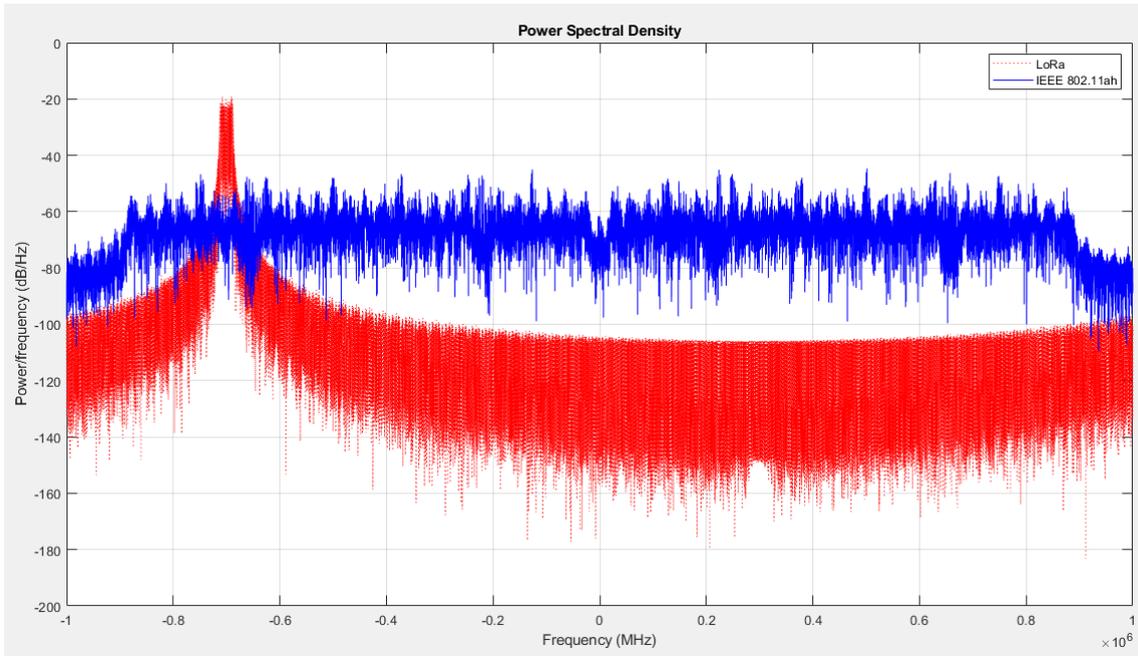


Figura 76. PSD de la interferencia entre LoRa e IEEE 802.11ah.  
Fuente: Autor.

## 4.2 EVALUACIÓN DE LAS MÉTRICAS

Para evaluar óptimamente los datos recogidos se ha desarrollado una interfaz gráfica en Matlab. La GUI de la Figura 77 se encuentra dividida en tres secciones, la primera permite observar la distribución de los nodos de ambas tecnologías para cada una de las iteraciones y variaciones realizadas; la segunda muestra los parámetros de configuración de la variación-iteración escogida, así como los resultados principales de la simulación seleccionada; la sección tercera contiene las gráficas combinadas de las métricas mencionadas anteriormente.

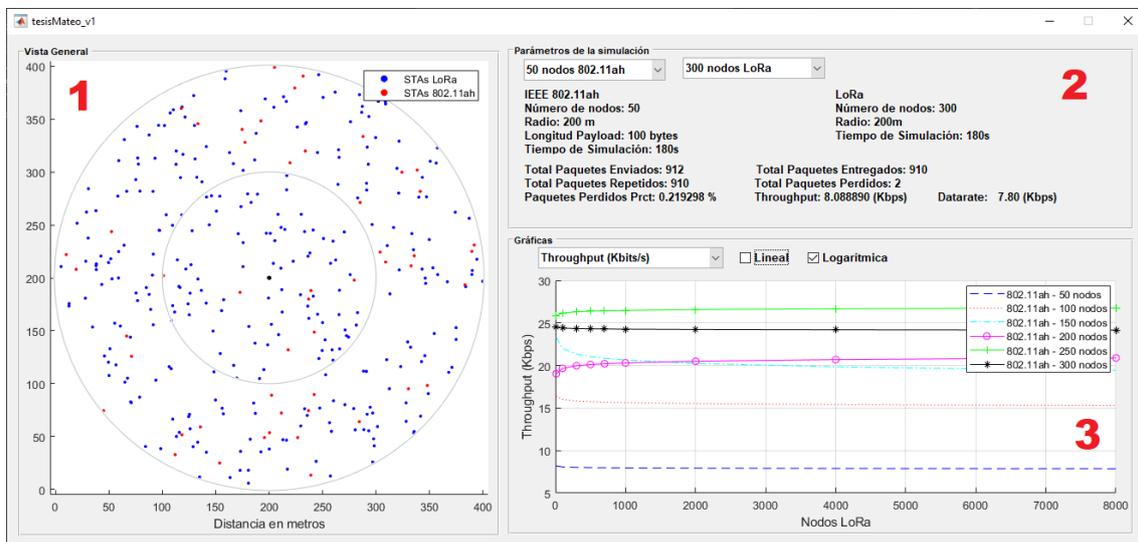


Figura 77. GUI en Matlab para evaluar métricas.  
Fuente: Autor.

Adicionalmente, dentro de la sección tercera se encuentra implementada la opción de visualizar las gráficas de las métricas a través de un ajuste logarítmico (Seber & Wild, 1989), con la finalidad de apreciar más fácilmente la tendencia que tienen los datos recolectados. La curva logarítmica mostrada se genera mediante la Ec. 20, donde  $a$  y  $b$  son los coeficientes obtenidos a partir de las Ec. 21 y Ec. 22 respectivamente.

$$Y = a * \ln(x) + b \quad (\text{Ec. 20})$$

$$a = \frac{\sum \ln(x)y - \bar{y} \sum \ln(x)}{\sum (\ln(x))^2 - \bar{\ln(x)} \sum \ln(x)} \quad (\text{Ec. 21})$$

$$b = \bar{y} - (a \bar{\ln(x)}) \quad (\text{Ec. 22})$$

Con todas las consideraciones antes mencionadas y siguiendo los parámetros de las Tablas 8 y 9 se realizaron 60 simulaciones, 10 por cada cantidad definida de estaciones de IEEE 802.11ah. De estas primeras iteraciones se generaron gráficas lineales y logarítmicas para cada una de las métricas (throughput, paquetes perdidos y SINR). Los primeros resultados que se obtuvieron se analizaron a detalle en busca de algún comportamiento o tendencia atípica.

En el throughput graficado linealmente en la Figura 78 se pudo detectar un patrón de comportamiento para las simulaciones de 50 y 100 nodos de IEEE 802.11ah, identificando una ligera tendencia a la disminución del throughput conforme aumenta el número de estaciones de LoRa. Sin embargo, el resto de las variaciones de nodos de IEEE 802.11ah no exhibe una correlación entre sí.

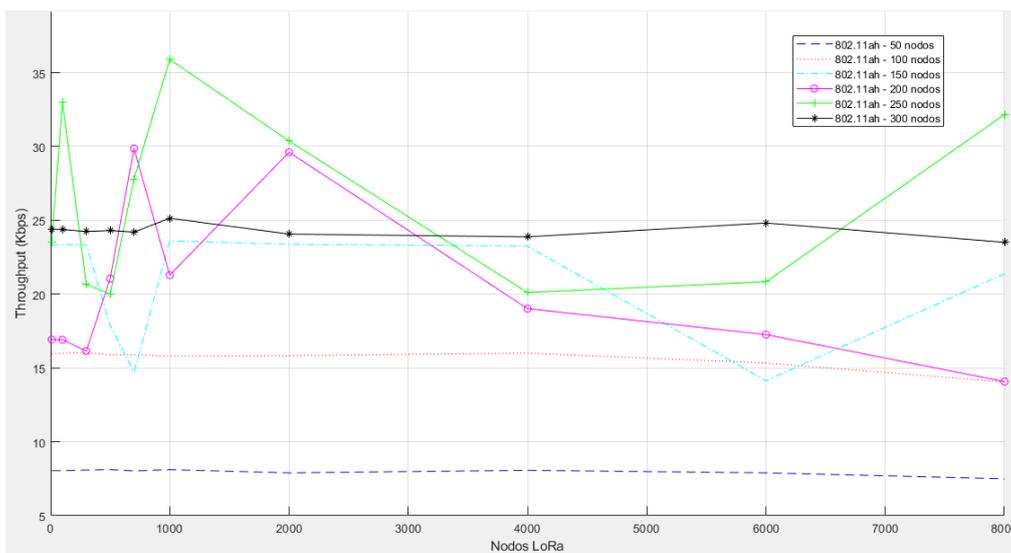


Figura 78. Gráfica lineal del throughput para los primeros resultados obtenidos.  
Fuente: Autor.

Una conducta similar se encontró en la gráfica logarítmica de la Figura 79, donde la tendencia de las curvas varía entre creciente y decreciente para todas las variaciones de nodos de IEEE 802.11ah.

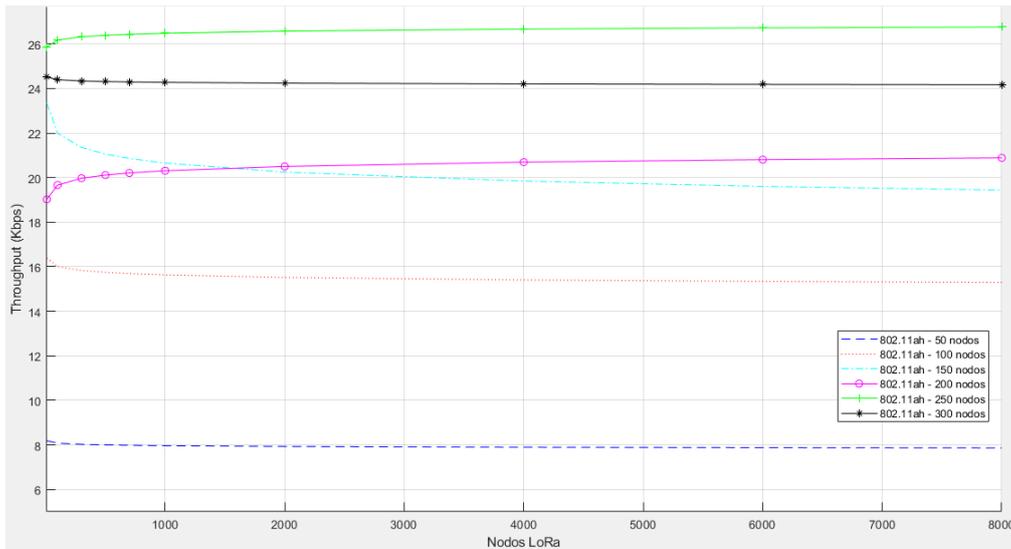


Figura 79. Gráfica logarítmica del throughput para los primeros resultados obtenidos.  
Fuente: Autor.

Para confirmar o descartar que se tratase de un caso aislado en los resultados obtenidos se analizó también la Figura 80. Esta gráfica reúne todos los datos de los paquetes perdidos y al igual que en la gráfica del throughput, los resultados para 50 y 100 nodos de IEEE 802.11ah presentan similitudes en sus rectas, con un aumento claro en el porcentaje de paquetes perdidos relacionado directamente con el aumento de nodos LoRa. En cambio los demás resultados (150, 200, 250 y 300 nodos de IEEE 802.11ah) se comportan de manera errática y de nueva cuenta, sin correlación aparente.

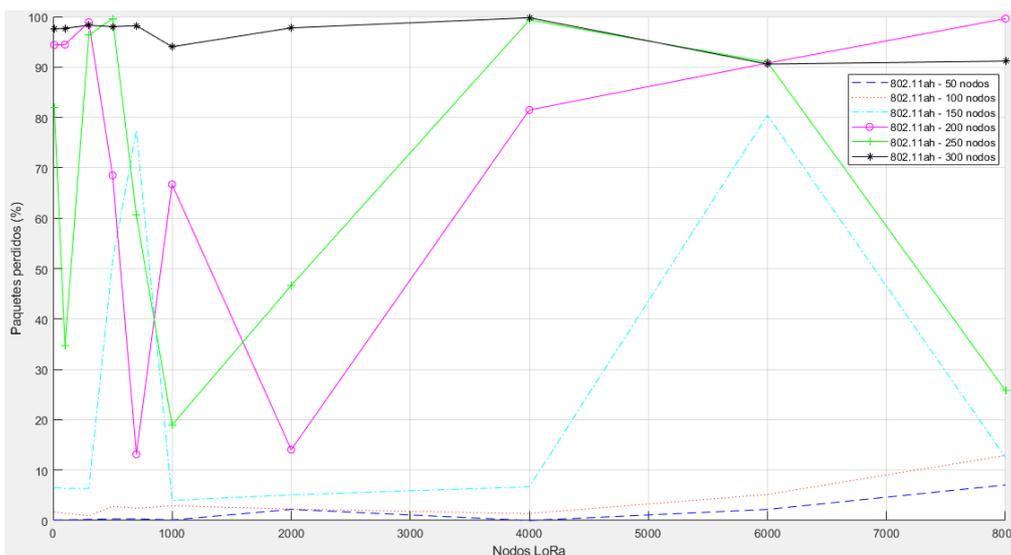


Figura 80. Gráfica lineal del porcentaje de paquetes perdidos para los primeros resultados obtenidos.  
Fuente: Autor.

La curva ajustada logarítmicamente de la Figura 81, también mezcla los dos tipos de curvas, creciente y decreciente.

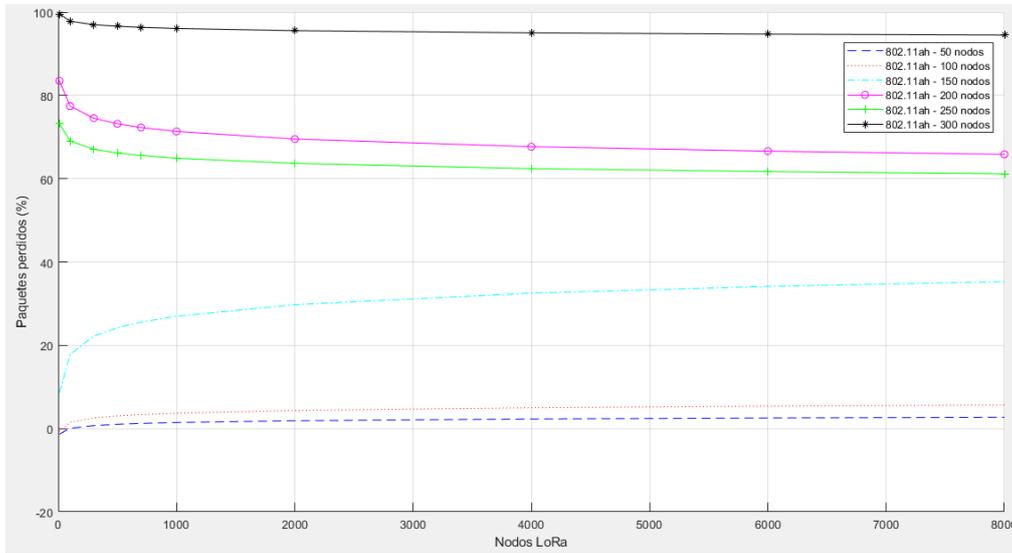


Figura 81. Gráfica logarítmica del porcentaje de paquetes perdidos para los primeros resultados obtenidos. Fuente: Autor.

Finalmente, para terminar con la comprobación sistemática de los datos obtenidos en esta primera instancia, se revisó la Signal-to-Interference + Noise Ratio. Si bien la Figura 82 parece contener resultados más correlacionados, al revisar a fondo se identifican incongruencias, como el valor mínimo registrado correspondiendo a las iteraciones de 50 nodos de IEEE 802.11ah.

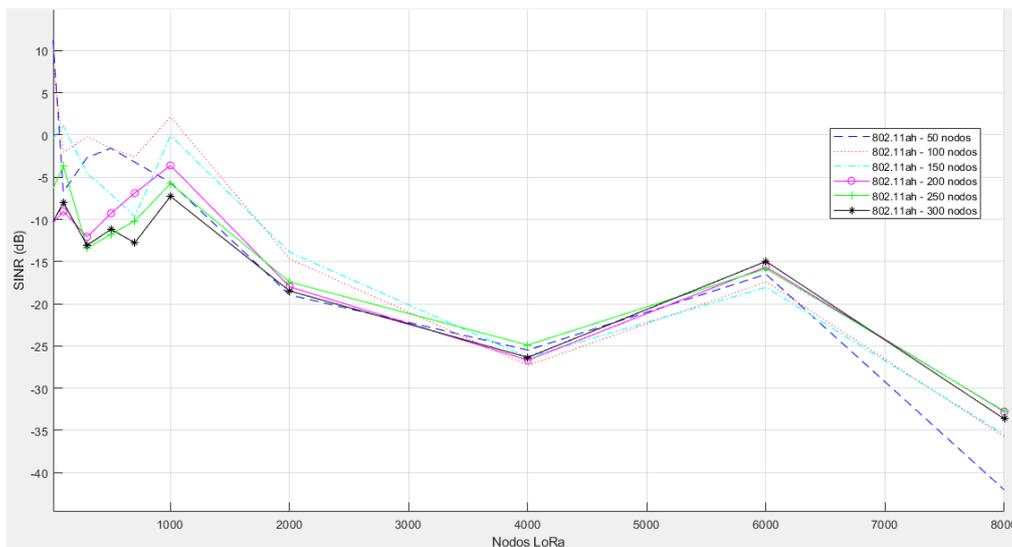


Figura 82. Gráfica lineal del SINR para los primeros resultados obtenidos. Fuente: Autor.

A pesar de que para esta métrica todas las curvas logarítmicas de la Figura 83 son del mismo tipo (decreciente), no es suficiente para dar por sentado estos resultados como válidos completamente. Será necesaria una revisión de las dependencias del módulo de IEEE 802.11ah para identificar la causa de este comportamiento.

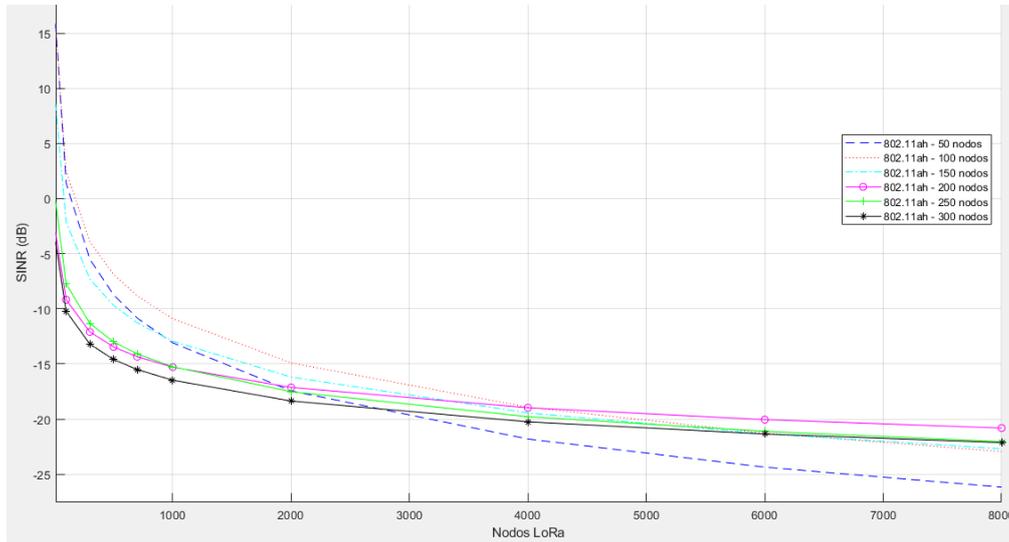


Figura 83. Gráfica logarítmica del SINR para los primeros resultados obtenidos.  
Fuente: Autor.

Al buscar una explicación para el comportamiento analizado, se simuló una vez más por cada variación de nodos de IEEE 802.11ah pero sin contabilizar la interferencia por LoRa, los datos que se obtuvieron se encuentran la Tabla 16.

El porcentaje de paquetes perdidos correspondiente a 50 y 100 nodos de IEEE 802.11ah es sumamente bajo comparado con los obtenidos en el resto de las simulaciones, esto explicaría porque en las gráficas de las Figuras 78 y 80 los resultados con más correlación y menos datos atípicos eran también los de 50 y 100 nodos (IEEE 802.11ah). El tener un valor tan alto de paquetes perdidos por defecto en la simulación es inaceptable y su revisión tardía será considerada como un error durante el desarrollo del proyecto.

En lo que respecta al throughput se puede notar un incremento hasta los 150 nodos de IEEE 802.11ah para luego pasara a una drástica disminución, que está relacionada directamente con el alto porcentaje de paquetes perdidos para estas simulaciones.

Tabla 16. Throughput y porcentaje de paquetes perdidos sin interferencia de LoRa.

<b>Nodos IEEE 802.11ah</b>	<b>Throughput (Kbps)</b>	<b>Paquetes perdidos (%)</b>
50	8.08444	0.219539
100	16.0622	0.441745
150	23.3378	6.48012
200	16.9244	94.3689
250	32.8578	35.3596
300	31.6489	99.7326

Fuente: Autor.

Después de analizar las modificaciones realizadas en el módulo de IEEE 802.11ah se logró identificar que los resultados que constan en la Tabla 16 se deben a la configuración de los parámetros de RAW utilizados. Estos parámetros se pueden encontrar en el archivo RAWConfigFile y siguen la estructura de la Tabla 17.

Tabla 17. Estructura del archivo RAWConfigFile.

RPS								
NRAWGroup								
RawControl	CrossSlotBoundary	SlotFormat	NRawSlotCount	NRawSlotNum	Page	Aid_start	Aid_end	

Fuente: Autor.

Donde cada uno de los campos dentro de la Tabla 17 cumple con las siguientes funciones:

- **RawControl:** Determina si cualquier estación dentro del grupo RAW puede acceder a RAW o solo las estaciones paginadas dentro del grupo RAW.
- **CrossSlotBoundary:** Indica si las STA pueden transmitir después del límite de ranura RAW asignado (1 sí, 0 no).
- **SlotFormat:** Formato de los Slots de la RAW. Si es 1 un periodo de RAW individual se compondrá de máximo 8 slots y un valor máximo de C de 2047, limitando la duración de cada slot 246.14 [ms]. Por otro lado si el valor es 0 se podrá contar con hasta 64 slots con un valor máximo de C de 255 y una duración máxima de slot de 31.1 ms (Liu & Lv, 2020).

- **NRawSlotCount:** Se utiliza para calcular la duración del slot de RAW.
- **NRawSlotNum:** Número de slots por grupo RAW.
- **Page:** Índice de páginas del subconjunto de AID, para una mayor organización de las estaciones.
- **Aid\_start:** Estación con la AID más baja asignada en el RAW.
- **Aid\_end:** Estación con la AID más alta asignada en el RAW.

Habiendo comprendido tanto la estructura del RawConfigFile como la función de cada uno de los campos se procede a revisar la configuración actual para cada variación de nodos de IEEE 802.11ah.

Los parámetros utilizados en primera instancia y que fueron los causantes de los resultados erróneos se encuentran organizados en la Tabla 18, estos eran generados de manera automática por el ejecutable diseñado para controlar ambas simulaciones (IEEE 802.11ah y LoRa) en un modelo único.

Tabla 18. Configuraciones de parámetros RAW iniciales.

<b>Nodos</b>	<b>NRPS</b>	<b>NRawGroup</b>	<b>NRawSlotCount</b>	<b>NRawSlotNum</b>	<b>Rango AID</b>
50	1	1	849	1	1-50
100	1	1	849	1	1-50, 51-100
150	1	1	849	1	1-50, 51-100, 101-150
200	1	1	849	1	1-50, 51-100, 101-150, 151-200
250	1	1	849	1	1-50, 51-100, 101-150, 151-200, 201-250

					1-50, 51-100, 101-
300	1	1	849	1	150, 151-200, 201-
					250, 251-300

---

Fuente: Autor.

Cabe mencionar que en la Tabla 18 no se han incluido los valores para RawControl, CrossSlotBoundary y SlotFormat, debido a que estos siempre serán 0, 1 y 1 respectivamente, durante todas las iteraciones de simulación. De esta manera en todas las simulaciones solo podrán acceder al canal las STAs asignadas en cada grupo, las estaciones tendrán la capacidad de terminar su transmisión aunque su ventana haya terminado, el valor máximo de C será 2047 y el máximo de slots 8.

Tanto el valor de C y el número de slots deberán asignarse respetando un límite, este límite es la duración entre cada beacon, que para todas las simulaciones de IEEE 802.11ah será de 102400  $\mu$ s. La duración de un slot o ranura de RAW se calcula según la Ec. 8 (ver Sección 2.1.1.2.5 del Capítulo II).

Con el objetivo de obtener mejores resultados en las simulaciones de 100 nodos de IEEE 802.11ah en adelante, se definió una nueva forma de agrupar las estaciones en grupos más pequeños, vigilando de cerca que se respete el valor adecuado de NRawSlotCount de acuerdo con la Ec. 8.

$$D_{slot} = 500\mu s + C_{slot} * 120\mu s$$

$$D = 500\mu s + 849 \times 120\mu s$$

$$D = 102380\mu s$$

El resultado se deberá a su vez utilizar dentro de la Ec. 9 para obtener la duración total de la Ventana de Acceso Restringido, al hacer el cálculo respectivo se evidencia que para un solo slot  $D_{RAW} = 102380\mu s$ , siendo menor que el tiempo entre cada beacon que es 102400 $\mu$ s.

$$D_{RAW} = D_{slot} * N_{RAW}$$

$$D_{\text{RAW}} = 1 * 102400\mu\text{s}$$

Con los cálculos realizados y comprobados se da paso a la creación de la Tabla 19, utilizando el nuevo valor de NRawSlotCount para las simulaciones con nodos mayores a 100. Se decidió conservar la configuración previa de los parámetros de RAW para 50 y 100 nodos de IEEE 802.11ah, debido a que las métricas del primer barrido arrojaron resultados satisfactorios, no siendo así el caso del resto de configuraciones.

Tabla 19. Lista de nuevas configuraciones de RAW para las iteraciones de IEEE 802.11ah.

<b>Nodos</b>	<b>NRPS</b>	<b>NRawGroup</b>	<b>NRawSlotCount</b>	<b>NRawSlotNum</b>	<b>Rango AID</b>
50	1	1	829	1	1-50
100	2	4	209	1	1-50, 50-100
150	2	1	849	1	1-127, 128-150
200	4	1	849	1	1-127, 128-150, 151-177, 178-200

Fuente: Autor.

Al realizar de nueva cuenta las simulaciones sin interferencia de LoRa con este nuevo conjunto de parámetros de RAW (ver Tabla 19), se obtuvo como resultado nuevos valores de throughput y porcentaje de paquetes perdidos, mismos que se muestran en la Tabla 20. El cambio más notable se encuentra en la simulación de 200 nodos, donde se pasó de un casi 95% de paquetes perdidos a tan solo un 8.36%.

Tabla 20. Nuevos resultados obtenidos de las configuraciones de parámetros de RAW.

<b>Nodos IEEE 802.11ah</b>	<b>Throughput (Kbps)</b>	<b>Paquetes perdidos (%)</b>
50	8.08444	0.219539
100	16.0622	0.441745
150	23.8267	2.57732

Luego de aplicadas las configuraciones del caso dentro del fichero ejecutable, de manera que cada grupo de parámetros de RAW se maneje de acuerdo con la Tabla 19, se procedió a simular cada una de las iteraciones contabilizando la interferencia de LoRa, tal y como se desarrolló en el Capítulo III.

Estas correcciones realizadas al funcionamiento de la simulación dan paso a un análisis más detallado de cada métrica planteada.

#### 4.2.1 THROUGHPUT

El throughput o tasa de transferencia efectiva, se puede definir como la capacidad del enlace para transportar información útil, es decir, es una medida de la cantidad de información útil que puede transmitirse por unidad de tiempo (Contreras Higuera , 2012). En base a esta definición, el throughput en la simulación de IEEE 802.11ah se calcula con la Ec. 23, mediante el producto de la totalidad de paquetes exitosos y reenviados y la longitud del Payload en bits, dividido para el tiempo de simulación, esto dará como resultado el throughput en bps.

$$\mathit{throughput} = \frac{(totalPaquetesExitosos + totalPaquetesRespondidos) * longitudPayload * 8}{tiempoSimulación} \quad (\text{Ec. 23})$$

Utilizando los datos recolectados al final del Capítulo III se han generado las gráficas respectivas para cada una de las variaciones e iteraciones de simulación. Cada una de las variaciones ha sido graficada de manera individual, con el fin de analizar más a detalle el comportamiento de la tecnología IEEE 802.11ah cuando la cantidad de sus nodos comienza a aumentar, tendiendo a formar una red densa.

#### 50 Nodos de IEEE 802.11ah

El primer set o conjunto de datos corresponde a 50 nodos de IEEE 802.11ah, en la Figura 84 se encuentra graficado linealmente el throughput en relación con la cantidad de nodos de LoRa, siendo esta última variable la independiente. Como dato adicional, el throughput de la simulación de IEEE 802.11ah sin la interferencia de LoRa es de 8.08444 Kbps, este valor actuará como un valor referencial.

Entre el rango de 10 y 700 nodos de LoRa, la simulación ha dado como resultado un throughput que ronda el valor referencial con un porcentaje de cambio de 0.27% en decremento y 0.44% en incremento<sup>17</sup>. Los resultados empiezan a variar drásticamente y en una clara tendencia a la baja desde los 1000 nodos de LoRa, con un porcentaje de cambio de -1.1%. Entre 2000 y 6000 nodos de LoRa es donde se registra una caída desde -1.21% hasta -14.62% del valor referencial, que corresponde a ir desde 7.98667 Kbps hasta 6.90222 Kbps de throughput, es decir, una reducción de 1.18222 Kbps. El valor final obtenido en 8000 nodos LoRa es de 5.05333 Kbps, esto representa una disminución del 37.49% desde el valor referencial. De todas estas iteraciones se puede concluir que, conforme al crecimiento de la cantidad de nodos LoRa el throughput de la red disminuye, llegando a perder hasta un total de 3.03111 Kbps en la última iteración (8000 nodos LoRa).

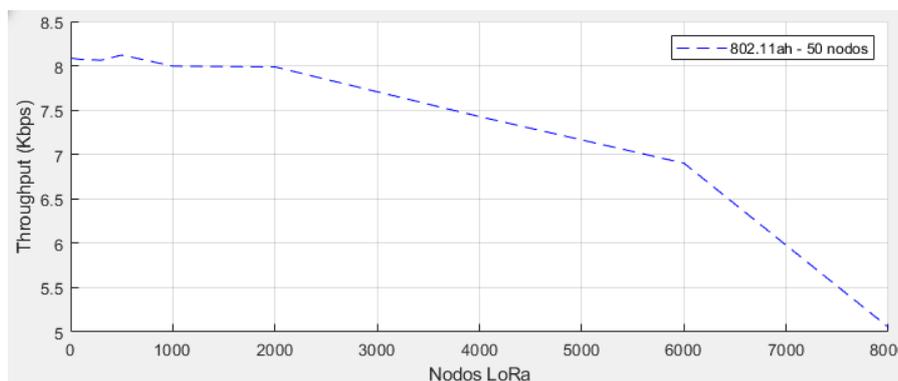


Figura 84. Gráfica lineal del throughput para 50 nodos de IEEE 802.11ah.  
Fuente: Autor.

<sup>17</sup> El porcentaje de cambio en incremento o decremento se obtiene dividiendo el valor resultante de la iteración por el valor referencial y luego multiplicando por 100.

La Figura 85 utiliza los mismos resultados de la Figura 84, pero ajustados a una curva logarítmica, esto permite identificar rápida y fácilmente que se trata de una curva descendente, teniendo en cuenta los valores obtenidos en el análisis anterior, no se recomienda utilizar la gráfica logarítmica para extraer datos, al tener una precisión menor a la gráfica lineal debido al ajuste de tendencia, el objetivo de esta gráfica no es otro que demostrar la curva de tendencia descendente o ascendente respecto a las iteraciones realizadas para la variación actual.

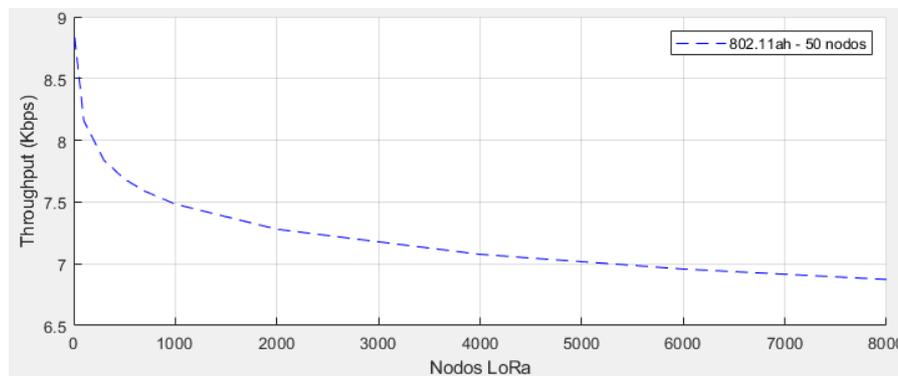


Figura 85. Gráfica logarítmica del throughput para 500 nodos de IEEE 802.11ah.  
Fuente: Autor.

### 100 Nodos de IEEE 802.11ah

Al aumentar la cantidad de nodos de IEEE 802.11ah se obtiene un mayor throughput debido al aumento en la cantidad de transmisiones realizadas, esto se ve reflejado en el valor referencial (sin el impacto de la interferencia de LoRa) que es de 15.8978 Kbps. Sin embargo, tal como se observa en la Figura 86, conforme aumenta la cantidad de nodos de LoRa el throughput desciende hasta alcanzar su punto más bajo en 13.5289 Kbps con 8000 nodos de LoRa, lo que representa una disminución de 14.9% (2.3689 Kbps) del valor referencial.

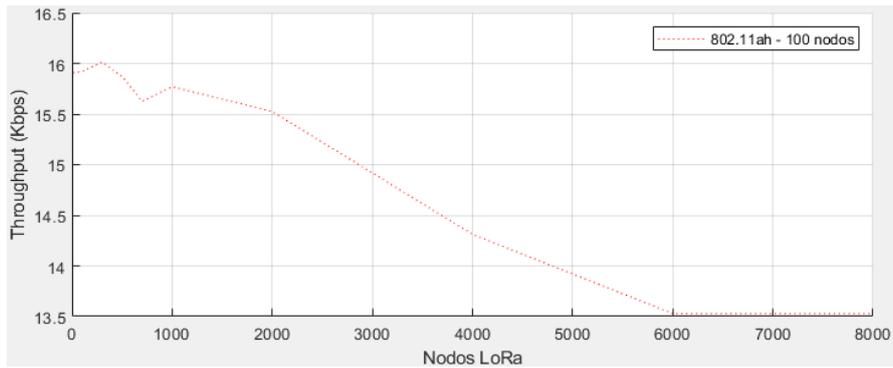


Figura 86. Gráfica lineal del throughput para 100 nodos de IEEE 802.11ah.  
Fuente: Autor.

De igual manera a los resultados mostrados para 50 nodos de IEEE 802.11ah, la Figura 87 contiene una curva logarítmica descendente, siendo igual de pronunciada que en la Figura 85. En este sentido, se puede concluir hasta el momento que la relación entre la cantidad de nodos de LoRa y el throughput de la red IEEE 802.11ah es inversamente proporcional.

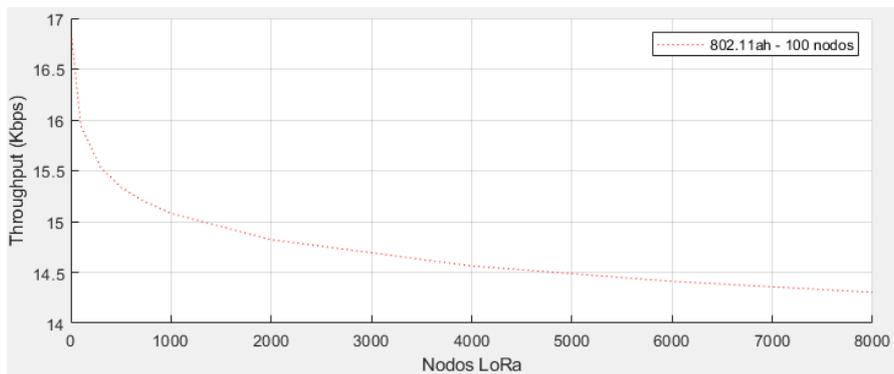


Figura 87. Gráfica logarítmica del throughput para 100 nodos de IEEE 802.11ah.  
Fuente: Autor.

### 150 Nodos IEEE 802.11ah

Tomando en cuenta que el valor referencial para 150 nodos de IEEE 802.11ah es de 23.8267 Kbps, en la Figura 88 se identificó un incremento del 0.09% en la simulación con 10 nodos de LoRa, esto presumiblemente debido a la pseudo aleatoriedad con la que se maneja NS-3, ya que el throughput medido en las siguientes iteraciones (100 a 1000 nodos de LoRa) presentaron una disminución de hasta el 1.1%. Siendo valores aceptables si se toma en cuenta la cantidad total de dispositivos totales. Por otra parte, el punto de inflexión desde el que empieza a caer drásticamente el throughput es la iteración con 2000

nodos de LoRa, donde se registra un 2.01% de disminución en relación con el valor referencial. Para la iteración con 4000 nodos de LoRa, el throughput de la red ha disminuido en 10.69%, es decir, una reducción de 2.5467 Kbps. Entre los 6000 y 8000 nodos de LoRa se mantiene una disminución del 13.04% o 3.1067 Kbps, siendo esta reducción de throughput más significativa en comparación a las dos variaciones anteriores.

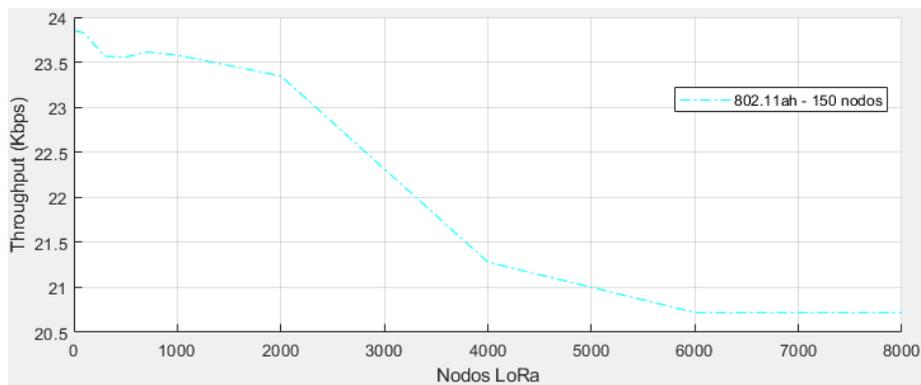


Figura 88. Gráfica lineal del throughput para 150 nodos de IEEE 802.11ah.  
 Fuente: Autor.

La curva logarítmica para el conjunto de iteraciones descritas mantiene su forma decreciente, ver Figura 89.

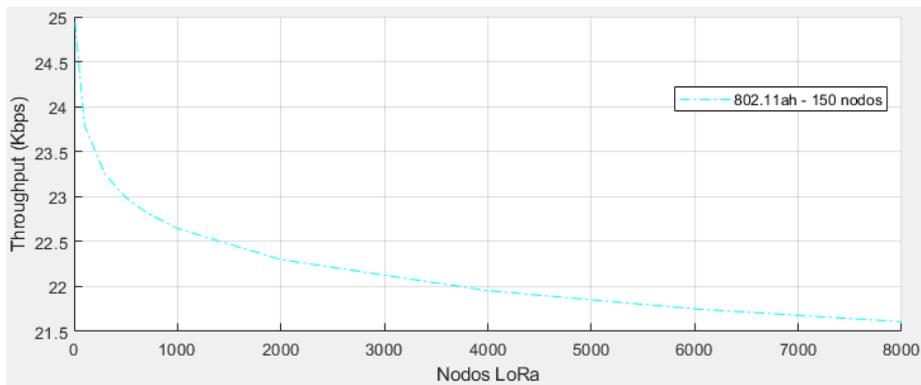


Figura 89. Gráfica logarítmica del throughput para 150 nodos de IEEE 802.11ah.  
 Fuente: Autor.

### 200 Nodos de IEEE 802.11ah

Los resultados obtenidos en esta variación (ver Figura 90) tuvieron que ser corregidos debido a la existencia de valores atípicos, estos se produjeron debido a una concentración de los nodos de LoRa en una misma ubicación provocada por el propio

modelo de movilidad del simulador, que está basado en generar localizaciones pseudo-aleatorias. Los puntos que se revisaron y se simularon con una cantidad diferente de nodos de LoRa fueron: 300, 2000 y 4000; al reducir en uno la cantidad de nodos de los puntos mencionados se consiguió generar nuevas localizaciones para cada estación, con una distribución más adecuada (ver Figura 91).

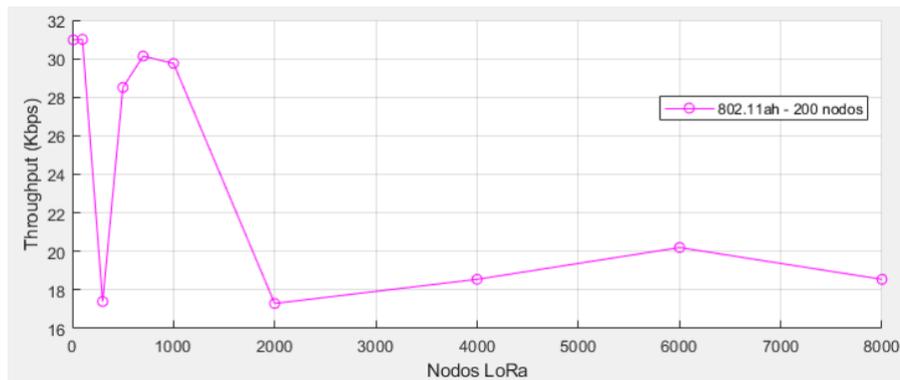


Figura 90. Gráfica lineal del throughput para 200 nodos de IEEE 802.11ah, previa corrección.

Continuando con el análisis (ver Figura 91), el valor referencial del throughput es de 30.956 Kbps y similar a lo visto en la Figura 88, se identificó un incremento (0.14%) para la iteración de 10 nodos de LoRa. Mientras que, en 300 nodos de LoRa se registró un throughput de 30.04 Kbps, que significa una reducción de menos del 3%, para luego registrar una disminución de solo el 7.91% en 700 nodos. En el punto correspondiente a 2000 nodos de LoRa, la disminución del throughput alcanza el 4.72% (29.48 Kbps). El valor medio del Throughput para las iteraciones entre 4000 y 8000 nodos de LoRa es de 22.18 Kbps. De modo que en su punto más bajo el Throughput cae hasta los 18.51 Kbps, que es equivalente a una disminución del 60% de su valor referencial.

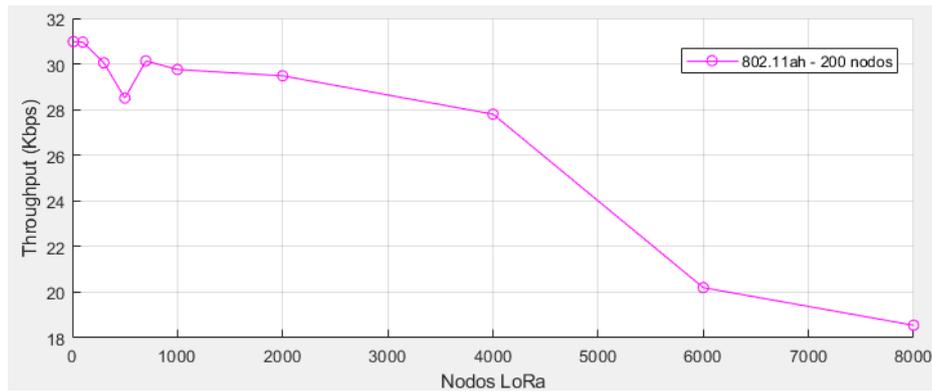


Figura 91. Gráfica lineal del throughput para 200 nodos de IEEE 802.11ah, corregida.  
Fuente: Autor.

La curva logarítmica de la Figura 92 representa al conjunto de iteraciones previamente analizadas, lo que se puede extraer de esta gráfica es su tipo, siendo esta decreciente y manteniendo a su vez, la concordancia con el resto de las curvas logarítmicas asociadas al Throughput.

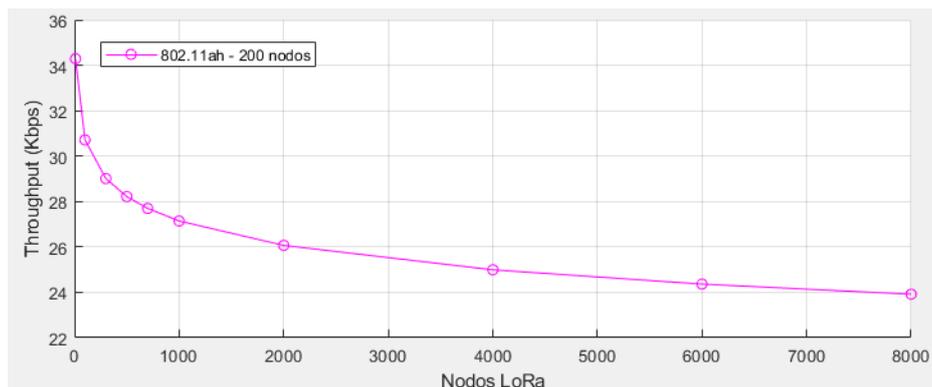


Figura 92. Gráfica logarítmica del throughput para 200 nodos de IEEE 802.11ah.  
Fuente: Autor.

### Todas las variaciones de nodos de IEEE 802.11ah

Al agrupar todas las gráficas lineales del Throughput hasta el momento, se obtiene la Figura 93, donde cada color representa a un conjunto de resultados específico, la azul 50 nodos, la roja 100 nodos, los cian 150 nodos y la violeta 200 nodos, todos de IEEE 802.11ah. Comparando las 3 primeras líneas se puede notar una similitud clara entre ellas, iniciando en un valor cercano al referencial y luego disminuyendo poco a poco conforme se acerca a 8000 nodos de LoRa.

La recta asociada a 200 nodos de IEEE 802.11ah mantiene el comportamiento de reducir el Throughput conforme se aumentan los nodos de LoRa, destacando que las caídas en 6000 y 8000 nodos de LoRa han sido revisadas a detalle, simulando puntos adicionales a 6000 y 8000, con el fin de determinar si estos valores son obtenidos debido a un bug en la pseudo-aleatoriedad de las ubicaciones u otra razón.

Finalmente, se llegó a determinar que este par de valores tan marcados se debe al modelo utilizado para calcular el SINR, donde se considera la duración del solapamiento para calcular la energía de la interferencia, esto provoca que los solapamientos de tipo parcial den como resultado un valor SINR sobreestimado, para ampliar acerca de este fenómeno se recomienda leer la Sección 4.3 del presente capítulo.

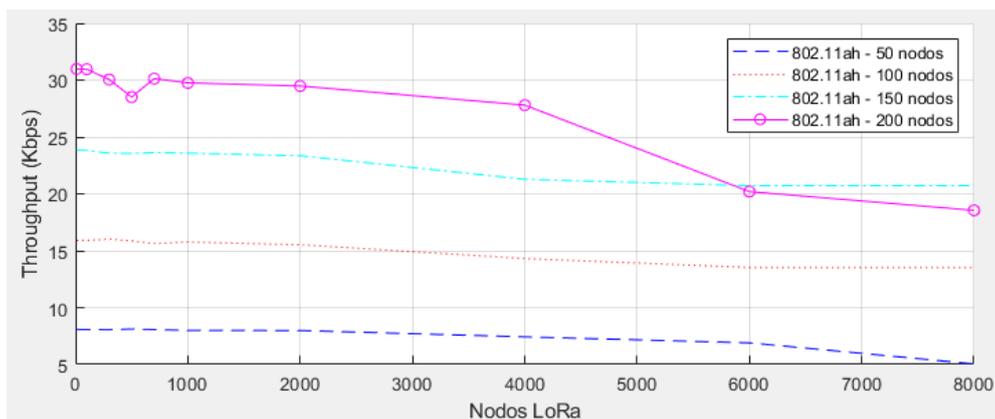


Figura 93. Gráfica lineal del throughput para todas las variaciones de IEEE 802.11ah.  
Fuente: Autor.

El comportamiento del grupo de resultados para 200 nodos de IEEE 802.11ah se puede evidenciar inclusive en las curvas logarítmicas de la Figura 94, donde la curva de 200 nodos de IEEE 802.11ah presenta una inclinación mucho mayor que las demás curvas, pero sin llegar a colocarse por debajo de la curva de 150 nodos IEEE 802.11ah.

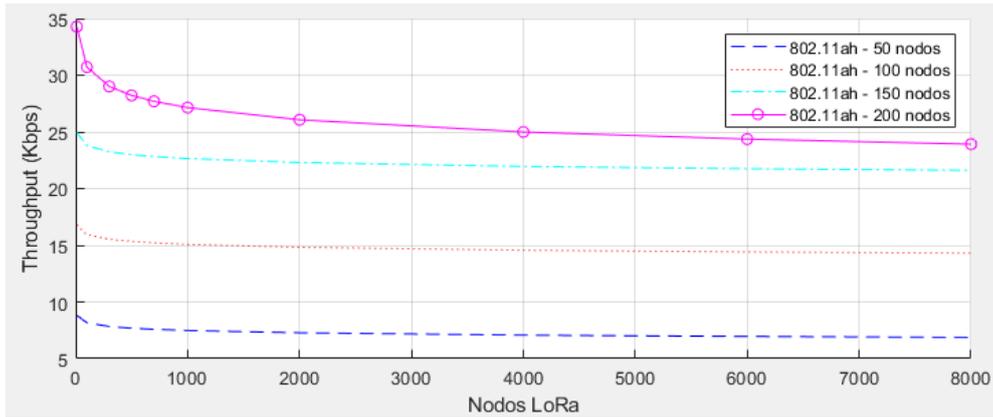


Figura 94. Gráfica logarítmica del throughput para todas las variaciones de IEEE 802.11ah.  
Fuente: Autor.

### Variación del radio de LoRa

Adicionalmente se han graficado los resultados correspondientes a una variación del radio de LoRa, siguiendo con lo establecido en la Tabla 25. Estos datos se encuentran en la Figura 95, donde se observa la gráfica lineal que corresponde múltiples iteraciones de una simulación de 100 nodos de IEEE 802.11ah en 200 m de radio. Cada iteración aumenta el radio de distribución de las estaciones de LoRa hasta un máximo de 4000 metros. Al analizar todas las iteraciones en una sola gráfica, se puede inferir que el Throughput de la red disminuye conforme el radio de la topología LoRa aumenta, que representa una disminución de cerca del 10% del Throughput.

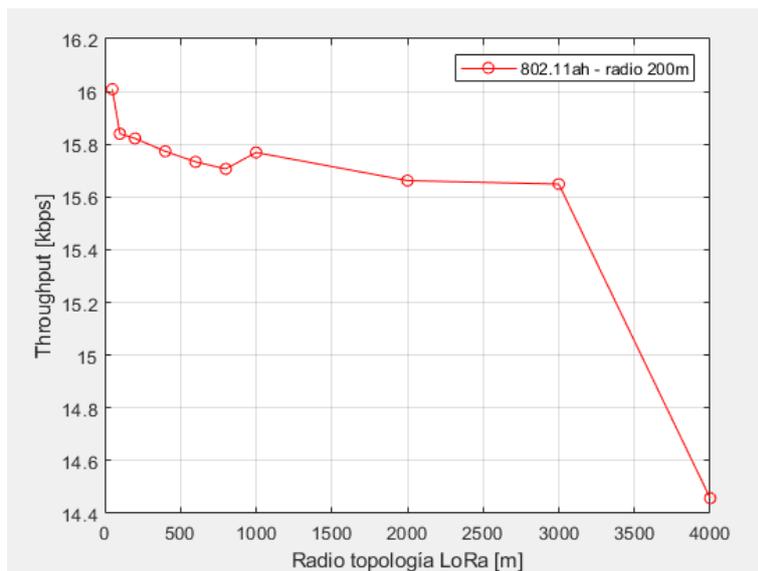


Figura 95. Gráfica lineal del Throughput para 100 nodos de IEEE 802.11ah, variando el Radio de LoRa.  
Fuente: Autor.

### Variación del Payload de IEEE 802.11ah

Otro de los parámetros de rendimiento de la Tabla 8 que se varió dentro de la simulación es el tamaño del Payload, en vista de que la longitud de la carga útil escogida para el resto de las simulaciones (100 bytes) no es ni el menor ni el mayor valor que se puede asignar de acuerdo con el MCS escogido (MCS 2). De hecho el valor máximo es 2297 bytes, de acuerdo con la Ec. 24. Donde la cantidad máxima de octetos a transmitirse sin agregación es  $511 \cdot [\text{max\_symbol}]$  para una S1G PDDU, ver la variable `aPSDUMaxLengthWithNoAggregation` en (IEEE 802.11 Working Group, 2016).

$$\mathbf{max\_ul\_bytes} = \frac{\mathbf{max\_symbol} \cdot \mathbf{bps} - \mathbf{SERVICE\_TAIL\_BIT}}{8} \quad (\text{Ec. 24})$$

$$\mathbf{max\_ul\_bytes} = \frac{511 \cdot 36 - 14}{8}$$

$$\mathbf{max\_ul\_bytes} = 2297.75 \text{ [bytes]}$$

De esta manera se simularon múltiples iteraciones con un valor fijo de nodos LoRa de 1000, una cantidad fija de estaciones IEEE 802.11ah de 100 y un radio en común para ambas tecnologías de 200 m. Por otro lado, el parámetro que aumentaba en 200 durante cada iteración es el Payload de IEEE 802.11ah. Formando la gráfica de la Figura 96, aquí se aprecia el aumento gradual del Throughput debido al propio aumento del Payload, recordando que esta medida de rendimiento obedece a la Ec. 23. Cada aumento es de aproximadamente 30 Kbps por cada 200 bytes incrementados en el Payload, este patrón se repite hasta que se alcanza los 1800 bytes de longitud de Payload, en este punto la razón de aumento antes mencionada disminuye al acercarse Payload máximo obtenido en la Ec. 24, en este punto el Throughput es de 290.787 Kbps.

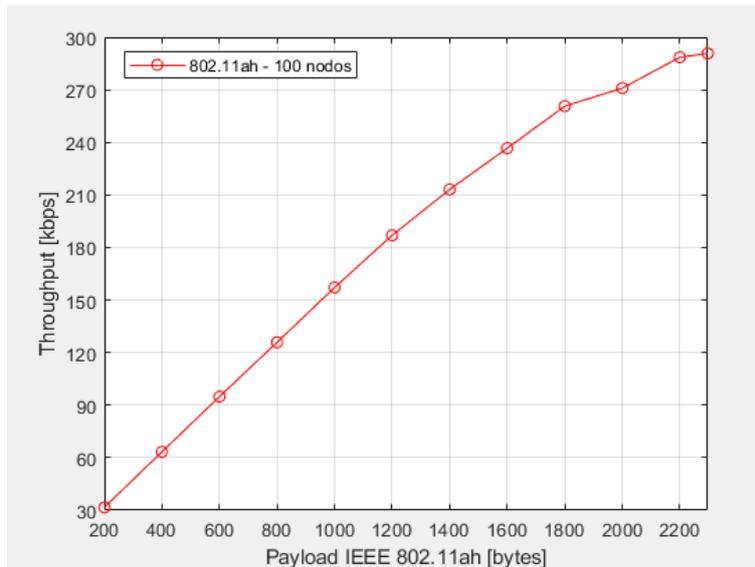


Figura 96. Gráfica lineal del Throughput para 100 nodos de IEEE 802.11ah, variando el Payload.  
Fuente: Autor.

#### 4.2.1 TOTAL DE PAQUETES PERDIDOS

La totalidad de los paquetes perdidos se obtiene a partir de la Ec. 24, la relación entre los paquetes respondidos y enviados permiten calcular el porcentaje de paquetes perdidos por la interferencia acumulada.

$$TotalPaquetesPerdidos (\%) = 100 - 100 * \frac{totalPaquetesRespondidos}{totalPaquetesEnviados} \quad (Ec. 24)$$

Al ejecutar la simulación de IEEE 802.11ah por defecto se tendrá un cierto porcentaje de paquetes perdidos, estos se deben al propio modelado de la tecnología en NS-3, donde se contabiliza solamente el ruido térmico y la interferencia generada por los propios paquetes IEEE 802.11ah.

#### 50 Nodos IEEE 802.11ah

El valor de referencia para una simulación de IEEE 802.11ah de 50 nodos con los parámetros de la Tabla 8 es de 0.219539% de paquetes perdidos. Este valor se ha mantenido durante la simulación con interferencia LoRa de 10 nodos. No obstante en la Figura 97, con 100 nodos de LoRa los paquetes perdidos se duplican pasando a 0.439078%, que equivale a un incremento del 100%, incremento que no es tan extremo al tratarse de un valor referencial muy pequeño.

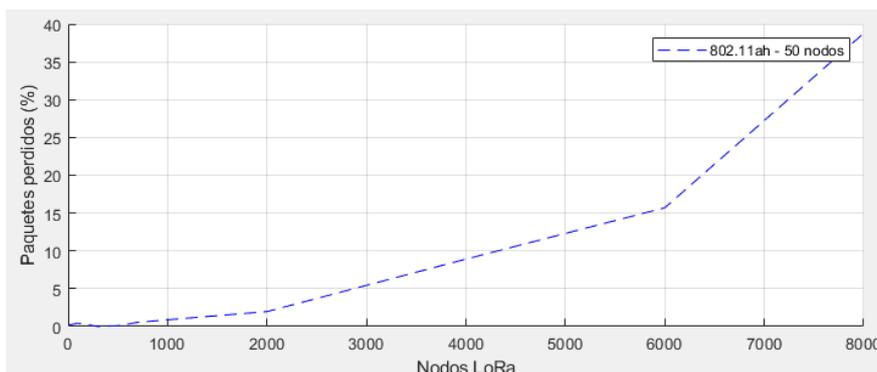


Figura 97. Gráfica lineal de los paquetes perdidos para 50 nodos de IEEE 802.11ah.  
Fuente: Autor.

El porcentaje de paquetes perdidos tiende a aumentar conforme crece el número de nodos de LoRa, de acuerdo con la Figura 98, donde el tipo de curva obtenida del ajuste logarítmico es creciente.

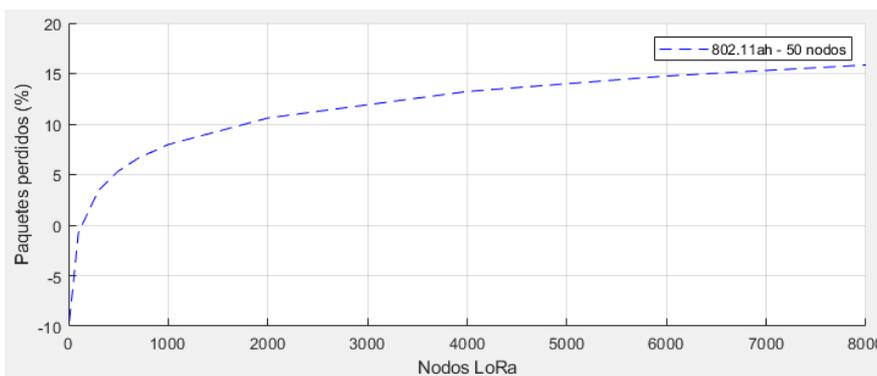


Figura 98. Gráfica logarítmica de los paquetes perdidos para 50 nodos de IEEE 802.11ah.  
Fuente: Autor.

### 100 Nodos de IEEE 802.11ah

Los resultados obtenidos en la métrica actual para esta variación (ver Figura 99) tienen como máximo 18.8889% de paquetes perdidos. Por otro lado, el porcentaje de pérdidas para una simulación de IEEE 802.11ah sin nodos LoRa es de 2.37438%. Al comparar estos dos extremos es posible concluir que, con 6000 o más nodos de LoRa las pérdidas de paquetes para IEEE 802.11ah pueden aumentar hasta 695.53 veces, que equivale a un incremento de 16.51452% en los paquetes perdidos. Cabe mencionar que las pérdidas de paquetes se mantienen dentro de valores aceptables hasta 2000 o incluso 3000 nodos de LoRa.

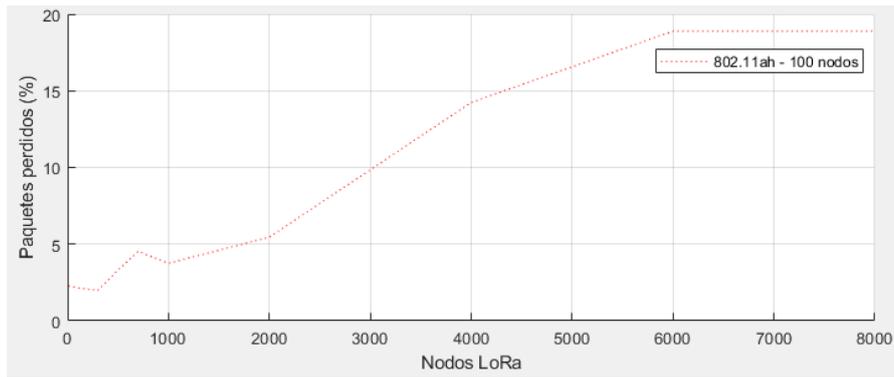


Figura 99. Gráfica lineal de los paquetes perdidos para 100 nodos de IEEE 802.11ah.  
Fuente: Autor.

De igual manera, la Figura 100 también corresponde a una curva creciente, con una inclinación no tan pronunciada.

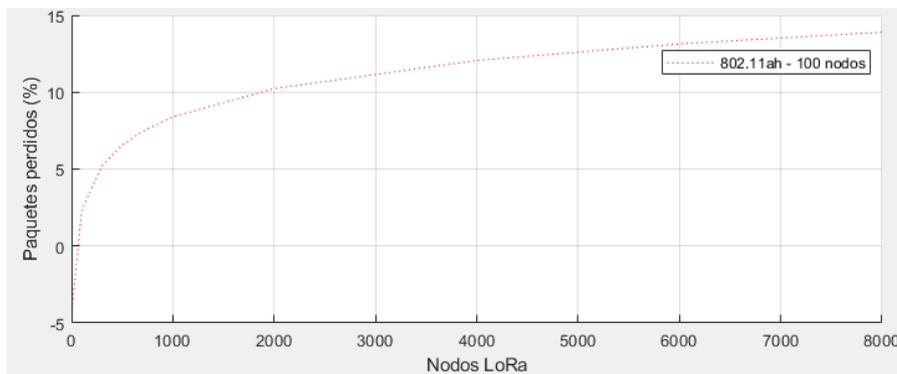


Figura 100. Gráfica logarítmica de los paquetes perdidos para 100 nodos de IEEE 802.11ah.  
Fuente: Autor.

### 150 Nodos de IEEE 802.11ah

Con una gran similitud a la variación anterior, los resultados de la Figura 101 muestran valores aceptables (5% - 10%) hasta 2000 y 3000 nodos de LoRa aproximadamente, identificando un aumento de 2.92286% y 7.92286% en los paquetes errados, respectivamente. Así mismo, entre 6000 y 8000 nodos LoRa se obtiene el mayor incremento en el porcentaje de pérdida paquetes, alcanzando hasta un 22.7928%.

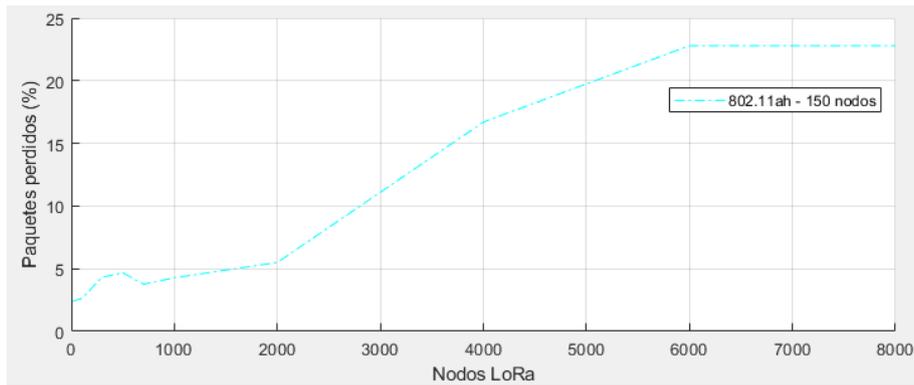


Figura 101. Gráfica lineal de los paquetes perdidos para 150 nodos de IEEE 802.11ah.  
Fuente: Autor.

La curva de tipo creciente en la Figura 102, tiene una inclinación más pronunciada que las vistas anteriormente, señalando un aumento en la pérdida de paquetes desde 1000 nodos de LoRa.

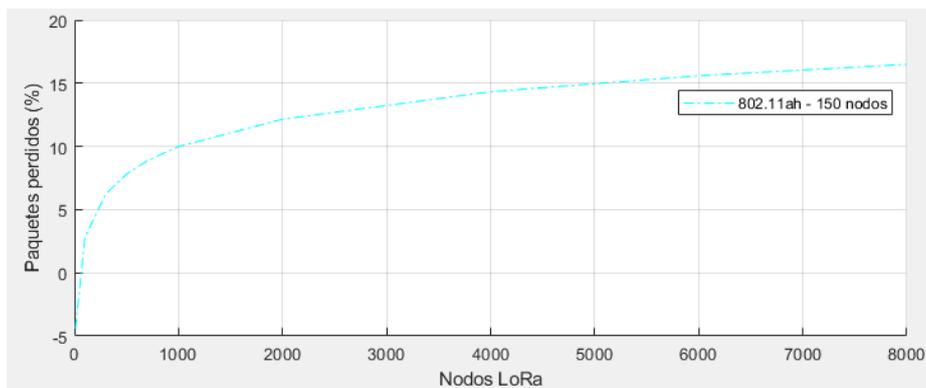


Figura 102. Gráfica logarítmica de los paquetes perdidos para 150 nodos de IEEE 802.11ah.  
Fuente: Autor.

## 200 Nodos de IEEE 802.11ah

De nueva manera, se debe mencionar que se realizó una corrección de las simulaciones en 300, 2000 y 4000 nodos de LoRa para solucionar el problema de pseudo-aleatoriedad de las ubicaciones de las estaciones (ver Figura 103).

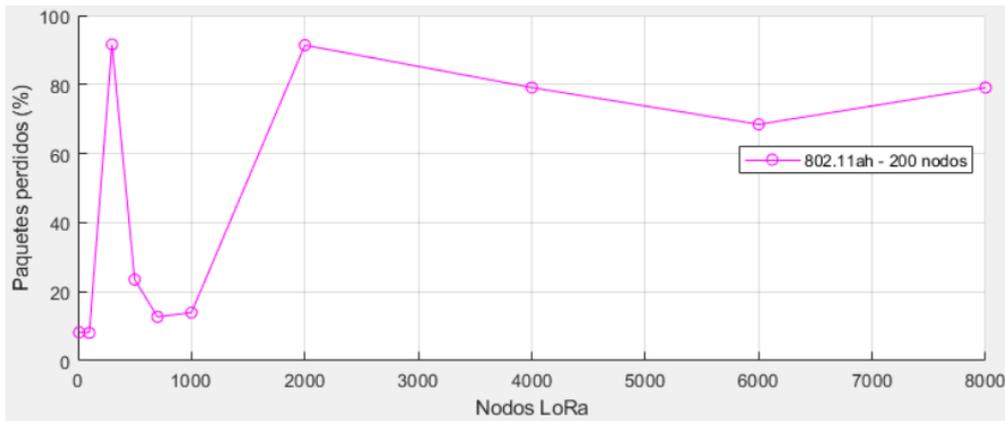


Figura 103. Gráfica lineal de los paquetes perdidos para 200 nodos de IEEE 802.11ah, previa corrección.  
Fuente: Autor.

Se empezará mencionando que el valor referencial para este caso en específico es de 8.36084%, a partir de dicho valor se identifica un ligero aumento hasta el 15.89% en 2000 nodos de LoRa, sin embargo a partir de este punto se dispara hasta un 79% de paquetes perdidos en 8000 nodos de LoRa (ver Figura 104). Se debe aclarar que este cambio tan radical en el aumento de paquetes perdidos no se debe a ningún bug, más sin embargo, se da como resultado del modelo utilizado para calcular el SINR y su consideración con los solapamientos parciales, este hecho se explicará más adelante en la Sección 4.3.

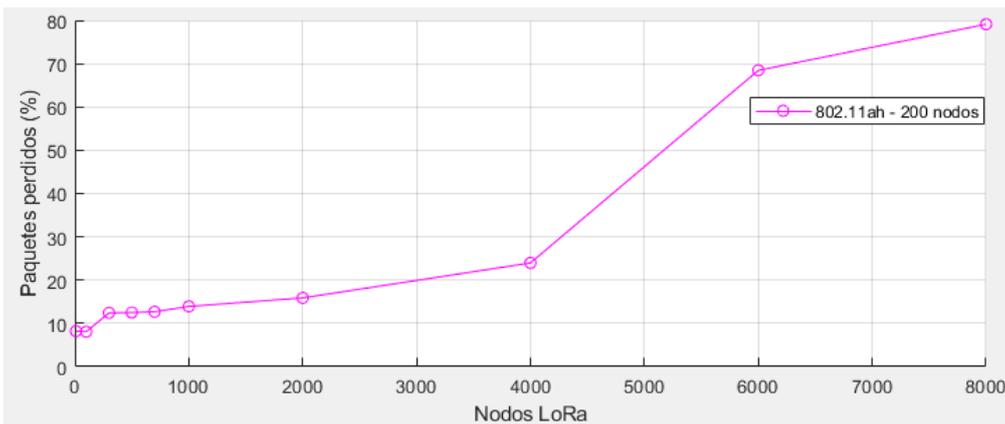


Figura 104. Gráfica lineal de los paquetes perdidos para 200 nodos de IEEE 802.11ah, corregido.  
Fuente: Autor.

La curva de tendencia de la Figura 103 tiene la mayor inclinación obtenida hasta el momento, debido a los valores tan abruptamente altos obtenidos como resultado en la figura 105.

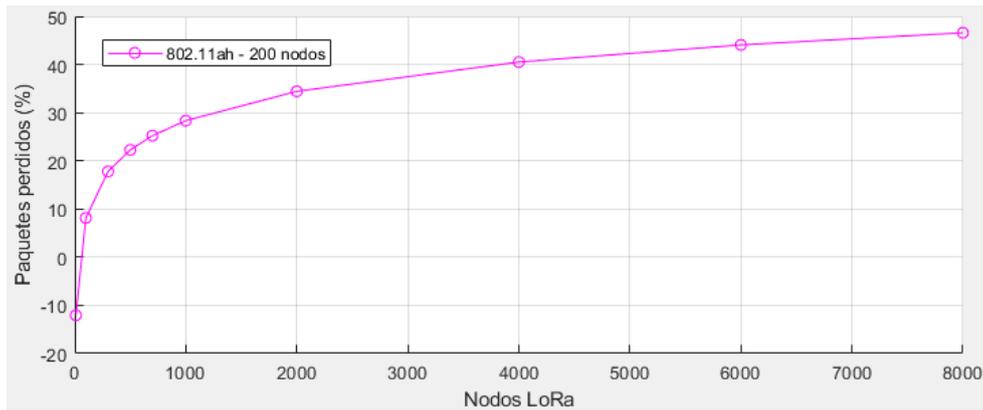


Figura 105. Gráfica logarítmica de los paquetes perdidos para 200 nodos de IEEE 802.11ah.  
Fuente: Autor.

### Todas las variaciones de nodos de IEEE 802.11ah

La Figura 106 contiene todos los resultados antes descritos y analizados, lo que más resalta de este es la diferencia entre las variaciones de 50, 100 y 150 nodos de IEEE 802.11ah con la variación de 200 nodos. Entre las 3 primeras variaciones se puede notar una tendencia al aumento gradual en el porcentaje de paquetes perdidos, siendo la línea cian (150 nodos IEEE 802.11ah) la que mayor pérdida presenta. A pesar de que la recta de 200 nodos de IEEE 802.11ah presenta valores por encima de los resultados anteriores para todos los nodos de LoRa (10 - 8000), el comportamiento de los datos permite concluir que existe una relación directamente proporcional entre los nodos de LoRa y el porcentaje de paquetes perdidos de IEEE 802.11ah. En otras palabras, mientras mayor sea la cantidad de nodos de la tecnología interferente, mayor será la probabilidad de pérdida de paquetes de la tecnología deseada.

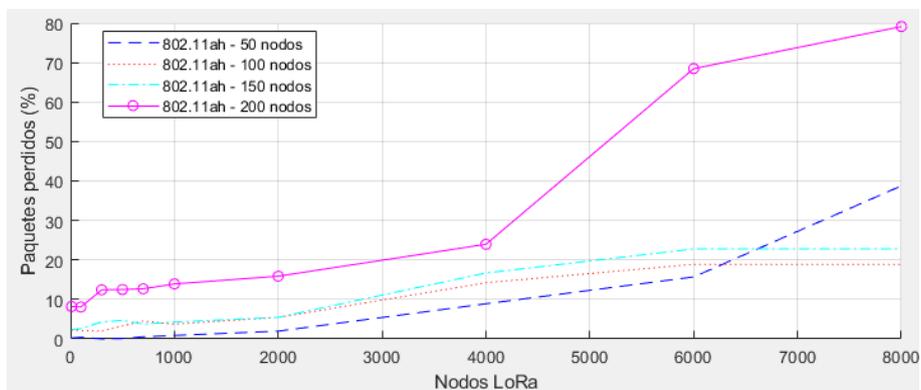


Figura 106. Gráfica lineal de los paquetes perdidos para todas las variaciones de IEEE 802.11ah.  
Fuente: Autor.

Todas las curvas logarítmicas de la Figura 107 son de tipo creciente. Similar a lo observado en las gráficas lineales, las curvas de 50, 100 y 150 nodos de IEEE 802.11ah mantienen inclinaciones similares, por otro lado la simulación de 200 nodos de IEEE 802.11ah muestra una curva con cambios más marcados si se la compara con sus homólogas.

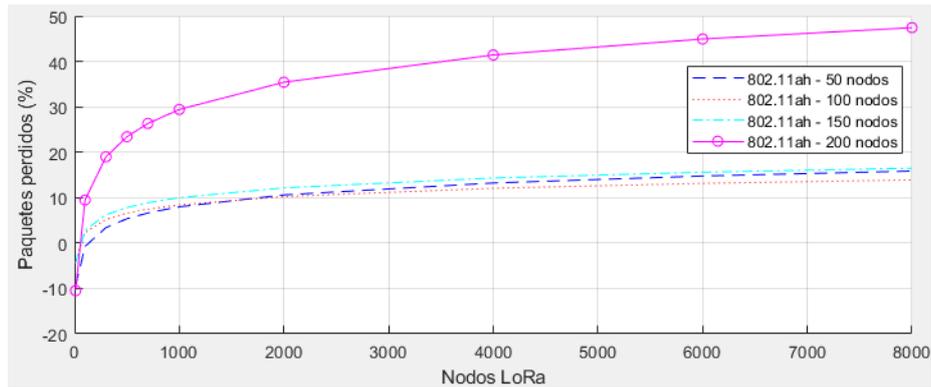


Figura 107. Gráfica logarítmica de los paquetes perdidos para todas las variaciones de IEEE 802.11ah.  
Fuente: Autor.

### Variación del radio de LoRa

Para la totalidad de paquetes perdidos también se ha decidido analizar que sucede al aumentar en cada iteración el radio de la topología de LoRa mientras se mantiene la de IEEE 802.11ah en 200m. La Figura 106 contiene dichos resultados, mismos que indican una relación directamente proporcional entre el Porcentaje de Paquetes Perdidos (PPP) y el radio de la topología de LoRa, partiendo de un valor de cercano al 2% de PPP hasta casi 14% al incrementar el radio de la simulación de LoRa a los 4000 m. A pesar de este incremento final considerable, se ha identificado una tendencia a un PPP menor al 6% hasta los 3000 m (ver Figura 108).

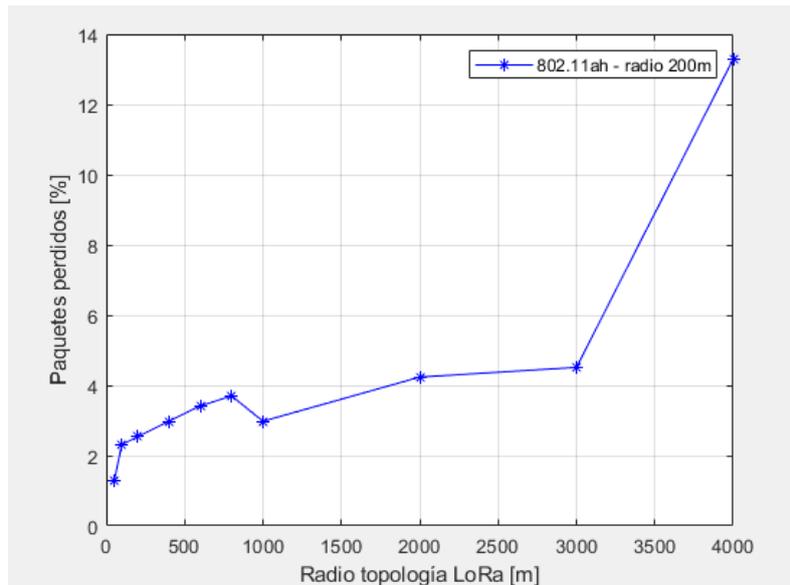


Figura 108. Gráfica lineal del Porcentaje de Paquetes Perdidos para 100 nodos de IEEE 802.11ah.  
Fuente: Autor.

### Variación del Payload de IEEE 802.11ah

Tal como se explicó en la sección anterior (4.2.1), se ha variado la longitud del Payload de IEEE 802.11ah, manteniendo fijo el resto de los parámetros (200 m de radio, 1000 nodos LoRa y 100 nodos IEEE 802.11ah). Formando la Figura 109, esta indica la relación que existe entre el Porcentaje de Paquetes Perdidos y el aumento gradual del Payload de IEEE 802.11ah. Tras analizar la gráfica, se puede afirmar que el PPP se mantiene dentro de un nivel bajo hasta los 1200 Bytes de longitud de Payload, pasado este punto el PPP aumenta hasta un pico máximo de 39.2265 %.

Este drástico aumento del PPP en los valores de Payload cercanos al límite de 2297 Bytes es comparable con la disminución del Throughput alrededor de estos mismos valores de la carga útil (ver Figura 96).

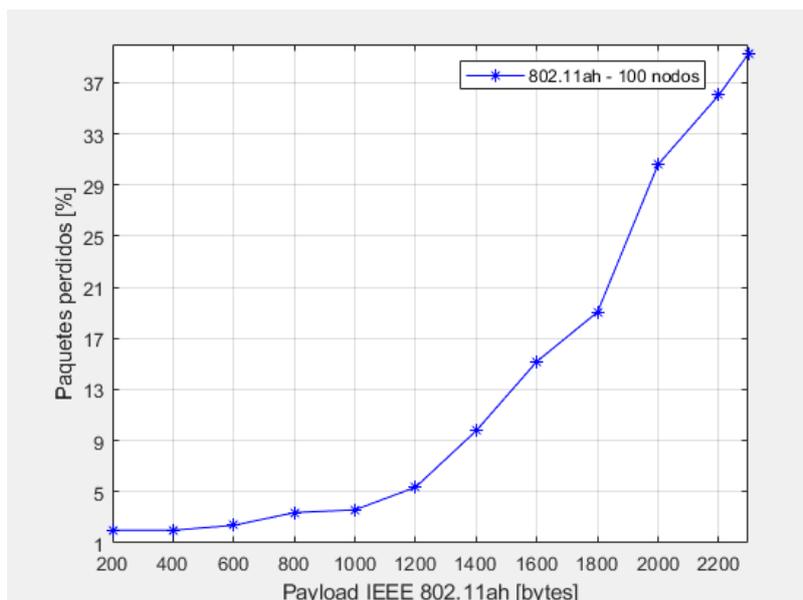


Figura 109. Gráfica lineal del PPP para 100 nodos de IEEE 802.11ah, variando el Payload.  
Fuente: Autor.

#### 4.2.2 SINR

El SINR es una medida de la calidad del canal, se obtiene de la relación entre la potencia de la señal sobre la sumatoria de la interferencia y el ruido de fondo, tal como se define en la Ec. 25.

$$SINR(dB) = 10 * \log_{10} \left( \frac{S}{I+N} \right) \quad (\text{Ec. 25})$$

Los valores de SINR corresponden al promedio de todas las transmisiones registradas que no hayan pasado el umbral de 16 dB establecido (ver Capítulo III), como nota adicional, NS-3 toma en cuenta cada una de las subportadoras para este proceso.

#### 50 Nodos de IEEE 802.11ah

Los resultados obtenidos de esta y de las demás variaciones no serán comparadas con ningún valor referencial, ya que el proceso por defecto del módulo de IEE 802.11ah no contabiliza un SINR con otras tecnologías.

Todos los valores de SINR en la Figura 110 son negativos, fenómeno que se produce cuando la potencia de la señal deseada es menor a la potencia de la interferencia + ruido. Hasta los 2000 nodos de LoRa el SINR no llega a ser menor de los -12 dB, sin embargo, conforme aumenta la cantidad de nodos de LoRa el SINR promedio baja hasta

un mínimo de -18 dB. Notando una clara relación entre la cantidad de nodos de LoRa y el SINR promedio.

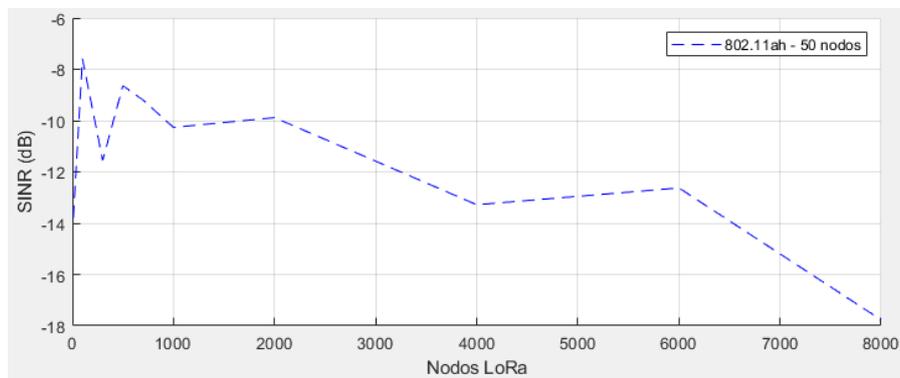


Figura 110. Gráfica lineal del SINR para 50 nodos de IEEE 802.11ah.  
Fuente: Autor.

La curva logarítmica de la Figura 111 se ajusta a los datos obtenidos del SINR promedio, dando como resultado una curva de tipo decreciente, se debe notar que la inclinación de la curva va a razón de 0.5 dB.

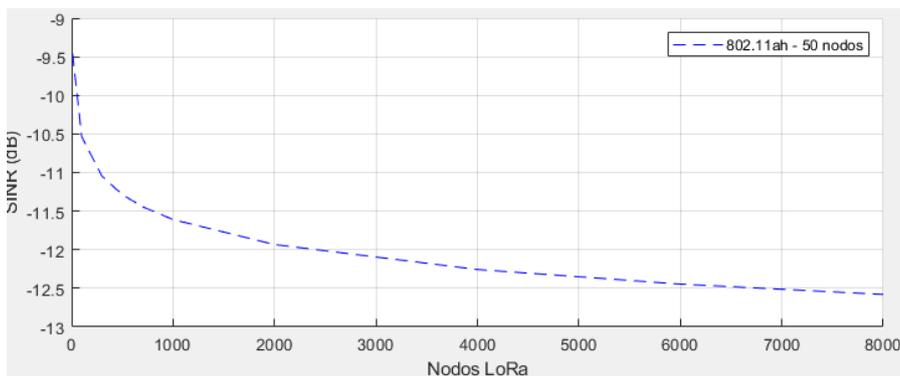


Figura 111. Gráfica logarítmica del SINR para 50 nodos de IEEE 802.11ah.  
Fuente: Autor.

### 100 Nodos de IEEE 802.11ah

Al aumentar la cantidad de nodos de la tecnología analizada, en la Figura 112 se distingue un comportamiento similar a la variación anterior (50 nodos IEEE 802.11ah) pero con un valor mínimo de casi -20 dB para 8000 nodos de LoRa. Lo que representa una disminución del SINR promedio de 2 dB.

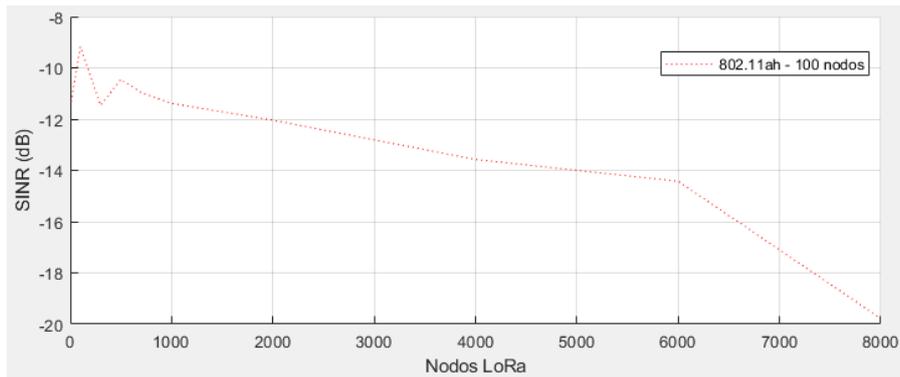


Figura 112. Gráfica lineal del SINR para 100 nodos de IEEE 802.11ah.  
Fuente: Autor.

La curva de la Figura 113 se ajusta logarítmicamente y de manera idéntica a la vista en la Figura 111 sin embargo, el grado de inclinación es mayor teniendo cambios a razón de 1 dB en lugar de 0.5 dB.

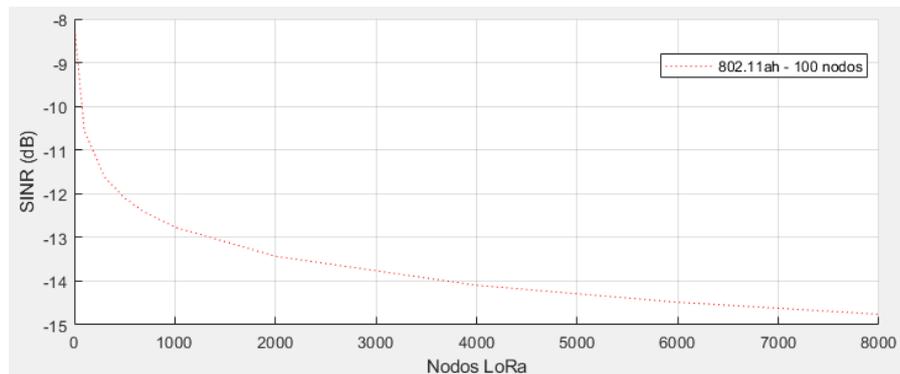


Figura 113. Gráfica logarítmica del SINR para 100 nodos de IEEE 802.11ah.  
Fuente: Autor.

### 150 Nodos de IEEE 802.11ah

Para esta cantidad de nodos, la gráfica lineal de la Figura 114 es casi idéntica a la variación anteriormente analizada. Ambas se mantienen entre -8 dB y -12 dB hasta los 2000 nodos de LoRa, inclusive el punto en 6000 nodos de LoRa ronda también los -14 dB, finalmente en el punto más alto de nodos de LoRa el SINR casi alcanza los -20 dB.

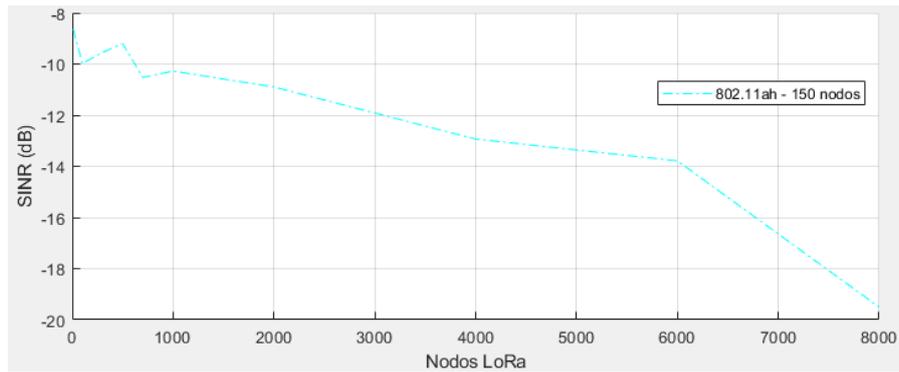


Figura 114. Gráfica lineal del SINR para 150 nodos de IEEE 802.11ah.  
Fuente: Autor.

Al contar con unos resultados tan similares en las gráficas lineales entre las tres primeras variaciones de IEEE 802.11ah, se espera que sus gráficas logarítmicas también presenten un comportamiento idéntico. Siendo todas son curvas decrecientes, incluyendo a la Figura 115.

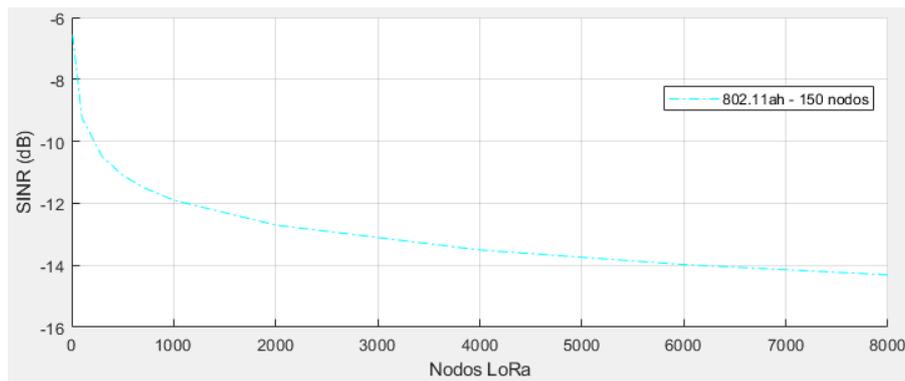


Figura 115. Gráfica logarítmica del SINR para 50 nodos de IEEE 802.11ah.  
Fuente: Autor.

### 200 Nodos de IEEE 802.11ah

En la Figura 116 se visualizan los resultados obtenidos en la última variación, desde el inicio presentan un comportamiento distinto a lo observado con anterioridad. Se distinguen valores de SINR más bajos en cada uno de los puntos (nodos LoRa) simulados. En primer lugar, se cuenta con valores que rondan los -12 dB y -16 dB entre 10 y 2000 nodos de LoRa. A pesar de ello, el punto observado en todas las variaciones de 6000 nodos de LoRa también ronda los -14 dB, igualmente el valor mínimo es de casi -20 dB.

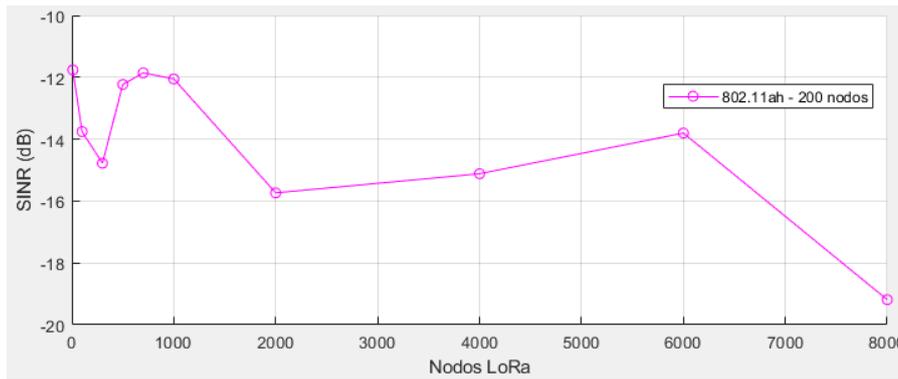


Figura 116. Gráfica lineal del SINR para 200 nodos de IEEE 802.11ah.  
Fuente: Autor.

Con esta última variación analizada es posible decir que todas las variaciones de SINR tienen una curva de tendencia decreciente en su gráfica logarítmica (ver Figura 117). Hecho que se ve reflejado en los comportamientos de los resultados en los puntos de análisis específicos, entre 10 y 2000 nodos, en 6000 nodos y en su valor mínimo.

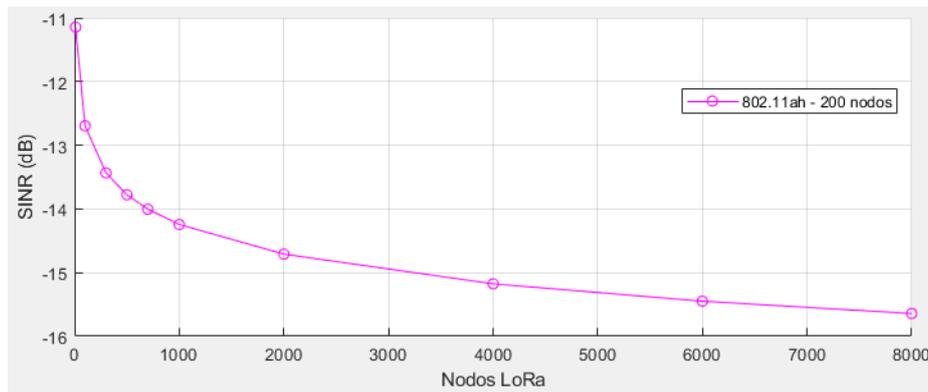


Figura 117. Gráfica logarítmica del SINR para 200 nodos de IEEE 802.11ah.  
Fuente: Autor.

### Todas las variaciones de nodos de IEEE 802.11ah

Al reunir todas las variaciones en una misma gráfica lineal dentro de la Figura 118, se puede notar con mayor facilidad las similitudes entre cada una de ellas. Salta a la vista el parecido entre las variaciones de 50 (azul) y 200 (violeta) nodos de IEEE 802.11ah, ambas con un comportamiento similar pero con valores de SINR menores. Además, se identifica una relación entre la cantidad de estaciones IEEE 802.11ah y el SINR, siendo este último menor mientras se incrementa la cantidad de nodos. Un SINR negativo en todas las iteraciones demuestra que la energía acumulada tanto de la

interferencia de LoRa como del Ruido blanco es mayor (y por mucho) a la energía de la señal deseada.

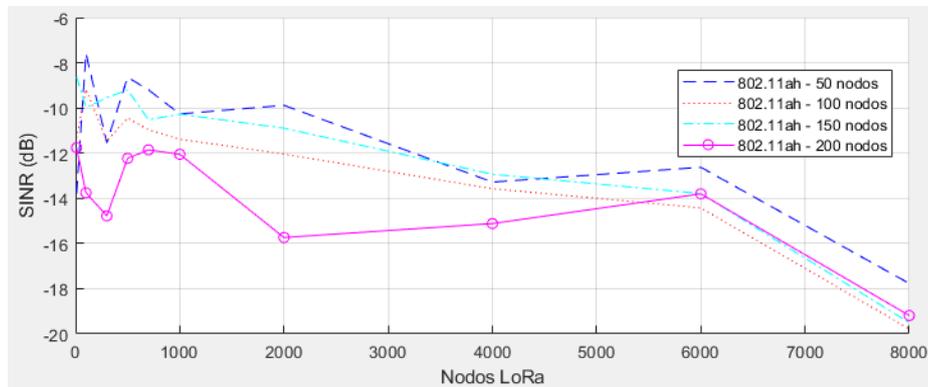


Figura 118. Gráfica lineal del SINR para todas las variaciones de IEEE 802.11ah.  
Fuente: Autor.

El comportamiento descrito anteriormente es más fácil de notar en la Figura 119, donde la línea punteada azul es la que presenta la menor inclinación, lo que corresponde con la menor cantidad de nodos de IEEE 802.11ah (50), mientras que la curva violeta es la que mayor inclinación tiene y a su vez representa a la mayor cantidad de estaciones (200) de la tecnología analizada.

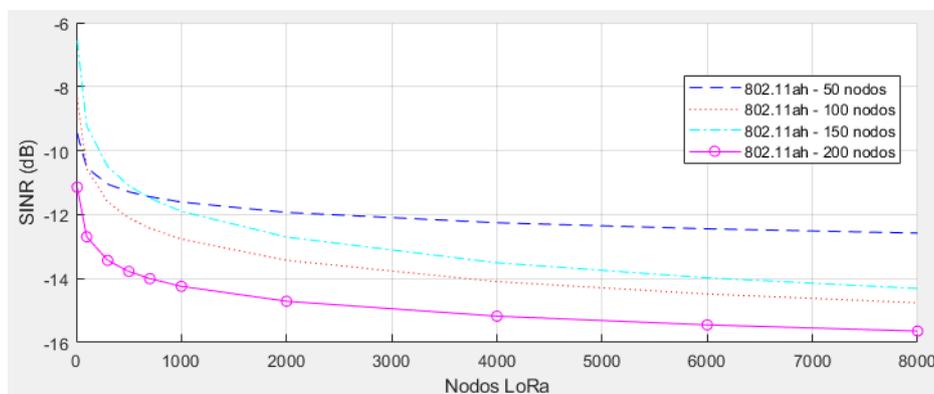


Figura 119. Gráfica logarítmica del SINR para todas las variaciones de IEEE 802.11ah.  
Fuente: Autor.

### Variación del radio de LoRa

La gráfica del SINR vs el radio de la topología de LoRa en la Figura 120 refleja una disminución significativa de la relación señal-interferencia + ruido cada vez que se aumenta el radio máximo de distribución de las estaciones de LoRa. El SINR promedio

de todas las iteraciones han sido negativos, fenómeno que indica una mayor potencia de la señal interferente por sobre la señal deseada.

Valores de hasta -40 dB de SINR pueden ser devastadores para una red, sin embargo si son comparados con el porcentaje de paquetes perdidos de la Figura 106, este no supera el 14% en su máximo punto, esto se debe a que el SINR es registrado únicamente en los casos en los que se ha detectado un solapamiento (ver Capítulo III - Sección 3.3) y tomando en consideración el funcionamiento de NS-3 en lo que respecta a subportadoras (ver Capítulo 4 – Sección 4.1) es factible concluir que la interferencia debido a LoRa no afecta a todas las subportadoras del símbolo OFDM de IEEE 802.11ah, para ampliar esta explicación se recomienda analizar la Figura 76 y su información asociada.

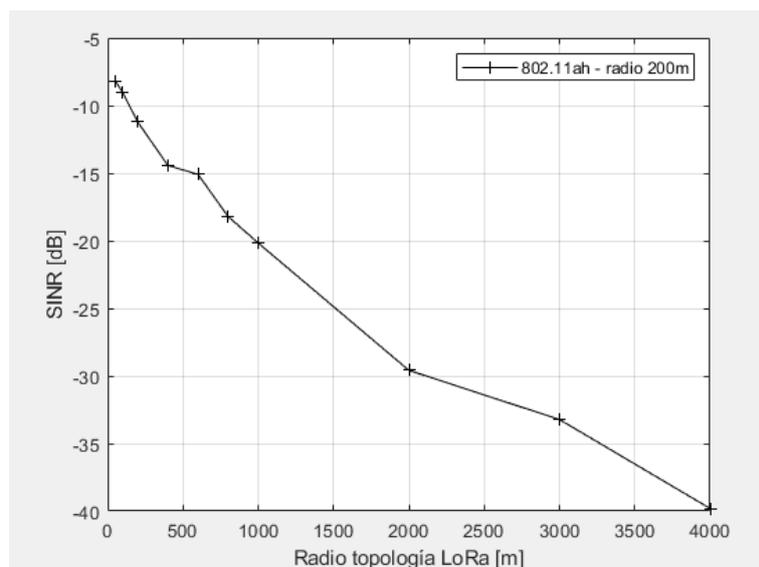


Figura 120. Gráfica lineal del SINR para 100 nodos de IEEE 802.11ah.  
Fuente: Autor.

### Variación del Payload de IEEE 802.11ah

Para mantener una congruencia en el análisis de los resultados, se han reunido los valores de SINR obtenidos de todas las iteraciones correspondientes al aumento de la longitud del Payload, en la Figura 121.

Resulta particular el comportamiento de esta métrica, destacando que no existe una variación considerable en el SINR calculado. Sin embargo sí que existe una ligera tendencia al aumento del SINR, esto se debe a que las estaciones de IEEE 802.11ah realizan un menor número de transmisiones conforme aumenta el Payload, esto se traduce en una reducción de la probabilidad de existir un solapamiento y por consecuencia una interferencia con LoRa.

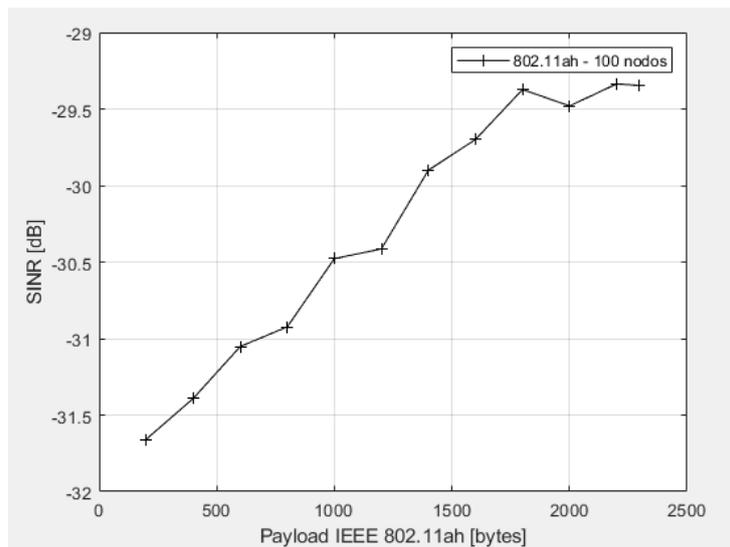


Figura 121. Gráfica lineal del SINR para 100 nodos de IEEE 802.11ah, variando el Payload.  
Fuente: Autor.

### 4.3 CONSIDERACIONES ADICIONALES

Durante el análisis de resultados se señaló que el comportamiento de algunas gráficas se debía a como está definido el cálculo del SINR y su relación con el tipo de solapamiento, pudiendo ser este completo o parcial. De manera que, en esta sección final se profundizará sobre dicho comportamiento y se lo justificará en base a datos extraídos de simulaciones particulares, independientes de las ya planteadas en la Sección 3.5.

La primera simulación particular que se ejecutó corresponde a 50 nodos de IEEE 802.11ah y 10 nodos de LoRa, sus resultados son reflejados en la Tabla 21 y muestran a los casos 1 y 2 como aquellos que acumulan la mayor cantidad de coincidencias (con 73.58% y 26.28% respectivamente). No obstante, estos casos no indican solapamiento alguno (ver Sección 3.3). Los que si señalan un solapamiento parcial son los casos 4 y 5, de los que solamente el segundo cuenta con coincidencias dentro de la tabla. Al considerar la totalidad de los solapamientos (parcial y completo), se determina que el caso 5 cuenta con el 90.97% de todos los solapamientos detectados. Este porcentaje indica una predominancia del solapamiento completo, que es aquel que trabaja de manera exacta con el umbral establecido (16 dB).

Tabla 21. Cantidad de solapamientos por casos para 50 nodos de IEEE 802.11ah y 10 nodos de LoRa.

<b>Casos</b>	<b>Cantidad</b>	<b>Porcentaje</b>
6	50	9.03
5	504	90.97
4	0	-
3	0	-
2	108340	No Solapa
1	303314	No Solapa
<b>TOTAL</b>	<b>412208</b>	<b>100</b>

El porcentaje restante (8.65%) corresponde al caso de solapamiento parcial, que tal como se había mencionado anteriormente, sobreestima al SINR. Esto se produce ya

que, el divisor (Interferencia + Ruido) es mucho menor en comparación al dividendo (Señal), debido al porcentaje reducido de solapamiento (en relación con el 100% de un solapamiento completo) durante el cálculo de la energía de interferencia (*potencia en Watts \* duración*), dando como resultado una amplificación del SINR.

Considerando que la cantidad de solapamientos depende del número de estaciones de LoRa, se realizaron más simulaciones variando su parámetro de nodos. La Tabla 22 organiza los datos obtenidos al incrementar hasta 100 las STAs de LoRa, manteniendo el resto de los parámetros sin cambios. Lo primero que se puede notar es, que el caso de solapamiento parcial 4 si tiene coincidencias, a diferencia de lo visto en la Tabla 21. Por otro lado, el caso 5 se repite como aquel que mayor cantidad de matches presenta, dando como resultado un 91.35% de todos los solapamientos detectados (contando los casos 3, 4, 5 y 6).

Tabla 22. Cantidad de solapamientos por casos para 50 nodos de IEEE 802.11ah y 100 nodos de LoRa.

<b>Casos</b>	<b>Cantidad</b>	<b>Porcentaje</b>
6	9	0.7
5	1173	91.35
4	102	7.95
3	0	-
2	79514	No Solapa
1	967778	No Solapa
<b>TOTAL</b>	<b>1048576</b>	<b>100</b>

Se continuó aumentando el número de estaciones de LoRa hasta 200 nodos y de esta simulación se generó la Tabla 23, cuyos resultados se alinean con los obtenidos en las anteriores simulaciones de esta sección, en donde la mayor cantidad de coincidencias

se agrupan en el caso 5, seguido del caso 4 y caso 6. El caso 5 se mantiene como aquel que mayor porcentaje de solapamientos representa, con un total del 96.88%.

Tabla 23. Cantidad de solapamientos por casos para 50 nodos de IEEE 802.11ah y 200 nodos de LoRa.

<b>Casos</b>	<b>Cantidad</b>	<b>Porcentaje</b>
6	20	0.89
5	2174	96.88
4	50	2.23
3	0	-
2	71490	No Solapa
1	974842	No Solapa
<b>TOTAL</b>	<b>1048576</b>	<b>100</b>

Finalmente, se cambió la estrategia de simulación para aumentar la cantidad de nodos de IEEE 802.11ah hasta 100, mientras se mantiene la cantidad de estaciones de LoRa en 10. El porcentaje de casos de solapamiento completo en los resultados obtenidos de la Tabla 24, corresponden al 95.99% de todos los solapamientos detectados.

Tabla 24. Cantidad de solapamientos por casos para 100 nodos de IEEE 802.11ah y 10 nodos de LoRa.

<b>Casos</b>	<b>Cantidad</b>	<b>Porcentaje</b>
6	100	4.01
5	2397	95.99
4	0	-
3	0	-
2	371288	No Solapa
1	674791	No Solapa
<b>TOTAL</b>	<b>1048576</b>	<b>100</b>

Con estas cuatro simulaciones particulares analizadas y centrándose específicamente en la cantidad de solapamientos de cada caso, se identificó un comportamiento claro, en donde el mayor porcentaje de solapamientos son de tipo completo, es decir, que la señal de LoRa interfiere de inicio a fin a la señal de IEEE 802.11ah. Si bien los porcentajes varían según los parámetros asignados, es posible decir con seguridad que la cantidad de solapamientos completos supera fácilmente el 90% de

todos los solapamientos. Esto se traduce en una validez del modelo utilizado para calcular el SINR, a partir de las energías de la señal, la interferencia y del ruido.

## CONCLUSIONES

A través de la investigación de distintas bibliografías fue posible una comprensión del funcionamiento en las capas PHY y MAC de las tecnologías IEEE 802.11ah y LoRa, así como de sus mecanismos tan característicos que las distinguen de otras tecnologías sub-GHz. Gracias a los datos extraídos de los documentos oficiales de cada estándar, se definieron los parámetros técnicos para la simulación de cada uno de los módulos en NS-3.

Aprovechando la capacidad de simulación de tecnologías sub-GHz en NS-3, fue posible superar la limitante de contar con una gran cantidad de equipos físicos para analizar el comportamiento de redes densas. Del mismo modo, por medio de las contribuciones públicas en GitHub se profundizó en la línea base de los protocolos hasta desarrollar un modelo de simulación que contabilice la interferencia de LoRa en IEEE 802.11ah.

Como parte de las modificaciones al módulo de LoRa dentro de NS-3, se consiguió que este opere exitosamente en la banda de 902 MHz a 928 MHz, que corresponde a las frecuencias utilizadas dentro del continente americano. Habiendo logrado simular LoRa en la frecuencia deseada, se realizaron también, cambios en el módulo de IEEE 802.11ah para procesar los datos de interferencia extraídos de las simulaciones de LoRa.

Fue necesario ir más allá de la investigación bibliográfica realizada y profundizar en los trabajos académicos relacionados con el BER, PER y SNR/SINR, a fin de definir un umbral adecuado y con sustento para ser utilizado dentro de la simulación y establecer en qué condiciones la señal recibida por una estación IEEE 802.11ah es destruida por la interferencia de LoRa. En ese sentido, fue indispensable una simulación adicional en Matlab sobre el Packet Error Rate de una comunicación IEEE 802.11ah en un enlace

extremo a extremo, dando como resultado un umbral máximo de 16 dB para un PER de  $10^{-3}$ .

Durante el análisis de resultados, se identificaron valores de Throughput o PPP que aparentemente se salen de la tendencia, esto se produce debido al modelo utilizado para calcular el SINR, donde es considerada la duración del solapamiento para el cálculo de la energía de interferencia, esto provoca que los solapamientos de tipo parcial den como resultado un valor SINR amplificado, sin embargo, los solapamientos completos sí que dan valores de SINR correctos y son estos últimos aquellos que representan más del 90% de los casos de solapamiento dentro de las simulaciones.

Del análisis de los resultados obtenidos de la simulación de IEEE 802.11ah con las modificaciones realizadas se concluyó que existe una relación inversamente proporcional entre el número de estaciones de LoRa y el Throughput de la red IEEE 802.11ah, es decir, mientras mayor es la cantidad de nodos LoRa menor será el Throughput medido en la red, alcanzando una reducción en el Throughput de hasta el 30.95% cuando existen 8000 nodos de LoRa (con respecto a 0 nodos de LoRa). Por otra parte, el análisis del Porcentaje de Paquetes Perdidos (PPP) arrojó que existe una relación directamente proporcional entre la cantidad de estaciones de LoRa y el PPP de IEEE 802.11ah, llegando hasta un 79.10% de PPP en 8000 nodos de LoRa (en 200 estaciones de IEEE 802.11ah).

Mediante el análisis del Signal-to-Interference + Noise Ratio (SINR) se determinó que, tiende a ser menor conforme aumenta la cantidad de estaciones de ambas tecnologías. Sin embargo, pesar de alcanzar valores negativos de hasta -20 dB, no se produce una pérdida completa del paquete, puesto que gracias a la comprobación teórico-matemática realizada, se identificó que la interferencia de una señal LoRa afecta solamente a un máximo de 4 subportadoras dentro un símbolo OFDM de IEEE 802.11ah.

Para una red IEEE 802.11ah configurada con un MCS2 y con 100 estaciones, se obtuvo un Throughput de 290.787 Kbps, utilizando la máxima longitud de Payload disponible (2297 bytes) para dicho esquema. Siguiendo esta línea, el Porcentaje de Paquetes Perdidos presentó una tendencia al aumento que va de la mano con el Payload, alcanzando una pérdida máxima de 39.22%. Mientras que, el SINR se redujo en mínima medida (entre 1 y 2 dB) debido a una disminución en la cantidad de transmisiones de IEEE 802.11ah, relacionada con el aumento del Payload.

## RECOMENDACIONES

Los estándares de cada tecnología IoT definen mecanismos para la coexistencia, sin embargo, si fuese necesario solucionar problemas relacionados a interferencias inter-protocolo se recomienda analizar mecanismos como: el cambio de la frecuencia de operación, la utilización de beamforming o reprogramar las Ventanas de Acceso Restringido.

El tiempo total de simulación para cada iteración dependerá de la cantidad de nodos asignados a cada tecnología, pudiendo tomar desde un par de minutos hasta varias horas, para mejorar estos tiempos es recomendable aumentar la cantidad de procesadores y núcleos por procesador en la máquina virtual utilizada (asignar más de 2 Gb de RAM y 4 procesadores), se aconseja también limpiar la caché de la MV después de ejecutar simulaciones extensas, para evitar el volcado de memoria.

Al trabajar con el simulador de red NS-3, es recomendable primero identificar los módulos necesarios y sus dependencias, de manera que se compile únicamente los que vayan a utilizarse, de este modo se evitan problemas de incompatibilidad y a su vez se reducen los tiempos de simulación.

Gracias a que NS-3 trabaja bajo el lenguaje de programación C++, se recomienda utilizar las librerías `iostream` y `fstream` al momento de manipular datos en bruto y exportarlos hacia archivos de texto plano u hojas de cálculo.

Se recomienda trabajar de la mano con estándares de calidad, como es el caso del ANSI C63.27-2017 (American National Standard for Evaluation of Wireless Coexistence), del cual se extrajeron las métricas a ser analizadas (Throughput, PER, etc) y además, fue utilizado como guía durante los procesos de evaluación.

Como trabajo a futuro, se considera variar otros parámetros de la simulación de IEEE 802.11ah que no fueron tomados en cuenta en el presente trabajo. Como es el caso del

Esquema de Modulación y Codificación (MCS) o el radio de distribución de los nodos de la tecnología interferida. Incluso se podría considerar aumentar el número de iteraciones, a fin de obtener resultados más exactos y por ende gráficas más suavizadas.

## Bibliografía

- Adame, T., Bel, A., Bellata, B., Barcelo, J., & Oliver, M. (2014). IEEE 802.11AH: THE WIFI APPROACH FOR M2M COMMUNICATIONS. *IEEE Wireless Communications*, 144-153.
- Ahmed, N., Rahman, H., & Hussain, I. (08 de 2016). *A comparison of 802.11ah and 802.15.4 for IoT*. Obtenido de Researchgate: [https://www.researchgate.net/publication/305991052\\_A\\_comparison\\_of\\_80211ah\\_and\\_802154\\_for\\_IoT](https://www.researchgate.net/publication/305991052_A_comparison_of_80211ah_and_802154_for_IoT)
- Ahmed, N., Rahman, H., & Hussain, M. (2016). A comparison of 802.11ah and 802.15.4 for IoT. *ICT Express*.
- Al Homssi, B., Dakic, K., & Maselli, S. (2021). IoT Network Design using Open-Source LoRa Coverage Emulator. *IEEE Access*, 99.
- Baños González, V. H. (09 de 2020). Contributions to IEEE 802.11-based long range communications. *Universitat Politècnica de Catalunya*. España: Wireless Networks Group (WNG), Department of Network Engineering. Obtenido de <https://www.tesisenred.net/bitstream/handle/10803/670226/TVHBG1de1.pdf?sequence=1&isAllowed=y>
- Barany, J. (23 de Abril de 2019). *Barani Design*. Obtenido de LoRaWAN USA frequencies, channels and sub-bands for IoT devices: <https://www.baranidesign.com/faq-articles/2019/4/23/lorawan-usa-frequencies-channels-and-sub-bands-for-iot-devices>
- Bouguera , T., Diouris, J.-F., Chaillout, J.-J., Jaouadi, R., & Andrieux, G. (2018). Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN. *sensors*.
- Casals Ibáñez, L., Mir Masnou, B., Vidal Ferré, R., & Gomez, C. (2017). Modeling the energy performance of LoRaWAN. *Sensors*.

- Castells-Rufas, D., Galin-Pons, A., & Carrabina, J. (2018). The regulation of unlicensed sub-GHz bands: Are stronger restrictions required for LPWAN-based IoT success? *arXiv*.
- Chen, H. (2010). RLINK: A realistic simulation model of links in wireless sensor networks. *Journal of System Simulation*, 2967-2973.
- CISCO. (2020). *Cisco IoT*. Obtenido de <https://www.cisco.com/c/dam/en/us/products/collateral/se/internetof-things/at-a-glance-c45-731471.pdf>
- Committee LoRa Alliance Technical. (11 de 09 de 2017). LoRaWAN™ 1.1 Specification. Semtech.
- Contreras Higuera , M. (2012). Estimación del Valor Teórico para el Throughput en Redes LAN Basadas en Tecnología Power Line Communications Bajo el Estándar Homeplug 1.0. *Revista Especializada en Ingeniería Universidad Nacional Abierta y a Distancia*, 89-98.
- Dachyar, M., Zagloel, T., & Saragih, L. (2019). Knowledge growth and development: internet of things (IoT) research. *Heliyon*, 2006–2018.
- Erceg, V. (2004). IEEE P802.11 Wireless LANs, TGn Channel Models. *IEEE 802.11-03*.
- Fulop, S., & Fitz, K. (2006). Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications. *Journal of the Acoustical Society of America*, 360–371.
- Garcia-Villegas, E., & López-Aguilera, E. (2018). *Universitat Politècnica de Catalunya*. Obtenido de IEEE 802.11ah sub 1GHz WLAN for IoT: [https://upcommons.upc.edu/bitstream/handle/2117/182119/802.11ah\\_wi-fi\\_iot-5709.pdf](https://upcommons.upc.edu/bitstream/handle/2117/182119/802.11ah_wi-fi_iot-5709.pdf)

- Gemalto, Actility, & Semtech. (Febrero de 2017). *A WHITE PAPER PREPARED FOR THE LoRa ALLIANCE™*. Obtenido de LoRa Alliance: [https://loralliance.org/wp-content/uploads/2020/11/lorawan\\_security\\_whitepaper.pdf](https://loralliance.org/wp-content/uploads/2020/11/lorawan_security_whitepaper.pdf)
- Goursaud, C., & Gorce, J.-M. (Octubre de 2015). Dedicated networks for IoT: PHY / MAC state of the art and challenges. *EAI Endorsed Transactions on Internet of Things*.
- Hata, M. (1980). Empirical formula for propagation loss in mobile radio. *IEEE Transactions on Vehicular Technology*, 317-325.
- Hervas, M., Alsina-Pagès, R., Pijoan, J., & Salvador, M. (2014). Single-carrier frequency domain equalisation proposal for very long haul HF radio links. *Electronics Letters*, 50(17).
- IEEE 802.11 Working Group. (2016). Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt. New York, USA: ANSI/IEEE Std 802.11.
- Keysight. (2017). *RFMW*. Obtenido de <http://rfmw.em.keysight.com/wireless/helpfiles/n7617/Content/Main/802.11ah%20Channelization.htm>
- Lazarakis, F., Dangakis, K., & Kostarakis, P. (03 de 1997). Evaluation of a terrain-based point-to-point propagation model in the 900 MHz band. *International Journal of Communication Systems*, 10(2):65-71.
- Liu, Z., & Lv, P. (2020). RO-RAW: Run-Time Restricted Access Window Optimization in IEEE 802.11ah Network with Extended Kalman Filter. *Wireless Communications and Mobile Computing*, 1-20.

- LoRa Alliance. (2015). *LoRa Alliance*. Obtenido de A technical overview of LoRa® and LoRaWAN™: <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>
- Magrin, D. (2016). Network level performances of a LoRa system. Padua, Italia: Università degli Studi di Padova.
- Magrin, D., Capuzzo, M., & Zanella, A. (2020). A Thorough Study of LoRaWAN Performance Under Different Parameter Settings. *IEEE Internet of Things Journal*, 116-127.
- Marais, J., Malekian, R., & Abu-Mahfouz, A. (2017). LoRa and LoRaWAN testbeds: A review. *IEEE AFRICON* (págs. 1496-1501). Cape Town: IEEE AFRICON.
- Meera, M., & Rao, S. (2017). A Survey of the State of the Art of 802.11ah. *IEEE International Conference on Computational Intelligence and Computing Research (ICIC)* (págs. 1-4). Coimbatore: ICIC.
- NS-3. (18 de Julio de 2021). NS-3. Obtenido de ns3::SpectrumWifiPhy Class Reference: [https://www.nsnam.org/doxygen/classns3\\_1\\_1\\_spectrum\\_wifi\\_phy.html](https://www.nsnam.org/doxygen/classns3_1_1_spectrum_wifi_phy.html)
- Oukessou, Y., Baslam, M., & Oukessou, M. (2018). LPWAN IEEE 802.11ah and LoRaWAN capacity simulation analysis comparison using NS-3. *4th International Conference on Optimization and Applications (ICOA)* (págs. 1-4). Mohammedia.
- Panwar, D. (2017). NS3 : A quick overview.
- Perahia, E. (2008). IEEE 802.11n Development: History, Process, and Technology. *IEEE Communications Magazine*, 48-55.
- Perwej, Y., Omer, M., Sheta, O., Harb, H., & Adrees, M. (2019). The Future of Internet of Things (IoT) and Its Empowering Technology. 20192 - 20203.

- Ryan, P., & Watson, R. (2017). Research Challenges for the Internet of Things: What Role Can OR Play? *Systems*.
- Seber, G., & Wild, C. (1989). *Nonlinear Regression*. New York: John Wiley and Sons.
- Semtech. (2015). *LoRa Developers*. Obtenido de What are LoRa® and LoRaWAN®?: <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>
- Semtech Corporation. (02 de 05 de 2015). *AN1200.22 LoRa™ Modulation Basics*. Obtenido de <https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf>
- Sun, W., Cho, M., & Sunghyun, C. (2013). IEEE 802.11ah: A Long Range 802.11 WLAN at Sub 1 GHz. *Journal of ICT Standardization*, 83-108.
- Tian, L., Seferagic, A., Santi, S., De Poorter, E., Hoebeke, J., & Famaey, J. (2018). Extension of the IEEE 802.11ah ns-3 Simulation Module.
- Tustón Torres, I. (2011). Evaluación del rendimiento del Estándar IEEE 802.15.4 (ZIGBEE) en entornos de interferencia. Riobamba: Escuela Superior Politécnica de Chimborazo.
- Varga, A., & Hornig, R. (2008). An overview of the OMNeT++ simulation environment.
- Yuan, Z., Haiguang, W., Shoukang, Z., & Zander, Z. (2013). WLAN, Advances in IEEE 802.11ah Standardization Machine-Type Communications in Sub-1GHz. *IEEE International Conference on Communications 2013: IEEE ICC'13 - Second IEEE Workshop on Telecommunication Standards: From Research to Standards* (págs. 1269 - 1273). Singapore: IEEE International Conference on Communications.

## 6. ANEXOS

### ANEXO 1: Código para determinar el solapamiento de LoRa en IEEE 802.11ah y calcular el SINR.

```
//Importa el fichero de interferencia LoRa y verifica si se solapan
los eventos.
//st1 y en1 -> Start & End Time de la señal deseada.
//st2 y en2 -> Start & End Time de la señal interferente (LoRa).
Time st1 = event->GetStartTime (); //Tiempo inicial deseada
Time en1 = event->GetEndTime (); //Tiempo final deseada
Time st2; //Tiempo inicial interferencia
Time en2; //Tiempo final interferencia
Time overlap_1; //Tiempo que sobrelapan
Time limite = Seconds(9999); //Bandera para el fin del fichero
Time duration = event->GetDuration (); //Duración de la tx
ifstream myfile("./Recursos/interfLora.txt"); //Ruta fichero
double power; //Potencia interferencia
bool banderah = false; //Bandera para evento sobrelapamiento
double cumulativeInterfEnergy_lora = 0; //Almacena la energía de LoRa
int contBand = 0; //Contador interferencias
bool endEarly = false; //Bandera cuando no hay interferencia
double col3; //Almacena posición en X del nodo LoRa
double col4; //Almacena posición en Y del nodo LoRa
double col6; //Almacena frecuencia de la sub-banda
double m_exponent = 3.0; //Exponente del modelo Path Loss
double m_referenceDistance = 1.0; //Distancia de referencia 1m.
double m_frequency = 900e6; //Frecuencia de referencia en Hz
double m_referenceLoss = 30; //Pérdida de referencia en dB a 1m

if (myfile.is_open()) {
    myfile >> st2; //Tiempo inicial interferencia
    myfile >> en2; //Tiempo final interferencia
    myfile >> col3; //Posición en X de la STA LORA
    myfile >> col4; //Posición en y de la STA LORA
    myfile >> power; //TxPowerdBm nominal de LoRa
    myfile >> col6; //Frecuencia de la sub-banda

    while (st2!=limite) { //Evalúa el fichero hasta el límite
        myfile >> st2;
        myfile >> en2;
        myfile >> col3;
        myfile >> col4;
        myfile >> power;
        myfile >> col6;

        if (en2 <= st1) { //Evalúa si existe un solapamiento
            overlap_1 = Seconds (0);
            banderah = false;
        }
        else if (en1 <= st2) { //El evento antes de la interferencia
            overlap_1 = Seconds (0);
            banderah = false;
            break;
        }
        else if (st1 < st2) { //Señal deseada antes de interferencia
            if (en2 < en1) {
                overlap_1 = en2 - st2;
                banderah = true;
            }
        }
    }
}
```

```

        contBand++;
    } else {
        overlap_1 = en1 - st2;
        banderah = true;
        contBand++;
    }
} //FIN else-if
else { //Interferencia antes al mismo tiempo que deseada
    if (en1 < en2) {
        overlap_1 = en1 - st1;
        banderah = true;
        contBand++;
    } else {
        overlap_1 = en2 - st1;
        banderah = true;
        contBand++;
    }
} //FIN else

//Evalúa solamente si hubo un solapamiento, convierte la
//potencia de dBm a W, luego calcula la energía de la
//interferencia con la duración del solapamiento y la potencia
//en W. Luego hace una sumatoria de la energía de todas las
//interferencias.
if ( banderah ) {
    //Determina la posición de la STA 802.11ah que está en rx
    //Con las posiciones de ambos nodos se calcula la distancia.
    Ptr<MobilityModel> rxMobility = this->GetMobility ()-
>GetObject<MobilityModel> ();
    Vector positionRx = rxMobility->GetPosition ();
    double distance = sqrt(pow(positionRx.x - (col3), 2) +
pow(positionRx.y - (col4), 2));

    //Cálculo del PathLoss con el modelo matemático
    //Log Distance.
    double pathLossdB = 10 * m_exponent * std::log10 (distance
/ m_referenceDistance) + 21 * std::log10 (col6 / 900e6);
    double rxc = pathLossdB - m_referenceLoss;
    double rxPowerDbm_lora = power + rxc;

    double interfPowerW_lora = pow (10, rxPowerDbm_lora / 10)
/ 1000;
    double interfEnergy_lora = overlap_1.GetSeconds () *
interfPowerW_lora;
    cumulativeInterfEnergy_lora += interfEnergy_lora;

} //FIN Bandera
} //FIN while

//Calcula la Energía de la señal deseada a partir de su potencia
//en W y de la duración de la señal, y SNIR.
double signalPowerW_1 = event->GetRxPowerW ();
double signalEnergy_1 = duration.GetSeconds () * signalPowerW_1;
double ruidoTermico = 3.4517500038e-16 * duration.GetSeconds();
//Ruido térmico a 200°K y con 125 kHz de AB
double sinr_lora = 10 * log10 (signalEnergy_1 /
(cumulativeInterfEnergy_lora + ruidoTermico));

//Si existió al menos un solapamiento se imprime la potencia de la
//señal, la energía y el SINR.
if ( contBand != 0 ) {

```

```

ofstream interfInfo;
string rutaInfo = "./Resultados/interfInfo.txt";
interfInfo.open(rutaInfo.c_str(), ios::out | ios::app);
interfInfo << sinr_lora << std::endl;
interfInfo.close();

//Compara el nivel mínimo de SINR y termina la comunicación.
if ( sinr_lora < 16 ) {
    NotifyRxDrop (packet); //Dropea el paquete
    m_state->SwitchFromRxEndError (packet, sinr_lora);
//Cambia el estado de la STA
    endEarly = true;
//Bandera para terminar antes el resto del proceso
} else {
    endEarly = false; //Bandera para no terminar el proceso
}
} //FIN contBand

} //FIN - my_file_is_open

```

## ANEXO 2: Código que modifica el módulo de LoRa para operar en la banda 902-928 MHz.

```

// ns-3/src/lorawan/helper/lorawan-mac-helper.cc
//Método para configurar la región US - end device
void
LorawanMacHelper::ConfigureForUSRegion (Ptr<ClassAEndDeviceLorawanMac>
edMac) const
{
    NS_LOG_FUNCTION_NOARGS ();

    ApplyCommonUSConfigurations (edMac);

    ///////////////////////////////////////////////////////////////////
    // TxPower -> Transmission power in dBm conversion //
    ///////////////////////////////////////////////////////////////////
    edMac->SetTxDbmForTxPower (std::vector<double>{16, 14, 12, 10, 8, 6,
4, 2});

    ///////////////////////////////////////////////////////////////////
    // Matrix to know which DataRate the GW will respond with //
    ///////////////////////////////////////////////////////////////////
    LorawanMac::ReplyDataRateMatrix matrix = {{{{0, 0, 0, 0, 0, 0}},
                                                {{1, 0, 0, 0, 0, 0}},
                                                {{2, 1, 0, 0, 0, 0}},
                                                {{3, 2, 1, 0, 0, 0}},
                                                {{4, 3, 2, 1, 0, 0}},
                                                {{5, 4, 3, 2, 1, 0}},
                                                {{6, 5, 4, 3, 2, 1}},
                                                {{7, 6, 5, 4, 3, 2}}}}};

    edMac->SetReplyDataRateMatrix (matrix);

    ///////////////////////////////////////////////////////////////////
    // Preamble length //
    ///////////////////////////////////////////////////////////////////
    edMac->SetNPreambleSymbols (8);

    ///////////////////////////////////////////////////////////////////
    // Second receive window parameters //

```

```

////////////////////////////////////
edMac->SetSecondReceiveWindowDataRate (0);
edMac->SetSecondReceiveWindowFrequency (923.30);
}
//

//Método para configurar la región US - gateway
void
LorawanMacHelper::ConfigureForUSRegion (Ptr<GatewayLorawanMac> gwMac)
const
{
    NS_LOG_FUNCTION_NOARGS ();

    //////////////////////////////////
    // ReceivePath configuration //
    //////////////////////////////////
    Ptr<GatewayLoraPhy> gwPhy =
        gwMac->GetDevice ()->GetObject<LoraNetDevice> ()->GetPhy ()-
>GetObject<GatewayLoraPhy> ();

    ApplyCommonUSConfigurations (gwMac);

    if (gwPhy) // If cast is successful, there's a GatewayLoraPhy
    {
        NS_LOG_DEBUG ("Resetting reception paths");
        gwPhy->ResetReceptionPaths ();

        std::vector<double> frequencies;

        frequencies.push_back (902.3);
        frequencies.push_back (902.5);
        frequencies.push_back (902.7);

        for (auto &f : frequencies)
        {
            gwPhy->AddFrequency (f);
        }

        int receptionPaths = 0;
        int maxReceptionPaths = 8;
        while (receptionPaths < maxReceptionPaths)
        {
            gwPhy->GetObject<GatewayLoraPhy> ()->AddReceptionPath ();
            receptionPaths++;
        }
    }
}
//

//Aplicar configuraciones para región US
void
LorawanMacHelper::ApplyCommonUSConfigurations (Ptr<LorawanMac>
lorawanMac) const
{
    NS_LOG_FUNCTION_NOARGS ();

    //////////////////////////////////
    // SubBands //
    //////////////////////////////////

    LogicalLoraChannelHelper channelHelper;

```

```

channelHelper.AddSubBand (902.2, 903.7, 0.01, 14);
channelHelper.AddSubBand (903.9, 905.3, 0.01, 14);
channelHelper.AddSubBand (905.5, 906.9, 0.01, 27);

////////////////////////////////////
// Default channels //
////////////////////////////////////
Ptr<LogicalLoraChannel> lc1 = CreateObject<LogicalLoraChannel>
(902.3, 0, 5);
Ptr<LogicalLoraChannel> lc2 = CreateObject<LogicalLoraChannel>
(902.5, 0, 5);
Ptr<LogicalLoraChannel> lc3 = CreateObject<LogicalLoraChannel>
(902.7, 0, 5);
channelHelper.AddChannel (lc1);
channelHelper.AddChannel (lc2);
channelHelper.AddChannel (lc3);

lorawanMac->SetLogicalLoraChannelHelper (channelHelper);

////////////////////////////////////
// DataRate -> SF, DataRate -> Bandwidth //
// and DataRate -> MaxAppPayload conversions //
////////////////////////////////////
lorawanMac->SetSfForDataRate (std::vector<uint8_t>{12, 11, 10, 9, 8,
7, 7});
lorawanMac->SetBandwidthForDataRate (
    std::vector<double>{125000, 125000, 125000, 125000, 125000,
125000, 250000});
lorawanMac->SetMaxAppPayloadForDataRate (
    std::vector<uint32_t>{59, 59, 59, 123, 230, 230, 230, 230});
}
//

```

### ANEXO 3: Códigos para generar el fichero de interferencia LoRa.

```

// ns-3/src/lorawan/model/lora-interference-helper.cc
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

NS_LOG_FUNCTION (this << event);
NS_LOG_INFO ("Current number of events in LoraInterferenceHelper: " <<
m_events.size ());
// We want to see the interference affecting this event: cycle through
// events that overlap with this one and see whether it survives the
// interference or not.

// Imprimir información de interferencia
ofstream interfInfo;
string rutaInfo = "./Resultados/interfLora_temp1.txt";
interfInfo.open(rutaInfo.c_str(), ios::out | ios::app);
interfInfo << event->GetStartTime () << " " << event->GetEndTime () <<
std::endl;
interfInfo.close ();

```

```

// ns-3/src/lorawan/model/simple-end-device-lora-phy.cc
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

//Con el mobilityModel obtengo la posicion del sender.
Ptr<MobilityModel> senderMobility = this->GetMobility ()-
>GetObject<MobilityModel> ();
Vector position = senderMobility->GetPosition ();
double x = position.x + 200;
double y = position.y + 200;

//Extrae Potencia de Tx para comprobar con el datasheet.
ofstream temp1;
string rutaTemp1 = "./Resultados/interfLora_temp2.txt";
temp1.open(rutaTemp1.c_str(), ios::out | ios::app);
//temp1 << this << " " << x << " " << y << " " << txPowerdBm << " "
<< duration << " " << frequencyMHz << std::endl;
temp1 << x << " " << y << " " << txPowerdBm << " " << frequencyMHz <<
std::endl;
temp1.close();
//

// ns-3/src/lorawan/examples/complete-network-example.cc
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int counta = 0;
string line;

//Determina cuantas entradas existen en los temporales.
ifstream file("./Resultados/interfLora_temp1.txt");
while (getline(file, line)) {
    counta++;
}
cout << "Número de lineas en el fichero: " << counta << endl;

//Lee los ficheros temporales y crea el fichero final.
ifstream myfile1("./Resultados/interfLora_temp1.txt"); //Ruta fichero
ifstream myfile2("./Resultados/interfLora_temp2.txt"); //Ruta fichero
Time col1; //Tiempo inicial
Time col2; //Tiempo final
string col3; //Sender posición x
string col4; //Sender posición y
string col5; //Sender TxPowerdBm
string col6; //Frecuencia usada

ofstream interfL;
string rutaInterfL = "./Resultados/interfLora.txt";
interfL.open(rutaInterfL.c_str(), ios::out | ios::app);

if (myfile1.is_open()) {
    myfile1 >> col1; //Almacena un valor de la 1era columna
    myfile1 >> col2; //2da columna

    if (myfile2.is_open()) {
        myfile2 >> col3; //3ra columna
    }
}

```

```

myfile2 >> col4; //4ta columna
myfile2 >> col5; //5ta columna
myfile2 >> col6; //6ta columna

ifstream file("./Resultados/interfLora_templ.txt");
while (getline(file, line)) {
    myfile1 >> col1;
    myfile1 >> col2;
    myfile2 >> col3;
    myfile2 >> col4;
    myfile2 >> col5;
    myfile2 >> col6;
    interfL << "+" << col1.GetNanoSeconds() << ".0ns +" <<
col2.GetNanoSeconds() << ".0ns " << col3 << " " << col4 << " " << col5
<< " " << col6 << std::endl;
}

interfL.close();
myfile1.close();
myfile2.close();
} //FIN if 2
} //FIN if 1

```

#### **ANEXO 4: Código del ejecutable para contralar las iteraciones de ambos modelos de simulación.**

```

#!/bin/bash
# SCRIPT PARA AUTOMATIZAR LA SIMULACIÓN

#####
#####
# Receptar parámetros por teclado.
cd ~
echo "Bienvenido al asistente de Simulación"

echo "Por favor ingrese los parámetros solicitados para IEEE 802.11ah: "
"
echo -e "Número de nodos: "
read nodos_ah
echo -e "Radio en metros: "
read radio_ah
echo -e "Tamaño del Payload en bytes: "
read payload_ah
echo -e "Tiempo en segundos: "
read tiempo_ah
echo "Cargando..."
echo ""

echo "Por favor ingrese los parámetros solicitados para LoRa: "
echo -e "Número de nodos: "
read nodos_lora
echo -e "Radio en metros: "
read radio_lora
echo -e "Tiempo en segundos: "
read tiempo_lora

#####
#####
# Realizar los cambios de acuerdo con las variables.

```

```

echo "Configurando parámetros..."
##### IEEE 802.11ah #####
# Nodos
case $nodos_ah in
50)
    sed -i "87c\    string RAWConfigFile =
\./OptimalRawGroup/RawConfig-50.txt\";" IEEE802.11ah/IEEE-802.11ah-
ns-3/scratch/test/Configuration.h
    ;;
100)
    sed -i "87c\    string RAWConfigFile =
\./OptimalRawGroup/RawConfig-100.txt\";" IEEE802.11ah/IEEE-802.11ah-
ns-3/scratch/test/Configuration.h
    ;;
150)
    sed -i "87c\    string RAWConfigFile =
\./OptimalRawGroup/RawConfig-150.txt\";" IEEE802.11ah/IEEE-802.11ah-
ns-3/scratch/test/Configuration.h
    ;;
175)
    sed -i "87c\    string RAWConfigFile =
\./OptimalRawGroup/RawConfig-175.txt\";" IEEE802.11ah/IEEE-802.11ah-
ns-3/scratch/test/Configuration.h
    ;;
200)
    sed -i "87c\    string RAWConfigFile =
\./OptimalRawGroup/RawConfig-200.txt\";" IEEE802.11ah/IEEE-802.11ah-
ns-3/scratch/test/Configuration.h
    ;;
250)
    sed -i "87c\    string RAWConfigFile =
\./OptimalRawGroup/RawConfig-250.txt\";" IEEE802.11ah/IEEE-802.11ah-
ns-3/scratch/test/Configuration.h
    ;;
300)
    sed -i "87c\    string RAWConfigFile =
\./OptimalRawGroup/RawConfig-300.txt\";" IEEE802.11ah/IEEE-802.11ah-
ns-3/scratch/test/Configuration.h
    ;;
esac

# Radio
sed -i "67c\    string rho=\"\$radio_ah\"; //50 " IEEE802.11ah/IEEE-
802.11ah-ns-3/scratch/test/Configuration.h

##### LoRa #####
# Nodos
sed -i "43c\int nDevices = $nodos_lora;" LORAWAN/ns-
3/src/lorawan/examples/complete-network-example.cc

# Radio
sed -i "45c\double radius = $radio_lora; //7500 metros" LORAWAN/ns-
3/src/lorawan/examples/complete-network-example.cc

# Tiempo
sed -i "46c\double simulationTime = $tiempo_lora; //600 segundos"
LORAWAN/ns-3/src/lorawan/examples/complete-network-example.cc

#####
#####
# Ejecución de las simulaciones.

```

```

echo ""
echo "Iniciando simulación de LoRa..."

# Ejecución de LoRa
cd ~
cd LORAWAN/ns-3/
./waf --run complete-network-example

# Copiar el archivo de interferencia LoRa hacia la carpeta Recursos de
IEEE 802.11ah
cd ~
rm IEEE802.11ah/IEEE-802.11ah-ns-3/Recursos/interfLora.txt
cp LORAWAN/ns-3/Resultados/interfLora.txt IEEE802.11ah/IEEE-802.11ah-
ns-3/Recursos/interfLora.txt
lineas=$(wc -l < LORAWAN/ns-3/Resultados/interfLora.txt)
sed -i "$lineas a 9999 9999 9999 9999 9999 9999" IEEE802.11ah/IEEE-
802.11ah-ns-3/Recursos/interfLora.txt
echo " "
echo "*****"
echo "Simulación de LoRa terminada correctamente"

echo ""
echo "Iniciando simulación de IEEE 802.11ah..."
# El bash espera 3 segundos antes de ejecutar el siguiente comando
sleep 3

# Payload y Tiempo van directo en la ejecución del waf de IEEE
802.11ah.
cd ~
cd IEEE802.11ah/IEEE-802.11ah-ns-3/
#./waf --run test
./waf --run "test --seed=1 --simulationTime=$tiempo_ah --
payloadSize=$payload_ah"
echo " "
echo "*****"
echo "Simulación de IEEE 802.11ah terminada correctamente"

sleep 1

#####
#####
# Preparación del archivo de la iteración actual
cd ~
ORIGINAL="IEEE802.11ah/IEEE-802.11ah-ns-3/Resultados/iteracion-
beta.txt"
COPIA="Iteraciones/AH_nodos$nodos_ah-radio$radio_ah-
payload$payload_ah-tiempo$tiempo_ah-LORAnodos$nodos_lora-
radio$radio_lora-payload$payload_lora-tiempo$tiempo_lora.txt"
cp $ORIGINAL $COPIA

sed -i "2 a CantidadNodosLORA RadioLORA SimTimeLORA" $COPIA
sed -i "3 a $nodos_lora $radio_lora $tiempo_lora" $COPIA
ORIGINAL1="IEEE802.11ah/IEEE-802.11ah-ns-3/Resultados/interfInfo.txt"
COPIA1="Iteraciones/SNIR-AH_nodos$nodos_ah-radio$radio_ah-
payload$payload_ah-tiempo$tiempo_ah-LORAnodos$nodos_lora-
radio$radio_lora-payload$payload_lora-tiempo$tiempo_lora.txt"
cp $ORIGINAL1 $COPIA1

lineas1=$(wc -l < $COPIA1)
lineas1=$((lineas1-1))
sed -i "1c\ $lineas1" $COPIA1

```

```

echo " "
echo "Sus resultados han sido guardados en la ruta
/home/mateo/Iteraciones con el siguiente nombre: "
echo " AH_nodos$nodos_ah-radio$radio_ah-payload$payload_ah-
tiempo$tiempo_ah-LORAnodos$nodos_lora-radio$radio_lora-
tiempo$tiempo_lora.txt"
echo ""
echo "Proceso Terminado Exitosamente!"

```

## ANEXO 5: Código de Matlab de la modificación del ejemplo PERvsSINR para la simulación de IEEE 802.11ah.

```

%% Waveform Configuration
% Create S1G configuration object for single user S1G short preamble
% transmission with 2 transmit antennas and 2 space-time streams
cfgS1G = wlanS1GConfig;
cfgS1G.ChannelBandwidth = 'CBW2'; % 2 MHz channel bandwidth
cfgS1G.Preamble = 'Short'; % Short preamble
cfgS1G.NumTransmitAntennas = 1; % 2 transmit antennas - 2
cfgS1G.NumSpaceTimeStreams = 1; % 2 space-time streams - 2
cfgS1G.APEPLength = 100; % APEP length in bytes - 256
cfgS1G.MCS = 2; % 64-QAM rate-5/6 - 7

%% Channel Configuration
% Create and configure the TGah channel
tgahChannel = wlanTGahChannel;
tgahChannel.DelayProfile = 'Model-B';
tgahChannel.NumTransmitAntennas = cfgS1G.NumTransmitAntennas;
tgahChannel.NumReceiveAntennas = 1; % 2
tgahChannel.TransmitReceiveDistance = 5; % Distance in meters for NLOS
tgahChannel.ChannelBandwidth = cfgS1G.ChannelBandwidth;
tgahChannel.LargeScaleFadingEffect = 'None';

%% Simulation Parameters
snr = 15:1:20; % snr = 25:10:45;

%%
% The number of packets tested at each SNR point is controlled by two
% parameters:

maxNumErrors = 1e3; % The maximum number of packet errors at an SNR
point - 1e2
maxNumPackets = 1e4; % Maximum number of packets at an SNR point - 1e3

```

## ANEXO 6: Configuraciones de RAW para cada iteración de la simulación de IEEE 802.11ah en NS-3.

50 nodos

1							
1							
0	1	1	829	1	0	1	50

100 nodos

<hr/>							
2							
<hr/>							
4							
<hr/>							
0	1	1	209	1	0	1	14
<hr/>							
0	1	1	209	1	0	15	29
<hr/>							
0	1	1	209	1	0	30	44
<hr/>							
0	1	1	209	1	0	45	50
<hr/>							
4							
<hr/>							
0	1	1	209	1	0	51	64
<hr/>							
0	1	1	209	1	0	65	79
<hr/>							
0	1	1	209	1	0	80	94
<hr/>							
0	1	1	209	1	0	95	100
<hr/>							

150 nodos

<hr/>							
2							
<hr/>							
1							
<hr/>							
0	1	1	849	1	0	1	127
<hr/>							
1							
<hr/>							
0	1	1	849	1	0	128	150
<hr/>							

200 nodos

<hr/>							
4							
<hr/>							
1							
<hr/>							
0	1	1	849	1	0	1	127
<hr/>							
1							
<hr/>							
0	1	1	849	1	0	128	150
<hr/>							
1							
<hr/>							
0	1	1	849	1	0	151	177
<hr/>							
1							
<hr/>							
0	1	1	849	1	0	178	200
<hr/>							