

Tesis Doctoral

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Universidad Tecnológica Nacional

Facultad Regional Buenos Aires

Doctorado en Ingeniería

Mención Procesamiento de Señales e Imágenes

Ing. Patricio Latini

Dr. Claudio Righetti

DEDICATORIA

A Sara que siempre está incondicionalmente, a mamá que siempre estuvo orgullosa de mis logros y le faltó muy poco para llegar a ver éste, a mis amigos y a mi familia

AGRADECIMIENTOS

A mi director de tesis, Doctor Claudio Righetti, por la confianza y el apoyo en la dirección de esta tesis.

Al Dr. Ricardo Armentano, por inspirarme a comenzar este proceso de doctorado. Y siempre estar presente para cualquier tipo de consulta.

Al plantel de profesores del doctorado, particularmente al Dr. Walter Legnani, al Dr. Claudio Delrieux y al Dr. Pablo Ristori por su guía y apoyo durante este proceso más allá de los contenidos impartidos

Al grupo de apoyo del doctorado, a Edith Ricciolini y Carolina Rodrigo por siempre ayudar con la mejor predisposición en todos los procesos de este doctorado.

Un especial recuerdo y agradecimiento a Diego Sica que fue un gran apoyo en mi primer parte del doctorado.

A los jurados de esta tesis, Dr. José Ignacio Alvarez-Hamelin, Dr. Esteban Mocskos y Dr. Edgardo Comas, por haber aceptado la responsabilidad y dedicar su tiempo y esfuerzo a la revisión de este trabajo.

A los profesores Ing. Alejandro Furfaro e Ing. Marcelo Doallo, ambos exdirectores del departamento de electrónica de la UTN-FRBA por haberme apoyado en este proceso.

A mi compañero de estudios del doctorado, Dr. Cristian Bonini por su ayuda durante la carrera.

A mi *alma máter*, la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional por permitirme continuar en casa con la carrera científica.

A mis compañeros y supervisores de las empresas Intraway y CASA Systems que colaboraron con recursos y tiempo para la realización de los trabajos experimentales realizados en esta tesis.

Finalmente, un agradecimiento especial a Sara, mi esposa, por el apoyo y sus consejos durante la ejecución de esta tesis.

TABLA DE CONTENIDOS

Dedicatoria.....	i
Agradecimientos.....	iii
Lista de Tablas	xi
Lista de Figuras.....	xii
Lista de Ecuaciones.....	xvi
Resumen.....	1
1. Introducción	3
1.1. Contribuciones y limitaciones de la investigación.....	4
1.2. Hipótesis.....	5
1.3. Objetivo Principal.....	5
1.4. Objetivos Específicos	5
1.5. Metodología de investigación utilizada	5
1.6. Contenido y Estructura de la tesis.....	6
2. Internet de las Cosas.....	9
2.1. Reseña del capítulo	9
2.2. Introducción a Internet de las Cosas.....	9
2.3. Redes de área amplia de baja potencia (LPWAN)	11
2.4. Aplicaciones de Redes LPWAN.....	12
2.5. Banda licenciada vs. No Licenciada	18
2.6. Conclusiones	18
3. Protocolo LoRaWAN.....	21
3.1. Reseña del capítulo	21
3.2. Introducción	21
3.3. Elementos de una red LoRaWAN	23
3.3.1. Dispositivo Final.....	25
3.3.2. Gateway	25

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

3.3.3.	Servidor de Red	25
3.3.4.	Servidor de Aplicaciones.....	26
3.3.5.	Servidores de Union	26
3.4.	Fundamentos operacionales una red LoRaWAN	27
3.4.1.	Modulación fisica DSSS.....	28
3.4.2.	Corrección de errores en Lora	31
3.5.	Formato de la trama física LoRa	35
3.6.	Utilización del tiempo de aire en LoRa.....	36
3.7.	Clasificación de los Dispositivos LoRaWAN.....	39
3.8.	Conclusiones.....	41
4.	Análisis de capacidad de una red LoRaWAN.....	43
4.1.	Reseña del capítulo	43
4.2.	Efectos de ALOHA en una red LoRaWAN.....	43
4.3.	Planes de frecuencia	47
4.3.1.	Plan de Frecuencia EE. UU.	47
4.3.2.	Plan de Frecuencia Argentina y Australia	48
4.4.	Modelos de propagación de señales	49
4.5.	Modelado de escenarios.....	52
4.5.1.	Urbano denso con edificios de hormigón y sensores interiores	52
4.5.2.	Suburbano disperso con casas y sensores interiores.....	54
4.6.	Conclusiones.....	56
5.	Materiales de laboratorio y trabajo de campo.....	57
5.1.	Reseña del capítulo	57
5.2.	MicroPython.....	57
5.2.1.	Utilización de python.....	57

5.2.2.	Utilización de micropython.....	57
5.2.3.	Ventajas de usar Python en IoT	58
5.3.	Equipamiento LoRAWAN	59
5.3.1.	FiPy.....	59
5.3.2.	PySense	61
5.3.3.	PyTrack.....	62
5.3.4.	PyGate.....	63
5.4.	Equipamiento de medición de radiofrecuencia	65
5.4.1.	Software defined Radio (SDR)	65
5.4.2.	LimeSDR	68
5.5.	Herramientas de software	69
5.5.1.	Chirpstack	69
5.5.2.	InfluxDB (Base de Datos).....	72
5.5.3.	Grafana.....	73
5.5.4.	GNURADIO	74
5.6.	Conclusiones	75
6.	Métodos y resultados de ensayos	77
6.1.	Reseña del capítulo	77
6.2.	Introducción	77
6.3.	Configuración de los componentes	78
6.3.1.	Chirpstack	78
6.3.2.	Gateway – PyGate.....	85
6.3.3.	Sensor – PySense	86
6.3.4.	Sensor – PyTrack	88
6.3.5.	LimeSDR	90
6.3.6.	GNURadio	91
6.3.7.	Grafana.....	91

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

6.4. Experimentos	92
6.4.1. Laboratorio	93
6.4.2. Rural – Abierto	94
6.4.3. Urbano - Semidenso	94
6.5. Resultados.....	95
6.5.1. Laboratorio	95
6.5.2. Mediciones de RSSI, SNR y perdida de paquetes - Rural.....	100
6.5.3. Mediciones de RSSI, SNR y perdida de paquetes - Urbano.....	101
6.6. Conclusiones.....	103
7. Simulaciones.....	105
7.1. Reseña del capítulo	105
7.2. Introducción	105
7.3. Software de simulación	105
7.3.1. LoRaSim.....	105
7.3.2. FLoRa	106
7.4. Metricas de simulación.....	107
7.4.1. Relación de entrega de paquetes (PDR)	107
7.4.2. número de colisiones	107
7.4.3. Distribución espacial de los nodos	107
7.4.4. Histograma de factores de dispersión	107
7.5. Parámetros de simulación.....	108
7.6. Resultados de simulación	108
7.6.1. Resultados de simulación rural.....	108
7.6.2. Resultados de simulación Urbana.....	114
7.7. Conclusiones.....	120

8.	Análisis de aplicación en redes existentes	123
8.1.	Reseña del capítulo	123
8.2.	Redes HFC.....	123
8.2.1.	Redes Nodo+0.....	123
8.2.2.	Redes HFC Nodo+1 o Nodo+2.....	124
8.3.	Densidad y rendimiento de la red.....	125
8.4.	Conectividad de alimentación y backhaul	130
8.4.1.	Alimentación de energía	130
8.4.2.	Conectividad de red	130
8.5.	Conclusiones	131
9.	Conclusiones y trabajos futuros	133
9.1.	Conclusiones	133
9.2.	Trabajos futuros.....	135
10.	Publicaciones y trabajos a que dio lugar la tesis.....	137
10.1.	Patentes	137
10.2.	Publicaciones en revistas internacionales	137
10.3.	Publicaciones en Congresos internacionales	137
10.4.	Publicaciones en Congresos	137
10.5.	Libros / Normas tecnicas	138
10.6.	Desarrollos tecnologicos	138
11.	Referencias Bibliográficas.....	139
	Anexo 1 – Codigo de sensores y aplicaciones.....	145
	Anexo 2 – Tablas de mediciones	161

LISTA DE TABLAS

Tabla 1. Aplicaciones de Internet de las Cosas. Adaptado de (Libelium, 2018).....	17
Tabla 2. Principales tecnologías LPWAN.(Adelantado et al., 2017)	18
Tabla 3. Tabla comparativa de pros vs contras según tipo de espectro. (Machina Research, 2015)	18
Tabla 4. Funciones de un servidor de red LoRaWAN.....	26
Tabla 5. Tasas de Código para LoRaWAN. Adaptado de (Noreen et al., 2017).	31
Tabla 6. Factor de Dispersión vs Longitud de Chip.	38
Tabla 7. Factor de dispersión vs Sensibilidad y Presupuesto de Enlace.....	39
Tabla 8. Tasa de bits cruda del canal frente al ancho de banda del canal, factor de dispersión y tasa de codificación	44
Tabla 9. Tasa de bits máxima considerandos encabezados MAC Y FRM.....	45
Tabla 10. Tasa de bits del canal frente al factor de dispersión y la tasa de codificación para ALOHA	47
Tabla 11. Tasa de bits total de la puerta de enlace frente al factor de dispersión y la tasa de codificación para Aloha en una puerta de enlace de 8 canales.	47
Tabla 12. Bandas de espectro para LoRaWAN en EE. UU.....	48
Tabla 13. Asignación de espectro para LoRaWAN en Argentina	49
Tabla 14. Modos de operación LoRa para el plan AU915	49
Tabla 15. Especificaciones técnicas de FiPy	60
Tabla 16. Especificaciones técnicas de PySense	62
Tabla 17. Especificaciones técnicas de PyTrack	63
Tabla 18. Especificaciones técnicas de Pygate	64
Tabla 19. Especificaciones técnicas de LoPy4	65
Tabla 20. Especificaciones técnicas del LimeSDR.....	69
Tabla 21. Definiciones de parámetros de dispositivos.....	82
Tabla 22. Tabla de frecuencias del gateway	86
Tabla 23. Carga útil de pruebas de campo	93

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Tabla 24. Distancias de puntos de medición en laboratorio	93
Tabla 25. Distancias de puntos de medición en escenario rural	94
Tabla 26. Distancias de puntos de medición en escenario urbano	95
Tabla 27. Exponentes de atenuación en diferentes entornos	99
Tabla 28. Cálculo de capacidad máxima de canal.....	114
Tabla 29. Cálculo de capacidad máxima de canal.....	120
Tabla 30. Tiempo de aire en función de la carga útil	126
Tabla 31. Ciclo de actividad y cantidad de dispositivos en función de la carga útil	127
Tabla 32. Ciclo de actividad y para diferentes tipos de dispositivos.....	128
Tabla 33. Ciclo de actividad agregado para casas	128
Tabla 34. Densidad de sensores por escenario	129
Tabla 35. Cálculo de cantidad de gateways de 8 canales según escenario.....	130

LISTA DE FIGURAS

Figura 1. Las dimensiones de las comunicaciones en IoT según lugar, momento o entre objeto. Recuperado de (ITU, 2012).....	10
Figura 2. Creación de ambientes y espacios inteligentes. Adaptado de (Smith, 2012).....	10
Figura 3. Proyección de dispositivos IoT conectados. Recuperado de (IoT Analytics, 2021).....	12
Figura 4. <i>Stack</i> del protocolo LORAWAN. Adaptado de Semtech.com.....	22
Figura 5. Topología estrella de estrellas LoRaWAN representando la ubicación de gateways y nodos. Adaptado de (Riegsecker, 2017).....	22
Figura 6. Arquitectura lógica de referencia de LoRaWAN. Elaboración Propia basado en (<i>LoRaWAN Specification 1.1</i> , 2017).....	23
Figura 7. Arquitectura de referencia de LoRaWAN mostrando los cuatro niveles de la red y los protocolos de comunicación entre ellos. Recuperado de The Things Network	24

Figura 8. Mapeo de LoRaWAN al modelo OSI. Adaptado de (<i>LoRa and LoRaWAN</i> , n.d.).....	28
Figura 9. Modulación con códigos. Elaboración Propia.....	29
Figura 10. Ejemplo del comportamiento en frecuencia y tiempo de una Modulación con espectro ensanchado y su robustez frente al ruido. Elaboración Propia	29
Figura 11. Modulación por <i>Chirps</i> . Elaboración Propia.....	30
Figura 12. Factor de dispersión vs duración de <i>Chirp</i> . Adaptado de (Sakshama Ghosly, 2018)	31
Figura 13. Dispersión de Símbolos en LoRaWAN. Elaboración Propia.....	32
Figura 14. Estructura de paquete LoRaWAN. Elaboración propia	36
Figura 15. Tiempo de Aire vs Carga Útil en LoRaWAN. Elaboración Propia.	38
Figura 16. Contención en LoRaWAN. Elaboración Propia.....	40
Figura 17. Transmisión de un dispositivo Clase A. Elaboración Propia	40
Figura 18. Transmisión de un dispositivo Clase B. Elaboración Propia	41
Figura 19. Transmisión de un dispositivo Clase. Elaboración Propia	41
Figura 20. Curva de rendimiento de LoRaWAN considerando encabezados y corrección de errores. Elaboración propia.....	45
Figura 21. Curva de rendimiento del protocolo ALOHA vs carga del canal. (Polonelli et al., 2019).	46
Figura 22. Mapa de canales del plan de frecuencias LoRa en Argentina. (Semtech, n.d.).....	49
Figura 23. Comparación de FSPL vs Hata. Elaboración Propia.....	51
Figura 24. Atenuación vs distancia para escenario urbano denso. Elaboración Propia.....	53
Figura 25. Atenuación vs distancia para escenario suburbano. Elaboración Propia	55
Figura 26. Imagen de Pycom FiPy.....	60
Figura 27. Imagen de Pycom <i>Expansion Board</i>	61
Figura 28. Imagen de Pycom PySense.....	61
Figura 29. Imagen de Pycom PyTrack.....	62
Figura 30. Imagen de Pycom Pygate	63
Figura 31. Diagrama de bloques de un transceptor SDR.....	66

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Figura 32. Diagrama de bloques de un receptor SDR basado en FPGA	68
Figura 33. Imagen del Lime SDR.....	69
Figura 34. Arquitectura de ChirpStack.....	71
Figura 35. Interfaz gráfica de GNU Radio	75
Figura 36. Diagrama de entorno de Laboratorio	78
Figura 37. Chirpstack - Pantalla de acceso.....	79
Figura 38. Chirpstack - Agregar un Perfil de dispositivo.....	81
Figura 39. Chirpstack - Agregar un perfil de dispositivo – Codificador.....	82
Figura 40. Chirpstack - Agregar un Gateway.....	83
Figura 41. Chirpstack - Agregar una Aplicación.....	83
Figura 42. Chirpstack - Configurar la integración con la base de datos.....	84
Figura 43. Chirpstack - Agregar un Dispositivo a la aplicación.....	84
Figura 44. Chirpstack - Configurar el Application Key en el dispositivo.....	85
Figura 45. Fotografía de Gateway de Laboratorio.....	85
Figura 46. Fotografía del sensor PySense	87
Figura 47. Fotografía del sensor PyTrack	89
Figura 48. Fotografía del receptor LimeSDR.....	90
Figura 49. Diagrama en bloques de un proyecto GNURadio.....	91
Figura 50. Panel de visualización de datos en Grafana	92
Figura 51. Panel de visualización de datos en Grafana	92
Figura 52. Geolocalización de puntos de medición en escenario rural	94
Figura 53. Geolocalización de puntos de medición en escenario urbano.....	95
Figura 54. Espectrograma o diagrama de intensidad de señal con respecto a frecuencia y tiempo de una transmisión LoRaWAN - SF12 - 125 kHz.....	96
Figura 55. Análisis de espectro de un canal de 125 kHz en 917.0 MHz.....	97

Figura 56. Análisis de espectro de los ocho canales de 125 kHz en la subbanda 1	97
Figura 57. Mediciones de RSSI vs Distancia – Laboratorio.....	100
Figura 58. Mediciones de RSSI vs Distancia - Rural	101
Figura 59. Mediciones de SNR vs Distancia – Rural	101
Figura 60. Porcentaje de entrega de paquetes - Rural.....	101
Figura 61. Mediciones de RSSI vs Distancia - Urbano	103
Figura 62. Mediciones de SNR vs Distancia – Urbano	103
Figura 63. Porcentaje de entrega de paquetes – Urbano	103
Figura 64. Simulación – Rural - Tasa de entrega de paquetes vs cantidad de sensores vs canales	109
Figura 65. Simulación – Rural - Tasa de entrega de paquetes vs cantidad de sensores vs carga útil	110
Figura 66. Simulación – Rural - Tasa de entrega de paquetes vs cantidad de sensores vs intervalo	111
Figura 67. Simulación – Rural - Colisiones vs cantidad de sensores vs canales	112
Figura 68. Simulación – Rural – Visualización de factor de dispersión en función de diferentes distancias al gateway para una distribución aleatoria de sensores.....	113
Figura 69. Simulación – Rural – Distribución de sensores vs. factor de dispersión.....	114
Figura 70. Simulación – Urbano - Tasa de entrega de paquetes vs. cantidad de sensores vs canales	115
Figura 71. Simulación – Urbano - Tasa de entrega de paquetes vs. cantidad de sensores vs carga útil	116
Figura 72. Simulación – Urbano - Tasa de entrega de paquetes vs. cantidad de sensores vs intervalo	117
Figura 73. Simulación – Urbano - Colisiones vs cantidad de sensores vs canales	118
Figura 74. Simulación – Urbano – Distribución de sensores vs factor de dispersión	119
Figura 75. Simulación – Urbano – Distribución de sensores vs factor de dispersión	120
Figura 76. Red de distribución Nodo+0 de ejemplo	124

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Figura 77. Red de distribución Nodo+1 de ejemplo..... 125

LISTA DE ECUACIONES

Ecuación 1. Tasa de bits de LoRaWAN 32

Ecuación 2. Teorema de Shannon 33

Ecuación 3. Teorema de Shannon simplificado para Spread Spectrum 33

Ecuación 4. Ruido de Johnson-Nyquist..... 33

Ecuación 5. Ruido Johnson-Nyquist a temperatura ambiente 34

Ecuación 6. Sensibilidad de un receptor de radio..... 34

Ecuación 7. Cálculo de enlace 34

Ecuación 8. Atenuación del espacio libre..... 35

Ecuación 9. Tiempo de Aire de LoRaWAN..... 36

Ecuación 10. Tiempo de Preámbulo de LoRaWAN..... 36

Ecuación 11. Tiempo de Símbolo de LoRaWAN 37

Ecuación 12. Tasa de Símbolo de LoRaWAN 37

Ecuación 13. Tiempo de Carga Útil de LoRaWAN 37

Ecuación 14. Longitud de Carga Útil de LoRaWAN..... 37

Ecuación 15. Tiempo de paquete de LoRaWAN..... 38

Ecuación 16. Tasa de Bits de LoRaWAN 44

Ecuación 17. Formula de la atenuación en el espacio libre 50

Ecuación 18. Periodo de símbolos LoRaWAN 96

Ecuación 19. Potencia de señal recibida en función de transmitida y atenuación..... 98

Ecuación 20. Modelo log-normal atenuación..... 98

Ecuación 21. Modelo log-normal atenuación..... 99

Ecuación 22. RSSI modelado de datos	99
Ecuación 23. RSSI teórico con $n=1.61$	99
Ecuación 24. Tasa de entrega de paquetes (PDR)	107
Ecuación 25. Capacidad de canal con factores mixtos	113
Ecuación 26. Formula del ciclo de actividad	126
Ecuación 27. Formula de la cantidad máxima de sensores en condiciones ideales.....	127
Ecuación 28. Formula de la cantidad máxima de sensores en un canal ALOHA	127
Ecuación 29. Ciclo de actividad total de un grupo de sensores	129
Ecuación 30. Ciclo de actividad máximo en función de numero de canales	129

RESUMEN

En los últimos diez años, el mundo de las telecomunicaciones se enfocó en el desarrollo y la implementación de una infraestructura de internet de las cosas (IoT), así como en casos de aplicación y de negocios basados en IoT. Sin embargo, la mayoría de las implementaciones se han realizado sobre redes preexistentes conectadas a internet o también llamadas *Over-The-Top*. Esto se debe en gran medida a la naturaleza agnóstica de la red en muchas aplicaciones de IoT de consumo masivo. Las aplicaciones de IoT se conectan directamente a sus proveedores de IoT a través de conexiones abiertas a Internet, utilizando infraestructura de redes comerciales. En el ámbito de la República Argentina se ha regulado la banda de operación de redes de internet de las cosas de espectro no regulado en el espectro radio eléctrico en 915 MHz. Las bandas no licenciadas no son las mismas en diferentes partes del mundo.

Teniendo esto en cuenta, esta tesis se propone responder la siguiente pregunta: ¿Cuáles son los factores principales que influyen en mayor medida en la operación viable de una red de internet de las cosas en el espectro de radio frecuencia no licenciado de la República Argentina? Para ello se plantea como objetivo principal: evaluar los factores que influyen en mayor medida en la operación viable de una red de internet de las cosas en el espectro de radiofrecuencia no licenciado de la República Argentina, y como objetivos específicos: identificar los parámetros de operación más importantes de una red de internet de las cosas, describir qué factores influyen en mayor medida en la operación de una red IoT, comparar modelos teóricos de operación de redes IoT con resultados experimentales y/o simulados y, calcular el efecto de la variación de la cantidad de dispositivos en los parámetros de operación más relevantes. Y, plantea dos hipótesis de trabajo: i) la cantidad de dispositivos influye significativamente en la operación viable de una red de internet de las cosas en el espectro de radiofrecuencia no licenciado de la República Argentina y ii) La diversidad de áreas de aplicación influye significativamente en la operación viable de una red de internet de las cosas.

Para probar las hipótesis y cumplir los objetivos se definieron algunos parámetros mediante búsqueda bibliográfica exhaustiva y se realizaron una serie de experimentos concatenados. Se analizaron las principales alternativas de los protocolos de IoT que no requieren de una banda licenciada, particularmente los de Redes amplias (WAN) de baja potencia (LP-WAN) y las comparó con los protocolos basados en dispositivos móviles de banda licenciada como LTE-M y NB-IOT. Dado que las bandas no licenciadas no son las mismas en diferentes partes del mundo, el alcance de esta tesis es analizar el crecimiento de la cantidad de dispositivos utilizando este tipo de redes en el ámbito de la República Argentina. Se definió como marco teórico los principales

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

tipos de redes de internet de las cosas y, en detalle del protocolo LoRaWAN como caso más representativo de redes en espectro no licenciado. A su vez, se realizó un estudio teórico de los factores principales más influyentes en la performance de este tipo de redes que sirvieron para definir el alcance: la cantidad de espectro disponible, la cantidad de dispositivos, y el terreno donde la red se encuentra desplegada. Posteriormente, se realizó un análisis detallado y una implementación real de una red con el objetivo de determinar experimentos que permitan validar bajo condiciones reales —tanto de laboratorio como de campo— el efecto de los factores identificados como principales. Esto llevó a la realización de las pruebas en tres diferentes puntos representativos de una ciudad, y en cuatro puntos de un escenario rural con el objetivo de confirmar los resultados teóricos y la validación de los modelos de propagación en los dos escenarios diferentes (rural y urbano), con los parámetros de red aplicables a la República Argentina. Por último, a efectos de validar la hipótesis de la influencia de cantidad de dispositivos, se utilizaron tanto los modelos teóricos como los ajustes obtenidos en el campo para realizar catorce escenarios de simulación. Estos permitieron concluir que la performance de las redes se ve afectada por el aumento de la cantidad de dispositivos, el periodo del ciclo actividad de estos, y por ende del tipo de medición que realizan, variando sus límites según el escenario sea urbano o rural y por la cantidad de canales que se dedican al servicio. Con las conclusiones obtenidas y los datos de los análisis previos se propone un modelo de cobertura de red y de localización de puertas de enlace LoRa para optimizar la implementación de este tipo de redes donde en general se verificó una buena correlación entre la ubicación actual de los equipos activos de red con la localización potencial de puertas de enlace LoRaWAN.

Finalmente, y como puntapié para trabajos futuros se propone: trabajos en el análisis automatizado de los datos de telemetría de los sensores con herramientas de *machine learning* para optimizar la potencia de los sensores, un análisis de vulnerabilidades de seguridad y en uno particularmente importante en el área de legislación para la expansión del espectro disponible para este tipo de redes.

A nivel del país o la región, de esta tesis se podrá desprender información relevante como recomendación para ayudar a definir a nivel regulatorio las mejores bandas de operación para los protocolos de IoT en función de la mejor performance de estos y, por lo tanto, con un impacto tangible en la eficiencia de la operación de este tipo de redes.

1. INTRODUCCIÓN

Considerando el crecimiento exponencial previsto en la cantidad de dispositivos del tipo de internet de las cosas (IoT) (IoT Analytics, 2021) para los próximos años con una tasa anual agregada de 21%, este crecimiento llevara consigo a una demanda aumentada de capacidad de transmisión de datos y, por consiguiente, de espectro de radiofrecuencia para soportarla. Por otro lado, la diversidad emergente de protocolos de Redes de área amplia de baja potencia (LPWAN) —como NB-IoT, LTE CAT-M, LoRaWAN y SigFox, entre otros— harán uso de maneras muy diversas de ese espectro (Hassan, 2018). Es pertinente y relevante considerar la realización de un estudio del efecto de este crecimiento de la cantidad de dispositivos en redes del tipo LPWAN en las señales de radiofrecuencia utilizadas por los protocolos inalámbricos anteriormente mencionados. Y particularmente los efectos relacionados con a la operación viable de este tipo de redes y del efecto que puede tener en la capa de transmisión física a nivel de señales de radio frecuencia cuando uno o más protocolos operan en zonas del espectro muy cercanas o incluso comparten el mismo, tal como se ha realizado en otros casos de tecnologías inalámbricas (Valenta et al., 2010).

Desde un punto de vista más amplio, es de particular importancia mencionar el impacto que tienen actualmente las redes de internet de las cosas, que conectan cualquier tipo de objetos a internet donde sus efectos en la sociedad de hoy en día, van desde reducir la polución en las ciudades más grandes del mundo, mejorar la agricultura y la cadena de suministros de los alimentos o hasta detectar y contener virus potencialmente mortales (Kranz, 2018).

Como ha sido mencionando anteriormente, para poder soportar de manera sostenible este crecimiento exponencial en la cantidad de dispositivos, estas redes de área amplia de baja potencia, proveen una solución muy efectiva dado que permiten cubrir distancias muy grandes con un mínimo de potencia y mantenimiento (Petajajarvi et al., 2015), al mismo tiempo que estas redes pueden operar en bandas de frecuencias no licenciadas y por lo tanto, cualquier usuario final construir su propia red mediante la colocación de un *Gateway* de manera muy similar a lo que se hace hoy en día con las redes WiFi (Wixted et al., 2016).

En el ámbito de la República Argentina se ha regulado recientemente las bandas de operación de redes de internet de las cosas en el espectro radio eléctrico según resolución 581/2018 (*Resolución 581/2018*, 2018) y la Resolución de ENACOM 4653/2019 (*Resolución 4653/2019*, 2019) y por lo tanto, se considera particularmente relevante un estudio científico que permita responder a la siguiente pregunta de investigación: “¿Cuáles son los factores principales que influyen en mayor

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

medida en la operación viable de una red de internet de las cosas en el espectro de radiofrecuencia no licenciado de la República Argentina?”

El tipo de investigación realizada es del tipo descriptivo dado que tal como se describe en la pregunta de investigación, en primera instancia se buscará describir qué factores influyen en mayor medida en la operación de una red IoT, seguido por la definición de los parámetros de operación viable según las aplicaciones más comunes utilizadas en estas redes, para finalmente abordar la descripción de los factores más influyentes con respecto a los parámetros de operación viable, todo esto en el contexto del espectro asignado en la República Argentina.

Los resultados obtenidos en esta tesis podrán servir como base para otros proyectos de tesis o investigación futuros. Dicho esto, este proyecto puede tener un impacto en la producción científica del Doctorado en Ingeniería de la facultad, mediante el establecimiento de una continuidad de investigación en el área de señales y redes inalámbricas. No será objetivo de esta tesis proponer mejoras o cambios a los protocolos existentes, así como tampoco proponer nuevos protocolos para IoT.

1.1. CONTRIBUCIONES Y LIMITACIONES DE LA INVESTIGACIÓN

A nivel del país o la región latinoamericana, de esta tesis se podrá desprender información relevante como recomendación para ayudar a definir a nivel regulatorio las mejores bandas de operación para los protocolos de IoT en función de la mejor performance de estos y, por lo tanto, con un impacto tangible en la eficiencia de la operación de este tipo de redes.

Por otro lado, permitirá a nuevos emprendedores de este tipo de redes comprender las capacidades y limitaciones de capacidad para realizar una mejor planificación en el despliegue de este tipo de proyectos.

Por último, la revisión y análisis exhaustivo del protocolo LoRaWAN permitirá a quien lea este trabajo poder comprender en detalle los métodos de operación disponibles y cómo poner en marcha y analizar una red de este tipo.

No será objetivo de esta tesis proponer mejoras o cambios a los protocolos existentes, así como tampoco proponer nuevos protocolos para IoT. Tampoco se han analizado detalles sobre potenciales optimizaciones en tiempo real del funcionamiento de una red de este tipo, así como

tampoco potenciales afectaciones al servicio por ataques de denegación de servicio lo que fue comentado como áreas de trabajo futuras.

Finalmente, por limitaciones de recursos y tiempo no se analizaron escenarios extremadamente densos como por ejemplo el microcentro de la Ciudad Autónoma de Buenos Aires o alguna metrópolis del misma configuración y envergadura.

1.2. HIPÓTESIS

Las hipótesis fundamentales de este trabajo son: a) La cantidad de dispositivos influye significativamente en la operación viable de una red de internet de las cosas en el espectro de radiofrecuencia no licenciado de la República Argentina y b) La diversidad de áreas de aplicación influye significativamente en la operación viable de una red de internet de las cosas.

1.3. OBJETIVO PRINCIPAL

Evaluar los factores que influyen en mayor medida en la operación viable de una red de internet de las cosas en el espectro de radiofrecuencia no licenciado de la República Argentina.

1.4. OBJETIVOS ESPECÍFICOS

Identificar los parámetros de operación más importantes de una red de internet de las cosas

Describir qué factores influyen en mayor medida en la operación de una red IoT.

Comparar modelos teóricos de operación de redes IoT con resultados experimentales y/o simulados

Calcular el efecto de la variación de la cantidad de dispositivos en los parámetros de operación más relevantes

1.5. METODOLOGÍA DE INVESTIGACIÓN UTILIZADA

Actividades Realizadas:

- i) investigación bibliográfica;
 - a. Publicaciones de la IEEE Communications Society
 - i. *Journal on Wireless Communications*
 - ii. *Journal on Internet of Things*
 - b. Publicaciones en conferencias relacionadas con *Internet of Things* y *Sensor Networks*

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

- ii) Asistencia y aprobación de doctorado que aportaron el conocimiento necesario para llevar a cabo la investigación de este trabajo
- iii) Participación en congresos, conferencias y seminarios.
- iv) Definición de modelos teóricos: basados en la bibliografía, se definieron modelos teóricos esperados de la operación de IoT para interacciones entre diferentes protocolos, cantidades de dispositivos y aplicaciones para la banda no licenciada utilizada en la República Argentina.
- v) Diseño de experimentos: se definieron las mediciones y simulaciones de operación de diferentes protocolos IoT y sus interacciones a ser realizadas.
- vi) Realización de experimentos: realización de mediciones y simulaciones definidas previamente en diferentes condiciones: se realizaron mediciones de señales utilizando transceptores IoT en la banda no licenciada utilizando los protocolos más comunes, así como también utilizando diferentes aplicaciones con diferentes anchos de banda. Se buscaron obtener formas de señales y utilización del espectro mediante el uso de analizadores de espectro basados en *Software Defined Radio*, así como de analizadores de protocolos.
- vii) Realización de simulaciones: con el objetivo de lograr mayor escalabilidad de los resultados para condiciones donde no es posible realizar mediciones de laboratorio, como por ejemplo para conocer los resultados de interacciones de grandes cantidades de dispositivos y protocolos, se realizaron simulaciones utilizando MATLAB y Python a efectos de poder efectuar acciones de procesamiento sobre datos simulados, así como operaciones de procesamiento digital de señales.
- viii) Comparación entre modelos teóricos y datos empíricos obtenidos de experimentos y simulaciones. Se realizó una comparación de los resultados obtenidos de las mediciones de dispositivos reales y de las simulaciones realizadas con Matlab y Python, en caso de que se encuentren diferencias significativas se buscara encontrar el origen de estas.

1.6. CONTENIDO Y ESTRUCTURA DE LA TESIS

La tesis se encuentra estructurada en seis partes.

En la primera parte, capítulo 1, se presenta una introducción general al problema, la metodología utilizada, hipótesis y objetivos de investigación, contribuciones y limitaciones de la investigación.

En la segunda parte, capítulos 2 y 3, se describen los marcos teóricos principales que incluyen los conceptos teóricos básicos de la internet de las cosas y del protocolo LoRaWAN. En capítulo 2 se realiza una revisión del estado del arte de la internet de las cosas pasando por el impacto en la sociedad de sus aplicaciones y describiendo en detalles los principales casos de uso y tecnologías disponibles particularmente separando las bandas licenciadas y no licenciadas. Mientras que en el capítulo 3 se presenta al protocolo en banda no licenciada llamado LoRaWAN, se hace una revisión en detalle del protocolo pasando por sus componentes principales, bases teóricas de operación y ecuaciones descriptivas de dicha operación.

En la tercera parte, capítulo 4, se realiza una evaluación teórica y descripción detallada de la capacidad de tráfico del protocolo LoRaWAN, se establecen las limitantes para los diferentes escenarios de cantidad de canales y se presenta la aplicación de estos cálculos al escenario de la República Argentina.

En la cuarta parte, que incluye los capítulos 5 y 6, se presentan los materiales y métodos utilizados para los ensayos de laboratorio y de campo. En el capítulo 5 se hace una revisión de todos los materiales a ser utilizados para crear un entorno de red LoRaWAN y aplicarlo al trabajo de laboratorio y de campo. Se describen las herramientas de software y hardware utilizadas y los beneficios de estas. Mientras que en el capítulo 6 se describen los conceptos utilizados para los ensayos de laboratorio y campo, se presentan las configuraciones de cada uno de los elementos junto con la definición de cada uno de los experimentos para los escenarios de laboratorio, rural y urbano y los resultados de estos.

En la quinta parte, capítulo 7 y 8, se presentan escenarios de simulación para permitir evaluar la escalabilidad de dispositivos junto con una aplicación en redes reales de los resultados obtenidos en una red real del tipo híbrido fibra coaxial, calculando su capacidad y conectividad.

Y, en la sexta parte, capítulos 9 y 10, se exponen las conclusiones, los futuros trabajos y las publicaciones que generó este trabajo.

2. INTERNET DE LAS COSAS

2.1. RESEÑA DEL CAPÍTULO

En este capítulo se ofrece una introducción inicial al concepto de la internet de las cosas, pasando por sus inicios, pasos iniciales de adopción para después mostrar una lista resumida de sus campos de aplicaciones. Una vez introducido el tema se discute el concepto de redes de área amplia de baja potencia como base fundacional de la tecnología de la internet de las cosas que permite grandes áreas de cobertura para dispositivos con poco consumo de energía y con un potencial de crecimiento a billones de dispositivos en los próximos años. Se realiza una enumeración detallada de los diferentes casos de uso por diferentes verticales y finalmente se comparó las diferentes tecnologías LPWAN en bandas licenciadas y no licenciadas.

2.2. INTRODUCCIÓN A INTERNET DE LAS COSAS

El término “Internet de las Cosas” (*Internet of Things* o *IoT*) fue empleado por primera vez por Kevin Ashton en el año 1999 en el contexto de una presentación acerca de cómo mejorar la eficiencia de los sistemas de la cadena de suministro en una empresa de provisión bienes mediante el uso marcadores de radiofrecuencia (*RFID*) en lugar de los códigos de barras y como lograr una recolección de datos automatizada mediante el uso de una red y principalmente la eliminación del factor humano en dicha recolección de los datos (Ashton, 2009). Cuestión particularmente importante dado que el término “Internet de las Cosas” trae aparejado el concepto de que “las cosas comienzan a usar la red de forma tal que las personas no lo necesiten hacer” y que relaciona la automatización de la comunicación de las cosas, ahora atada el crecimiento de internet como red de interconexión global (Gershenfeld, 1999).

En 2012, una de las definiciones más amplias de la Internet de las Cosas fue realizada por la Unión Internacional de Telecomunicaciones (ITU). La cual la define como a “la Infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación presentes y futuras (ITU, 2012). Así mismo, cada una de estas cosas u objetos se trata de un objeto del mundo físico (objetos físicos) o del mundo de la información (objetos virtuales) que puede ser identificado e integrado en las redes de comunicaciones.

Un factor muy importante también definido por la ITU es la ubicuidad de la comunicación. Tal como se ve en la Figura 1, el Internet de las Cosas agrega una nueva dimensión, la de comunicación entre objetos, y no solo entre computadoras o personas (ITU, 2012).

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

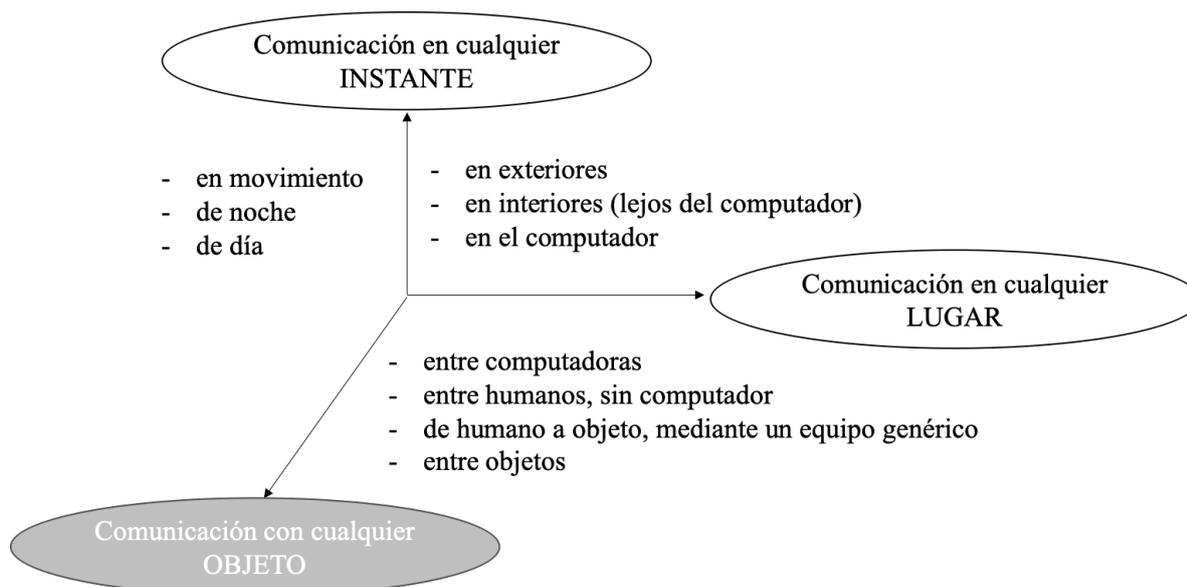


Figura 1. Las dimensiones de las comunicaciones en IoT según lugar, momento o entre objeto. Recuperado de (ITU, 2012)

Un objetivo de vital importancia para el desarrollo de la internet de las cosas fue la creación de ambientes y espacios inteligentes con cosas con “vida propia” (ver p. ej. transporte inteligente, productos, ciudades, áreas rurales, salud (Smith, 2012). Dichos espacios o ambientes inteligentes requieren para su funcionamiento de un conjunto de tecnologías funcionando en forma coordinada que va desde: los sensores; pasando por el procesamiento local; la interconexión de los sensores; y llegando a la utilización del procesamiento masivo de datos en la nube, como se muestra en la Figura 2.

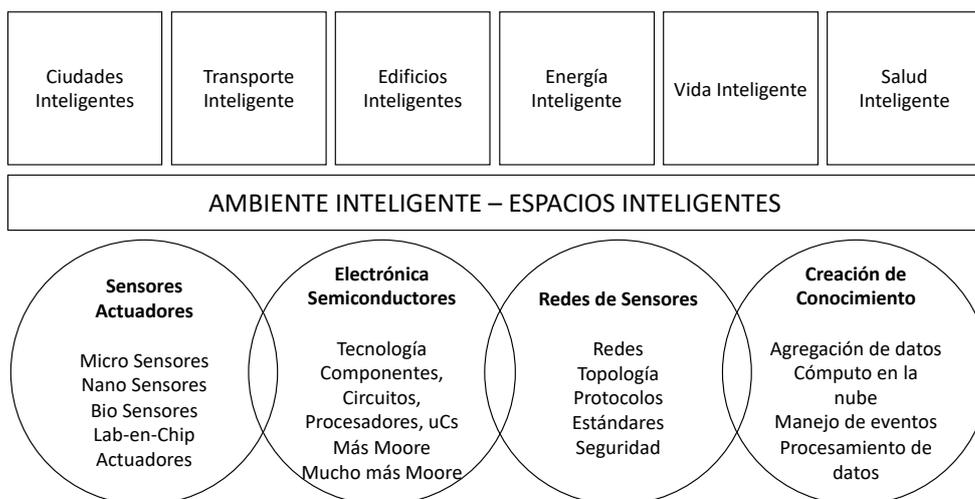


Figura 2. Creación de ambientes y espacios inteligentes. Adaptado de (Smith, 2012)

Una lista de diferentes aplicaciones de Internet de las Cosas basados en diferentes sensores y en diferentes segmentos de mercado se muestra en la Tabla 1, dicha lista incluye ocho diferentes verticales en las cuales se mencionan los escenarios más llamativos (Libelium, 2018).

Para intercambiar datos entre aplicaciones, dispositivos y cosas, existen varios estándares de comunicación como Bluetooth, WiFi y varios estándares móviles basados en GSM (2G/3G/4G), en general estos estándares han logrado, con avances tecnológicos, incrementar significativamente las tasas de transmisión de datos y por ende la capacidad de transmitir imágenes o videos en tiempo real. Sin embargo, la mayoría de los casos de uso de IoT mencionados anteriormente no requieren tasa de transmisión de datos muy altas, aunque al mismo tiempo se requiere que el consumo energético de los sensores sea muy bajo dados que los estos en general se encuentran en zonas remotas de no tan fácil acceso y además con espacio muy acotado para la colocación de baterías.

2.3. REDES DE ÁREA AMPLIA DE BAJA POTENCIA (LPWAN)

El término LPWAN o Red de área amplia de baja potencia, surgió en 2013 como una solución al tipo de conectividad necesaria descrito en la sección anterior.

Si bien no denota ninguna solución tecnológica o protocolo de comunicación en particular, sirve como un término genérico para referirse a la familia de tecnologías de red diseñadas para comunicarse de forma inalámbrica a largas distancias, utilizando menos energía que en el caso de otras redes tradicionales.

Estas dos características principales, la eficiencia energética y la amplia cobertura, no solo dan nombre de la tecnología, sino que son un soporte constitutivo fundamental para las redes IoT.

Se pueden mencionar algunas características clave de los dispositivos y tecnologías de estas redes tales como: Baterías de larga duración (a menudo superior a 10 años), conectividad de larga distancia (mayor a 30 Km.), conjuntos de chips y redes de bajo costo, rendimientos de ancho de banda de datos limitados.

En consecuencia, el objetivo de las Redes de área amplia de baja potencia (LPWAN) se ha convertido en un tema central en IoT. LPWAN tal como se menciona es un término amplio para una variedad de tecnologías utilizadas para conectar sensores y controladores a Internet sin el uso de WiFi o redes celulares tradicionales. Han surgido dos principales tipos de estándares para redes LPWAN, los basados en celulares y en general utilizando bandas licenciadas, por ejemplo, NB-

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

IoT o LTE-M y los desarrollados nativamente para casos de uso de IoT como lo son LoRaWAN y SigFox los cuales utilizan bandas no licenciadas. Las consideraciones de diseño predominantes son una fuerte penetración, la conexión de una gran cantidad de sensores y dispositivos de ancho de banda muy bajos (Hassan, 2018).

Desde 2016 y al menos hasta 2020, los dispositivos de IoT han aumentado a una tasa del 26 por ciento anual y se espera que hasta 2026 aumenten a una tasa compuesta anual de 22 por ciento, impulsada por los nuevos casos de uso mencionados anteriormente (IoT Analytics, 2021) y por lo tanto la necesidad de espectro será cada vez marcada para servir a los casi 27 mil millones de dispositivos de los cuales al menos 2100 millones serán del tipo LPWAN no licenciado. En la Figura 3 se representa el crecimiento esperado de dispositivos para los próximos 3 años.

Global IoT market forecast (in billion connected IoT devices)

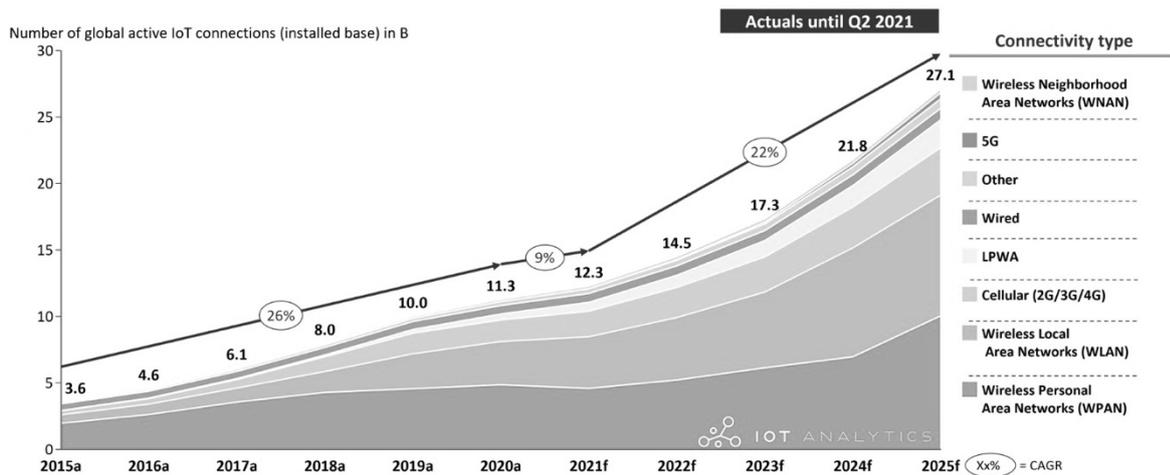


Figura 3. Proyección de dispositivos IoT conectados. Recuperado de (IoT Analytics, 2021)

2.4. APLICACIONES DE REDES LPWAN

Internet de las Cosas

Ciudades Inteligentes	Medio Ambiente y Agua	Medición	Seguridad y Emergencias	Retail y Logística	Control industrial	Agricultura y Ganadería	Domótica y eHealth
Estacionamiento inteligente	Detección de incendios forestales	Red Eléctrica inteligente	Control de acceso perimetral	Control de cadena de suministro	Aplicaciones M2M	Mejora de la calidad del vino	Uso de energía y agua
Monitoreo de disponibilidad de espacios de estacionamiento en la ciudad.	Monitoreo de gases de combustión y condiciones de incendio preventivas para definir zonas de alerta.	Control y gestión del consumo de energía.	Control de acceso a áreas restringidas y detección de personas en áreas no autorizadas.	Monitoreo de las condiciones de almacenamiento a lo largo de la cadena de suministro y seguimiento del producto con fines de trazabilidad.	Autodiagnóstico de máquinas y control de activos.	Controlar la humedad del suelo y el diámetro del tronco en los viñedos para controlar la cantidad de azúcar en la uva y la salud de la vid.	Monitoreo del consumo de energía y suministro de agua para obtener asesoramiento sobre cómo ahorrar costos y recursos.
Salud estructural	Contaminación del aire	Nivel de tanque	Presencia líquida	Pago NFC	Calidad del aire interior	Invernaderos	Electrodomésticos de control remoto
Monitoreo de vibraciones y condiciones materiales en edificios, puentes y monumentos históricos.	Control de emisiones de CO2 de fábricas, contaminación emitida por automóviles y gases tóxicos generados en granjas.	Monitoreo de niveles de agua, aceite y gas en tanques de almacenamiento y cisternas.	Detección de líquidos en centros de datos, almacenes y terrenos sensibles para evitar fallas y corrosión.	Procesamiento de pagos basado en la ubicación o duración de la actividad para el transporte público, gimnasios, parques temáticos, etc.	Monitoreo de los niveles de gas tóxico y oxígeno dentro de las plantas químicas para garantizar la seguridad de los trabajadores y los bienes.	Controlar las condiciones micro climáticas para maximizar la producción de frutas y verduras y su calidad.	Encender y apagar aparatos a distancia para evitar accidentes y ahorrar energía.
Mapas urbanos de Ruido	Control de nivel de nieve	Instalaciones fotovoltaicas	Niveles de radiación	Aplicaciones de compras inteligentes	Control de temperatura	Canchas de golf	Sistema de detección de intrusos

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Monitoreo de sonido en áreas de bares y zonas céntricas en tiempo real.	Medición del nivel de nieve para conocer en tiempo real la calidad de las pistas de esquí y permitir la prevención de avalanchas en el cuerpo de seguridad.	Monitoreo y optimización del rendimiento en plantas de energía solar.	Medición distribuida de los niveles de radiación en los alrededores de las centrales nucleares para generar alertas de fugas.	Obtener consejos en el punto de venta de acuerdo con los hábitos del cliente, las preferencias, la presencia de componentes alérgicos para ellos o las fechas de vencimiento.	Control de temperatura dentro de refrigeradores industriales y médicos con mercancía sensible.	Irrigación selectiva en zonas secas para reducir los recursos hídricos requeridos en el green.	Detección de aperturas y violaciones de ventanas y puertas para evitar intrusos.
Detección de teléfonos inteligentes	Prevención de derrumbes y avalanchas	Flujo de agua	Gases explosivos y peligrosos	Gestión inteligente de productos	Presencia de ozono	Redes de estaciones meteorológicas	Preservación de arte y mercancías
Detecta dispositivos iPhone y Android y, en general, cualquier dispositivo que funcione con interfaces WiFi o Bluetooth.	Monitoreo de la humedad del suelo, vibraciones y densidad de la tierra para detectar patrones peligrosos en las condiciones de la tierra.	Medición de la presión del agua en los sistemas de transporte de agua.	Detección de niveles de gas y fugas en entornos industriales, entornos de fábricas químicas y dentro de minas.	Control de rotación de productos en estanterías y bodegas para automatizar procesos de reabastecimiento.	Monitoreo de los niveles de ozono durante el proceso de secado de carne en las fábricas de alimentos.	Estudio de las condiciones climáticas en los campos para pronosticar la formación de hielo, la lluvia, la sequía, la nieve o los cambios de viento.	Monitoreo de condiciones dentro de museos y almacenes de arte.
Niveles de campo electromagnético	Detección temprana de terremotos	Cálculo de Stock de Silos		Calidad de las condiciones de envío	Localización interna	Compost	Detección de caídas

Internet de las Cosas

Medición de la energía radiada por las estaciones celulares y los enrutadores WiFi.	Control distribuido en lugares específicos de temblores.	Medición del nivel de vacío y peso de los bienes.	Monitoreo de vibraciones, golpes, aperturas de contenedores o mantenimiento de la cadena de frío para fines de seguros.	Ubicación interna del activo mediante el uso de etiquetas activas (ZigBee) y pasivas (RFID / NFC).	Control de los niveles de humedad y temperatura en alfalfa, heno, paja, etc. para prevenir hongos y otros contaminantes microbianos.	Asistencia para personas mayores o discapacitadas que viven de forma independiente.
Congestión del tráfico	Monitoreo de agua potable		Ubicación del artículo	Autodiagnóstico del vehículo	Hidroponía	Refrigeradores médicos
Monitoreo de vehículos y niveles peatonales para optimizar las rutas de manejo y peatonales.	Control de la calidad del agua de la canilla en las ciudades.		Búsqueda de artículos individuales en grandes superficies como almacenes o puertos.	Recopilación de información de CANBus para enviar alarmas en tiempo real a emergencias o brindar asesoramiento a los conductores.	Control de las condiciones exactas de las plantas cultivadas en agua para obtener los cultivos de mayor eficiencia.	Control de las condiciones dentro de los congeladores almacenando vacunas, medicamentos y elementos orgánicos.
Iluminación inteligente	Detección de fugas químicas en ríos		Detección de incompatibilidad de almacenamiento		Cuidado de crías	Cuidado de los deportistas
Iluminación inteligente y adaptable al clima en luminarias.	Detecta fugas y desechos de fábricas en los ríos.		Emisión de advertencia en contenedores que almacenan productos inflamables		Control de las condiciones de crecimiento de las crías en granjas de animales para	Vigilancia de signos vitales en centros y campos de alto rendimiento.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

		cerrados a otros que contienen material explosivo.	asegurar su supervivencia y salud.	
Gestión de residuos	Medición remota de la piscina	Seguimiento de flotas	Seguimiento de animales	Vigilancia de pacientes
Detección de niveles de basura en contenedores para optimizar las rutas de recolección de basura.	Control remoto de las condiciones de la piscina.	Control de rutas seguidas para productos delicados como medicinas, joyas o mercancías peligrosas.	Ubicación e identificación de animales pastando en pastos abiertos o en grandes establos.	Monitoreo de las condiciones de los pacientes dentro de los hospitales y en el hogar de ancianos.
Carreteras inteligentes	Niveles de contaminación en el mar		Niveles de gases tóxicos	Radiación ultravioleta
Autopistas inteligentes con mensajes de advertencia y desvíos según las condiciones climáticas y eventos inesperados como accidentes o congestiones.	Control de las filtraciones en tiempo real y los desechos en el mar.		Estudio de ventilación y calidad del aire en granjas y detección de gases nocivos de excrementos.	Medición de los rayos solares UV para advertir a las personas que no se expongan en determinadas horas.
	Fugas de agua			

Internet de las Cosas

Detección de presencia de líquido fuera de los tanques y variaciones de presión a lo largo de las tuberías.
Inundaciones fluviales
Monitoreo de las variaciones del nivel de agua en ríos, presas y embalses.

Tabla 1. Aplicaciones de Internet de las Cosas. Adaptado de (Libelium, 2018)

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

2.5. BANDA LICENCIADA VS. NO LICENCIADA

Una amplia gama de fabricantes de tecnologías se encuentra disponibles en el espacio LPWAN. Como se mencionó brevemente una de las diferencias fundamentales entre estas tecnologías es el tipo de espectro de radio que estas se utilizan las mismas.

Las tecnologías que se implementan en el espectro ISM (no licenciado) se benefician del recurso de espectro gratuito, pero poseen más limitaciones en términos de capacidad y soporte para seguridad de aplicaciones (Machina Research, 2015).

En la Tabla 2 se presentan las principales tecnologías LPWAN actuales según los dos tipos de licenciamiento, mientras que en la Tabla 3 se presenta un análisis comparativo de pros y contras de cada uno.

Licenciado	No Licenciado
	

Tabla 2. Principales tecnologías LPWAN.(Adelantado et al., 2017)

	Licenciado	No Licenciado
Pros	Transparencia de Calidad de Servicio, autenticación, seguridad Potencial para reutilizar los existentes sitios celulares Tiempo rápido para cubrir la cobertura	Tiempos de despliegue más rápidos. Opción para nuevos operadores, proveedores de servicios especializados. Mejor opción para operadores pequeños. Ecosistemas en desarrollo Sin sorpresas de costo
Contras	Dependencia de los operadores móviles El servicio en general es más costoso. Costo del espectro	Seguridad remota limitada Pocos actores tecnológicos

Tabla 3. Tabla comparativa de pros vs contras según tipo de espectro. (Machina Research, 2015)

2.6. CONCLUSIONES

Internet de las Cosas

Se puede concluir que es de particular interés el análisis de las redes del tipo no licenciado dado que al operar en un espectro que no requiere previa asignación de espectro, permite mucha más flexibilidad para el despliegue de nuevas aplicaciones.

3. PROTOCOLO LORAWAN

3.1. RESEÑA DEL CAPÍTULO

En este capítulo se ofrece un análisis en profundidad del protocolo de Internet de las cosas en banda no licenciada llamado LoRaWAN, pasando por sus orígenes y estructura topológica. Se presenta detalladamente cada uno de los componentes de una red LoRa para después hacer una revisión detallada de su esquema de modulación de radiofrecuencia y corrección de errores. Utilizando las bases anteriores se presenta un análisis de la capacidad de un enlace. Posteriormente avanzando al nivel de control de acceso al medio (MAC) se realiza un análisis detallado de la trama para poder así determinar los tiempos de utilización del espectro y los costos en *overheads* de los algoritmos de corrección de errores. Por último, se presentan los diferentes tipos de dispositivos y formas de comunicación de en una red LoRa, dejando así sentadas las bases teóricas para los futuros capítulos.

3.2. INTRODUCCIÓN

La historia de LoRaWAN comienza en 2009 cuando los fundadores de la empresa Cycleo comenzaron a desarrollar una tecnología de modulación de baja energía y largo alcance con el objetivo de permitir comunicaciones inalámbricas con medidores de electricidad, gas y agua.

En 2012 la empresa Semtech adquirió las operaciones de Cycleo y comenzó el desarrollo del primer chip LoRa para dispositivos finales (SX1272) y para *gateways* o puertas de enlace (SX1301). Al mismo tiempo creó un protocolo propietario llamado LoRaMAC, en el cual se especificaban los formatos de los mensajes, así como la seguridad de los mismos (Slats, 2020).

En 2015 algunas empresas (IBM, MicroChip, Cisco, Semtech, Bouygues Telecom, Singtel, KPN, Swisscom, Fastnet and Belgacom) decidieron formar la alianza LoRa con el objetivo de proveer un marco de estandarización e interoperabilidad entre los diferentes fabricantes de tecnología así como promover la adopción de LoRaWAN como principal protocolo de redes LPWAN (LoRa Alliance, 2022).

Fundamentos de la red LoRaWAN

Para comprender completamente las redes LoRaWAN, comenzaremos con un vistazo a las diferentes tecnologías que se ven involucradas en el protocolo. Como se muestra en la Figura 4, LoRa es la capa física (PHY), es decir, la modulación inalámbrica utilizada para crear el enlace de comunicación de largo alcance. LoRaWAN es un protocolo de red abierto que ofrece servicios

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

seguros de comunicación bidireccional, movilidad y localización, es mantenidos por LoRa Alliance. (LoRa and LoRaWAN, n.d.)

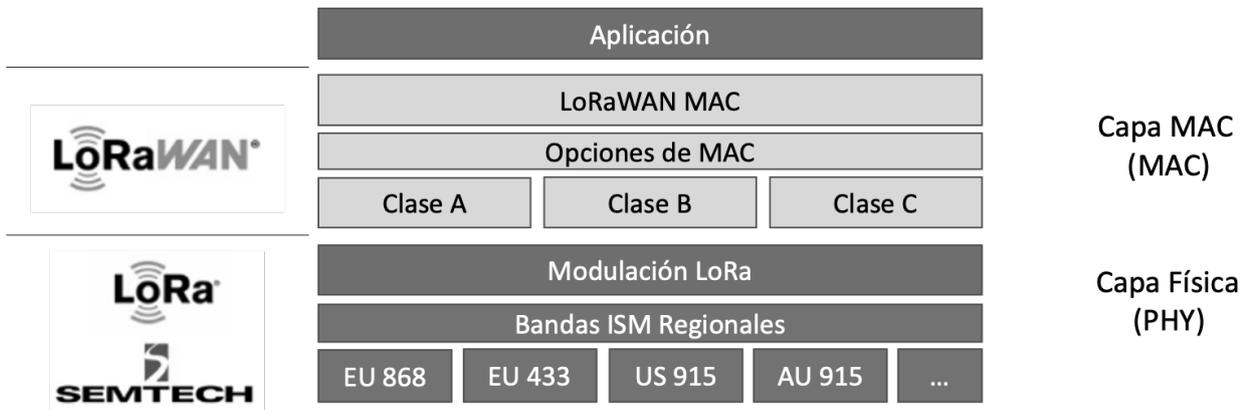


Figura 4. Stack del protocolo LORAWAN. Adaptado de Semtech.com

Las redes LoRaWAN generalmente se presentan en una topología de estrella de estrellas en la que los gateways transmiten mensajes entre los dispositivos finales y un servidor de red central tal como se representa en Figura 5. El servidor de red enruta los paquetes desde cada dispositivo de la red al servidor de aplicaciones asociado. Para proteger las transmisiones de radio, el protocolo LoRaWAN utiliza criptografía simétrica. En el backend, el almacenamiento de las claves simétricas de los dispositivos es realizado en un servidor, llamado servidor de unión.

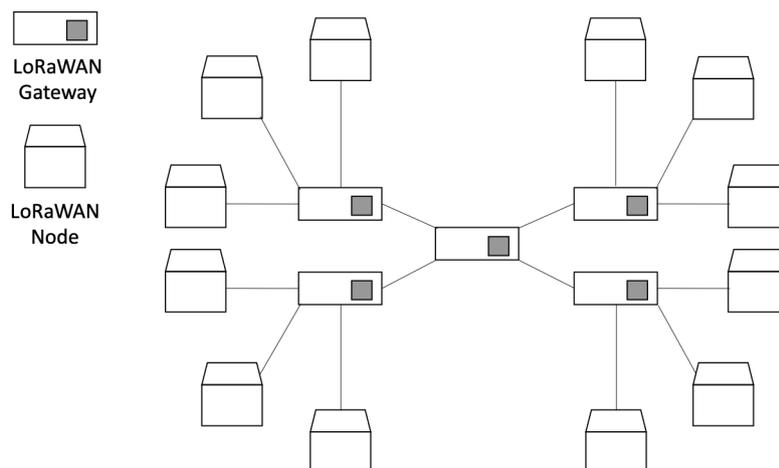


Figura 5. Topología estrella de estrellas LoRaWAN representando la ubicación de gateways y nodos. Adaptado de (Riegsecker, 2017)

Protocolo LoRaWAN

Existen tres tipos de servidores en la arquitectura de LoRaWAN, el servidor de red, el servidor de aplicaciones y el servidor de unión tal como se representa en la Figura 6, si bien estos tres servidores típicamente están co-localizados físicamente, existe la posibilidad según la arquitectura de red de que estén en diferentes ubicaciones físicas. Existen cuatro interfaces definidas entre los componentes, la JS-AS para distribuir la clave de sesión de la aplicación entre el servidor de unión y el de aplicación. La hNS-JS para el procedimiento de activación entre el servidor de unión y el servidor de red. La AS-hNS que permite la entrega de la carga útil de datos entre el servidor de red y el servidor de aplicación, así como también sus metadatos. Por último la ED-NS define la señalización de capa física de LoRaWAN y el protocolo de entrega de datos entre el dispositivo final y el servidor de red.

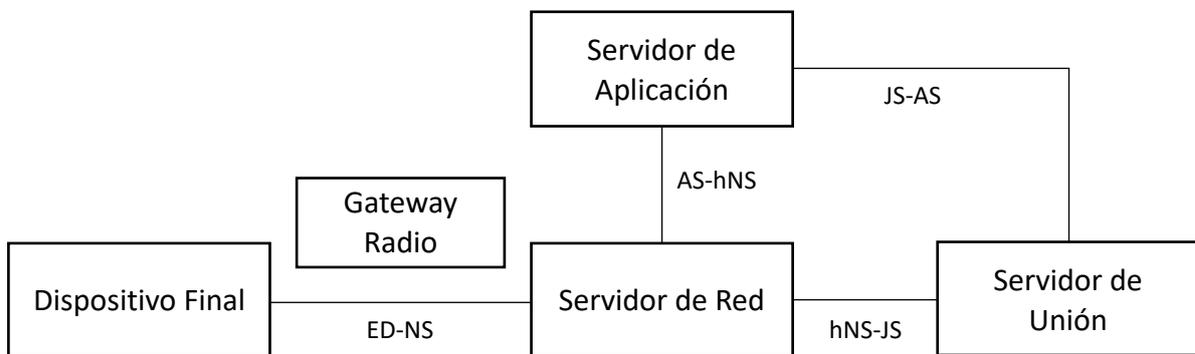


Figura 6. Arquitectura lógica de referencia de LoRaWAN. Elaboración Propia basado en (*LoRaWAN Specification 1.1*, 2017)

Los *gateways* se conectan al servidor de red a través de conexiones IP, mientras que los dispositivos finales utilizan comunicación LoRa. Las comunicaciones son generalmente bidireccionales, aunque a nivel de cantidad de datos sea predominantemente la comunicación de subida de datos desde un dispositivo final hacia el servidor de red, mientras que en el sentido de bajada sea solo comunicación de comando y control (LoRa Alliance, 2017).

3.3. ELEMENTOS DE UNA RED LORAWAN

Una red típica LoRaWAN consta de los siguientes componentes básicos según se representa en la Figura 7.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

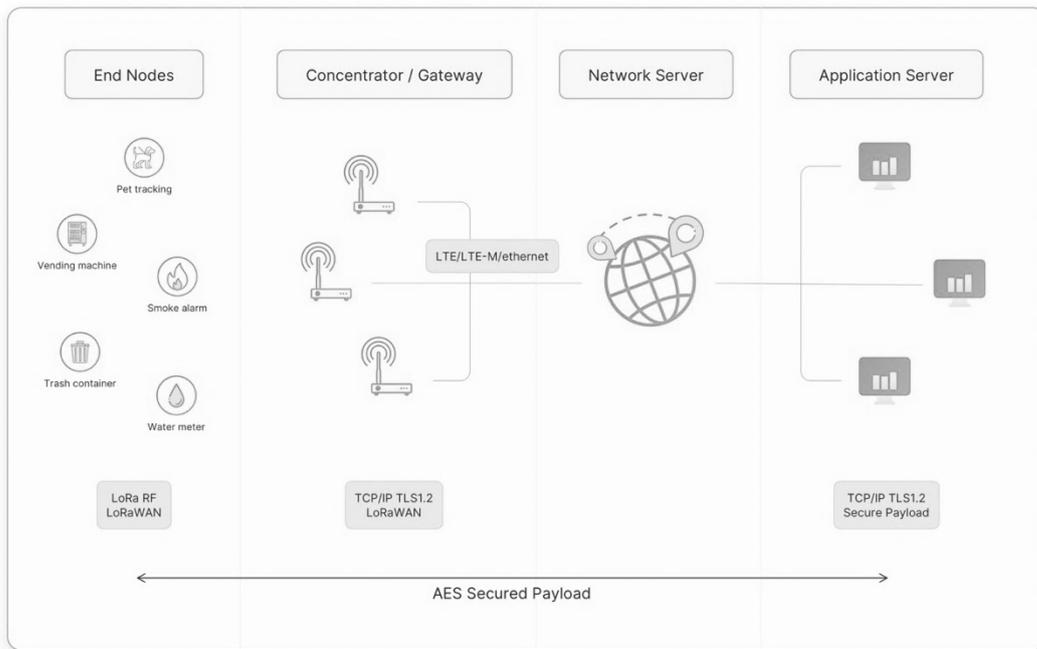


Figura 7. Arquitectura de referencia de LoRaWAN mostrando los cuatro niveles de la red y los protocolos de comunicación entre ellos. Recuperado de The Things Network

Dispositivos finales: los sensores envían mensajes inalámbricos a los *gateways* o reciben mensajes de forma inalámbrica desde los mismos.

Gateways: recibe mensajes desde dispositivos finales y los reenvía al servidor de red utilizando protocolo IP.

Servidor de red: es un software que se ejecuta en un servidor que administra los dispositivos de toda la red.

Servidores de aplicaciones: es un software que se ejecuta en un servidor que es responsable de recibir y procesar los datos de las aplicaciones

Servidor de unión: es un software que se ejecuta en un servidor que procesa los mensajes de solicitud de unión enviados por los dispositivos finales.

Los dispositivos finales se comunican con *gateways*, cada *gateway* está conectado a un servidor de red. Las redes LoRaWAN utilizan un protocolo basado en ALOHA, por lo que los dispositivos finales pueden aparearse con cualquier puerta de enlace. Los mensajes enviados por un dispositivo final pueden ser recibidos por uno o varios *gateways*, los cuales los reenvían al servidor de red. Si

el servidor de red ha recibido varias copias del mismo mensaje, conserva una sola copia y descarta las demás. Esto se conoce como deduplicación de mensajes (The Things Network, 2022).

3.3.1. DISPOSITIVO FINAL

Un dispositivo final LoRaWAN puede ser un sensor, un actuador o ambos. Suelen funcionar con pilas y/o energía solar. Estos dispositivos finales están conectados de forma inalámbrica a la red LoRaWAN a través de puertas de enlace que utilizan la modulación LoRa RF.

En la mayoría de las aplicaciones, un dispositivo final es un sensor autónomo, a menudo operado por baterías, que digitaliza las condiciones físicas y los eventos ambientales. Los casos de uso típicos para un actuador incluyen: alumbrado público, cerraduras inalámbricas, cierre de válvula de agua, prevención de fugas, entre otros.

Cuando se fabrican, a los dispositivos basados en LoRa se les asigna identificadores únicos. Estos identificadores se utilizan para activar y administrar el dispositivo de forma segura, para garantizar el transporte seguro de paquetes a través de una red privada o pública y para entregar datos encriptados a la nube. (*LoRa and LoRaWAN*, n.d.)

3.3.2. GATEWAY

Cada *gateway* se registra en un servidor de red. Un *gateway* recibe mensajes LoRa de dispositivos finales y simplemente los reenvía al servidor de red LoRaWAN. Los Gateway se conectan al servidor de red mediante el protocolo TCP/IP mediante un *backhaul* como celular (3G/4G/5G), WiFi, Ethernet, fibra óptica, HFC o enlaces de radio de 2,4 GHz.

Los *gateways* LoRaWAN se pueden clasificar en puertas de enlace interiores (picocelda) y exteriores (macrocelda).

Los *gateways* para interiores son apropiados para brindar cobertura en lugares como ubicaciones internas (espacios cerrados por varias paredes), sótanos y edificios de varios pisos.

Los *gateways* exteriores brindan una mayor cobertura que los interiores. Son adecuados para dar cobertura tanto en zonas rurales como urbanas. Estos *gateway* se pueden montar en torres de telefonía celular, los techos de edificios, mástiles, nodos de redes HFC, etc. Por lo general, un *gateway* exterior tiene una antena externa conectada mediante un cable coaxial.

Por lo general, la sensibilidad del receptor de un *gateway* exterior es más alta que la sensibilidad del receptor de uno interior (*LoRa and LoRaWAN*, n.d.).

3.3.3. SERVIDOR DE RED

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

El servidor de red administra los *gateways*, dispositivos finales, aplicaciones y usuarios en toda la red LoRaWAN.

Un servidor de red LoRaWAN típico tiene las funciones listadas en la Tabla 4.

Establecimiento de conexiones seguras AES de 128 bits para el transporte de mensajes entre los dispositivos finales y el servidor de aplicaciones (seguridad de extremo a extremo).
Validación de la autenticidad de los dispositivos finales y la integridad de los mensajes.
Deduplicación de mensajes ascendentes.
Selección del mejor <i>gateway</i> para enviar los mensajes descendentes.
Envío de comandos ADR para optimizar la tasa de datos de los dispositivos.
Comprobación de la dirección del dispositivo.
Proporcionar reconocimientos de mensajes de datos de enlace ascendente confirmados.
Reenvío de datos ascendentes a los servidores de aplicaciones apropiados
Reenvío de solicitudes de unión al servidor de unión
Responder a todos los comandos de la capa MAC.

Tabla 4. Funciones de un servidor de red LoRaWAN

3.3.4. SERVIDOR DE APLICACIONES

El servidor de aplicaciones procesa los mensajes de datos de aplicación recibidos de los dispositivos finales. También genera todas las cargas útiles de enlace descendente de la capa de aplicación y las envía a los dispositivos finales conectados a través del servidor de red. Una red LoRaWAN puede tener más de un servidor de aplicaciones.

3.3.5. SERVIDORES DE UNION

El servidor de unión ayuda en la activación segura de dispositivos, el almacenamiento de claves y la generación de claves de sesión.

El dispositivo final inicia el procedimiento de entrada enviando el mensaje de solicitud de incorporación al servidor de unión a través del servidor de red. El servidor de unión procesa el mensaje de solicitud de unión, genera claves de sesión y transfiere las claves de sesión al servidor de red y al servidor de aplicaciones, respectivamente.

A tal fin, el servidor de unión debe contener la siguiente información para cada dispositivo final bajo su control: DevEUI (identificador único de serie del dispositivo final), AppKey (clave de encriptación de aplicaciones), NwkKey (clave de encriptación de red), identificador del servidor de aplicaciones y perfil de servicio de dispositivo final.

3.4. FUNDAMENTOS OPERACIONALES UNA RED LORAWAN

La comunicación entre los dispositivos finales y *gateways* se distribuye en diferentes canales de frecuencia y velocidades de datos. La selección de la tasa de datos es un compromiso entre el rango de comunicación y la duración del mensaje, las comunicaciones con diferentes tasas de datos no interfieren entre sí. Las velocidades de datos de LoRa varían de 0,3 kbps a 50 kbps. Para maximizar la duración de la batería de los dispositivos finales y la capacidad general de la red, la red LoRa, la infraestructura puede administrar la velocidad de datos y la salida de RF para cada dispositivo final individualmente mediante un esquema de velocidad de datos adaptable (ADR).

LoRa se destaca por haber sido diseñado para uso de la banda de frecuencia ISM de 900 MHz no licenciada. Esto representa una variación del paradigma del uso de redes celulares, que requieren que los operadores de red otorguen licencias de espectro exclusivo y costoso para operar a mayor potencia y ciclos de trabajo más largos o continuos en canales protegidos. Al cumplir con los estatutos regulatorios permisivos, la implementación de una infraestructura LoRa e capaz de proporcionar varios kilómetros de cobertura en una red pública o privada pronto puede ser tan accesible como la implementación de una red 802.11.

En comparación con otras técnicas de modulación, la técnica de modulación de espectro expandido (*spread spectrum*) utilizada en LoRaWAN asegura un mayor alcance de los enlaces, así como una mejor inmunidad a las interferencias. LoRaWAN utiliza un ancho de banda generalmente de 125 kHz, para transmitir la señal aunque también permite el uso de ancho de banda escalable entre 125 kHz, 250 kHz o 500 kHz (Reynders et al., 2016), el uso de una banda más ancha hace que LoRaWAN sea resistente al ruido, a los efectos del *doppler*, a las variaciones a largo plazo de los osciladores y al fading. Sin embargo, la utilización de una señal de banda angosta en una banda mucho más ancha hace que se utilice el espectro de manera menos eficiente a menos que se logre una generación de señales perfectamente ortogonales entre los diferentes transmisores (Noreen et al., 2017).

El transmisor genera señales variando su frecuencia a lo largo del tiempo y manteniendo constante la fase entre símbolos adyacentes. La señal transmitida es una señal similar al ruido que es resistente al *fading multipath* y al doppler, robusta a las interferencias. El receptor puede

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

decodificar incluso una señal muy atenuada de 20 dB por debajo del nivel de ruido (Semtech, 2013)

LoRa es la implementación de capa física (PHY) o "bits", como se define en el Modelo de red de siete capas OSI, que se muestra en la Figura 8. En lugar de cableado, el aire se utiliza como medio para transportar ondas de radio LoRa desde un transmisor de RF en un dispositivo IoT a un receptor en una puerta de enlace, y viceversa.

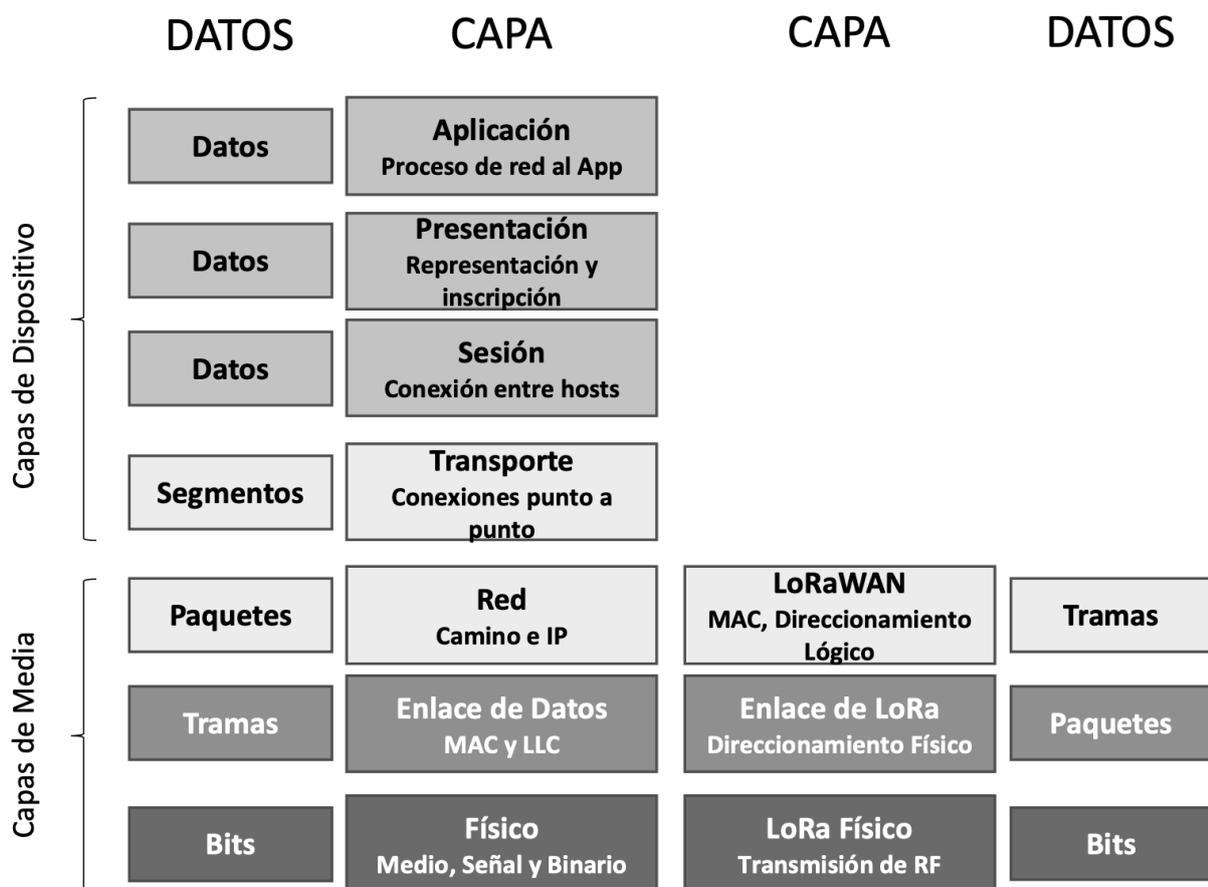


Figura 8. Mapeo de LoRaWAN al modelo OSI. Adaptado de (LoRa and LoRaWAN, n.d.)

3.4.1. MODULACIÓN FÍSICA DSSS

En un sistema de espectro ensanchado de secuencia directa (DSSS), la fase portadora de la señal del transmisor cambia de acuerdo con una secuencia de código como se muestra en la Figura 9. Al multiplicar la señal de datos con un patrón de bits predefinido a una velocidad mucho mayor, también conocido como código de propagación (o secuencia de chip), se crea una señal

Protocolo LoRaWAN

"más rápida" que tiene componentes de frecuencia más altos que la señal de datos original como se ve en la Figura 10. Esto significa que el ancho de banda de la señal se extiende más allá del ancho de banda de la señal original. En la terminología de RF, los bits de la secuencia de código se llaman chips (para distinguir entre los bits más largos y no codificados de la señal de datos original). Cuando la señal transmitida llega al receptor de RF, se multiplica con una copia idéntica del código de propagación utilizado en el transmisor de RF, lo que resulta en una réplica de la señal de datos original. (Torrieri, 2018)

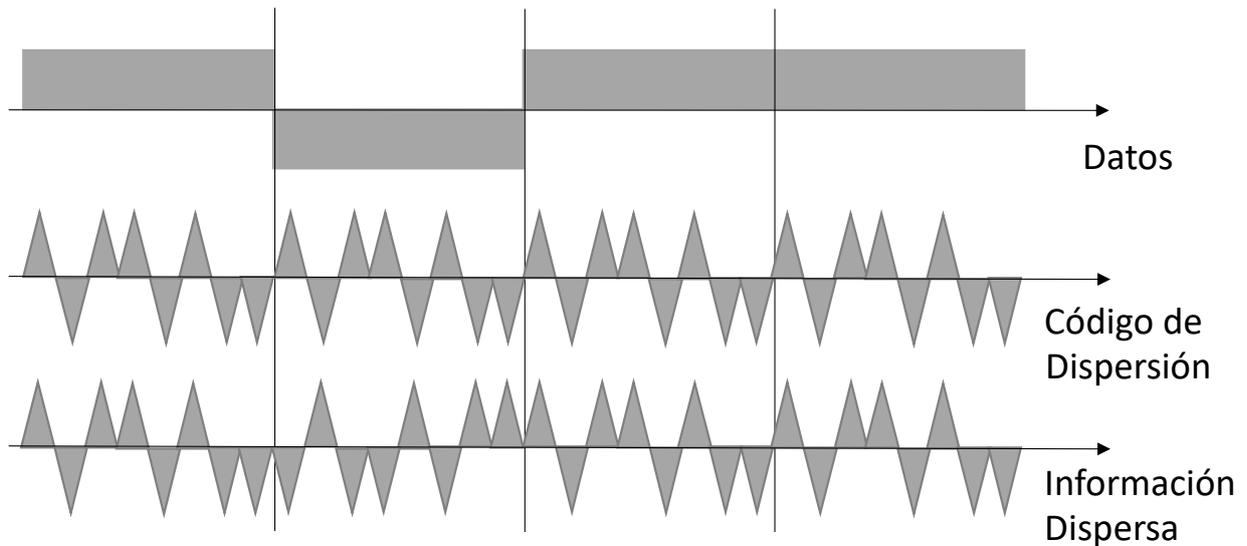


Figura 9. Modulación con códigos. Elaboración Propia

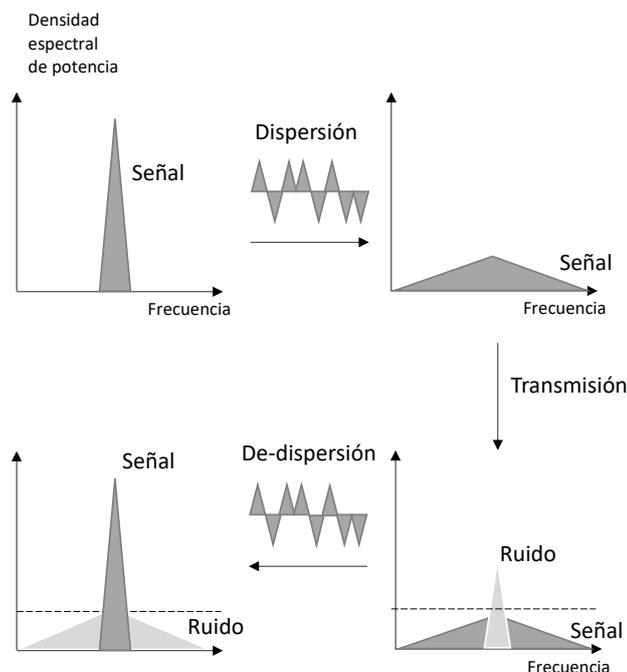


Figura 10. Ejemplo del comportamiento en frecuencia y tiempo de una Modulación con espectro ensanchado y su robustez frente al ruido. Elaboración Propia

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Pasar por esta multiplicación de secuencia de código genera un presupuesto de enlace de RF más alto, por lo que puede transmitir a una distancia más largo. La relación de la velocidad de chip de la secuencia de código y la velocidad de bits de la señal de datos se denomina ganancia de procesamiento (G_p). Esta ganancia es lo que permite al receptor recuperar la señal de datos original, incluso si el canal tiene una relación señal-ruido (SNR) negativa. LoRa tiene un G_p superior en comparación con la modulación de tecla de desplazamiento de frecuencia (FSK), lo que permite un nivel de potencia de salida del transmisor reducido mientras se mantiene la misma velocidad de datos de señal y un presupuesto de enlace similar. Una de las desventajas de un sistema DSSS es el hecho de que requiere un reloj de referencia altamente preciso. En la modulación LoRa, la modulación del espectro de la señal se logra generando una señal de *chirp* que varía continuamente en frecuencia, como se muestra en la Figura 11.

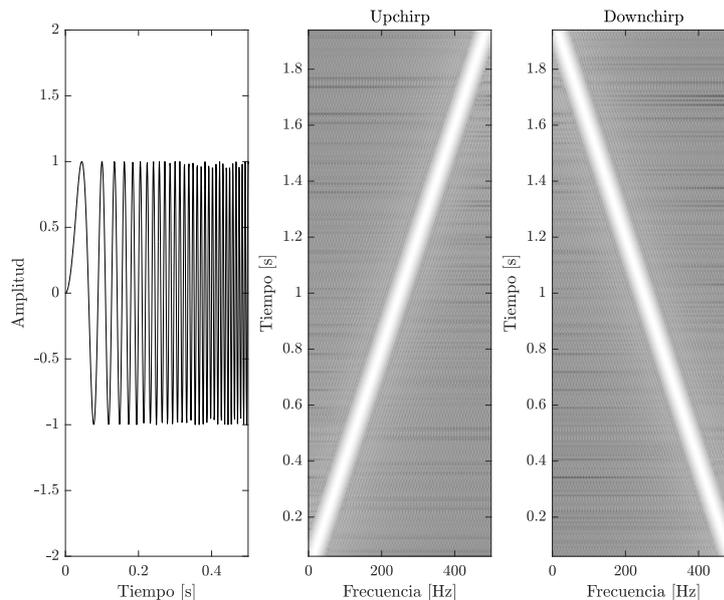


Figura 11. Modulación por *Chirps*. Elaboración Propia

Como se señaló anteriormente, la ganancia de procesamiento LoRa se introduce en el canal de RF multiplicando la señal de datos con un código de propagación o una secuencia de chip. Al aumentar la velocidad del chip, aumentamos los componentes de frecuencia del espectro de señal total. 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 relación señal-ruido (SNR) más baja. La cantidad de código de propagación que es aplicado a la señal original es llamado factor de propagación (SF). La modulación LoRa tiene un total de seis factores de propagación (SF7 a

SF12). Cuanto mayor sea el factor de propagación utilizado, más lejos podrá viajar la señal y aun así ser recibida sin errores por el receptor de RF a costa de reducir cantidad de datos transferidos por unidad de tiempo tal como se ve en la Figura 12.

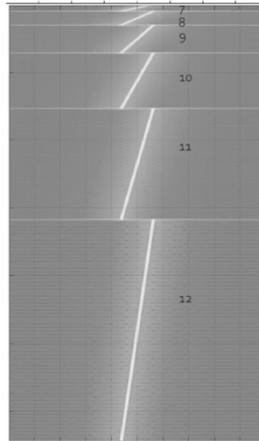


Figura 12. Factor de dispersión vs duración de *Chirp*. Adaptado de (Sakshama Ghoslya, 2018)

3.4.2. CORRECCIÓN DE ERRORES EN LORA

La técnica de corrección de errores que se utiliza en LoRaWAN para aumentar aún más la sensibilidad del receptor es del tipo FEC (*Forward Error Correction*). Particularmente mediante la utilización de un código Hamming de longitud ajustable (Seller, Olivier Bernard André & Sornin, Nicolas, n.d.). La tasa de código define la cantidad de FEC y LoRaWAN ofrece valores de CR entre 1 y 4. LoRaWAN utiliza tasas de código, $Coding Rate = \frac{4}{4+CR}$ o sea 4/5, 4/6, 4/7 y 4/8. Lo que significa que, si la tasa de código se denota como $k = n$, donde k representa información útil y el codificador genera n número de bits de salida, entonces $n - k$ serán los bits redundantes. La redundancia permite que el receptor detecte y corrija errores en el mensaje a costo de disminuir la velocidad de datos efectiva tal como se evidencia en la Tabla 5.

CR	1	2	3	4
<i>Coding Rate</i>	4/5	4/6	4/7	4/8
Eficiencia	0.8	0.666	0.571	0.5

Tabla 5. Tasas de Código para LoRaWAN. Adaptado de (Noreen et al., 2017).

En LoRaWAN se puede elegir un factor de dispersión variable (SF) en función de la relación señal a ruido recibida (SNR). Este factor de dispersión adapta la longitud del símbolo y al mismo tiempo, también especifica el número de bits por símbolo. Por lo tanto, al cambiar el factor de dispersión se obtiene una tasa de bits variable entre 366 bps para el factor de propagación más alto (SF=12)

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

y 48 kbps para el factor de propagación más bajo ($SF=12$) como se muestra en la Ecuación 1 donde BW es el ancho del Canal, SF es el factor de dispersión y R_b es la tasa de bits.

$$R_b = \frac{BW}{2^{SF}} * SF \text{ [bits/seg]}$$

Ecuación 1. Tasa de bits de LoRaWAN

Si bien la elección de un factor de dispersión más alto incrementa la tasa de transferencia en bits por otro lado reduce el máximo alcance de la transmisión y lo mismo en sentido inverso.

Cada símbolo se dispersa con código de dispersión de 2^{SF} chips de longitud. En el transmisor, el código de dispersión se subdivide en códigos de longitud $2^{SF} / SF$. Luego, cada bit del símbolo se dispersa utilizando el subcódigo. Por lo tanto, se necesitan 2^{SF} chips, para propagar un símbolo. Este código de dispersión también lo conoce el receptor. La sustitución de un símbolo por múltiples chips de información significa que el factor de expansión tiene una influencia directa en la velocidad de datos efectiva (Noreen et al., 2017), tal como se ve en la Figura 13.

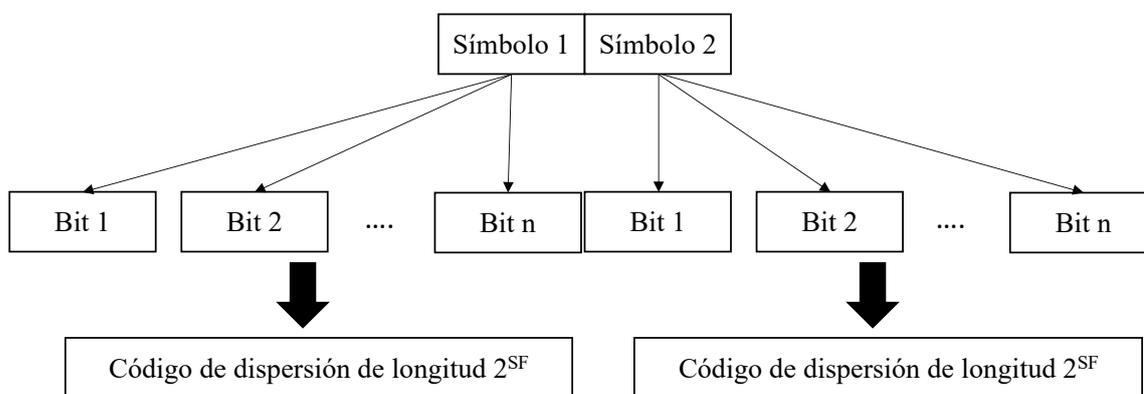


Figura 13. Dispersión de Símbolos en LoRaWAN. Elaboración Propia

La elección de una longitud más alta de código de dispersión mejora la distancia de transmisión, pero a costo de una tasa de bits más baja dada por el aumento del tiempo en el aire, esto se deduce de la aplicación del teorema de Shannon-Hartley que establece la tasa máxima posible de

información para un canal con ruido a determinado ancho de banda tal como se ve en la Ecuación 2 (Proakis, 2000).

$$C = B * \log_2 \left(1 + \frac{S}{N} \right) \text{ [bits/seg]}$$

Ecuación 2. Teorema de Shannon

Donde:

C = Capacidad del canal

B = Ancho de banda del canal

S = Potencia promedio de señal en el receptor

N = Potencia promedio de ruido en el receptor

S/N = Relación señal a ruido en el receptor

Por un lado, si se convierte la relación logarítmica de base 2 a base natural y además se supone que para una aplicación de espectro disperso (*spread spectrum*) la relación señal a ruido es pequeña y la potencia de señal es mucho menor que la de ruido entonces $S/N \ll 1$ y la Ecuación 2 se puede reescribir como la Ecuación 3

$$\frac{C}{B} = 1.43 * \frac{S}{N}$$

Ecuación 3. Teorema de Shannon simplificado para Spread Spectrum

Por lo tanto, de ahí se desprende que para transmitir información libre de errores a una relación señal ruido determinada es solo necesario incrementar el ancho de banda del canal para transmitir más información (Semtech, 2015).

Ahora bien, dado que el piso de ruido o también llamado ruido térmico de Johnson-Nyquist está determinado por el ancho de banda del canal según la Ecuación 4 (Lee & Miller, 1998).

$$N_{Floor} = 10 * \log_{10}(k_B * T * B * 1000) \text{ [dBm]}$$

Ecuación 4. Ruido de Johnson-Nyquist

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Donde:

k_B = Constante de Boltzman ($1.38 * 10^{-13} \text{ m}^2\text{kg/s}^2 \text{ K}$)

T = Temperatura [K]

B = Ancho de Banda del Canal [Hz]

1000 = Conversión de watt a mili watt

Suponiendo T=293 K y simplificando, se obtiene la $N_{Floor} = -174 + 10 * \log_{10}(B) \text{ dBm}$

Ecuación 5, donde se demuestra que el piso de ruido depende del ancho de banda del canal utilizado.

$$N_{Floor} = -174 + 10 * \log_{10}(B) \text{ [dBm]}$$

Ecuación 5. Ruido Johnson-Nyquist a temperatura ambiente

Si se desea obtener cual es la mínima señal detectable o sensibilidad (S) de un receptor, se debe considerar la figura de ruido de este.

La relación señal a ruido mínima requerida para una modulación dada puede ser despejada en la Ecuación 6.

$$S = -174 + 10 * \log_{10}(B) + NF + SNR \text{ [dBm]}$$

Ecuación 6. Sensibilidad de un receptor de radio

Si se desea determinar cuál será el presupuesto de enlace, el mismo está determinado por la Ecuación 7.

Presupuesto Enlace = Mínima Señal Detectable – Máxima Pot. de Transmision [db]

Ecuación 7. Cálculo de enlace

El alcance máximo está dado por el presupuesto de enlace considerando la atenuación de señal en el espacio libre (FSPL) según la Ecuación 8. (Lee & Miller, 1998).

$$FSPL = 20 * \log_{10}(d) + 20 * \log_{10}(f) + 20 * \log_{10} \left(\frac{4\pi}{c} \right) [dB]$$

Ecuación 8. Atenuación del espacio libre

3.5. FORMATO DE LA TRAMA FÍSICA LORA

La trama física o trama de radio de LoRa está formada por cuatro partes:

Preámbulo: se utiliza para mantener el receptor sincronizado con el flujo de datos entrante. El valor predeterminado es 12 de símbolos, 8 se utilizan en LoRaWAN. La longitud del preámbulo es variable y se puede establecer mediante configuración, por lo que la longitud del preámbulo se puede ampliar. La longitud del preámbulo del receptor debe ser la misma que la del transmisor. Si la longitud del preámbulo es desconocida o variable, la longitud del preámbulo del receptor debe establecerse en el valor máximo y ajustarse más tarde.

Cabecera: está formada por dos campos, el PHDR y el PHDR_CRC. Su función es describir información de la red, como el número de bytes en la carga útil, la velocidad de codificación y si se debe habilitar el CRC para la carga útil. Así como si se utiliza modo oculto o dominante. El modo dominante se utiliza en LoRaWAN. En el modo oculto, estos tres contenidos deben configurarse manualmente en ambos extremos de la comunicación.

Carga útil: Contiene la trama de la capa de acceso al medio LoRaWAN. El tamaño máximo de la carga útil es de 255 bytes.

CRC de la Carga útil: CRC de la Carga útil, 2 bytes.

El protocolo LoRaWAN establece un formato de paquete que se puede visualizar en la Figura 14, de este formato se desprende que cuanto más alta es la tasa de corrección de errores (CR), más largo será el paquete para una carga útil determinada.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

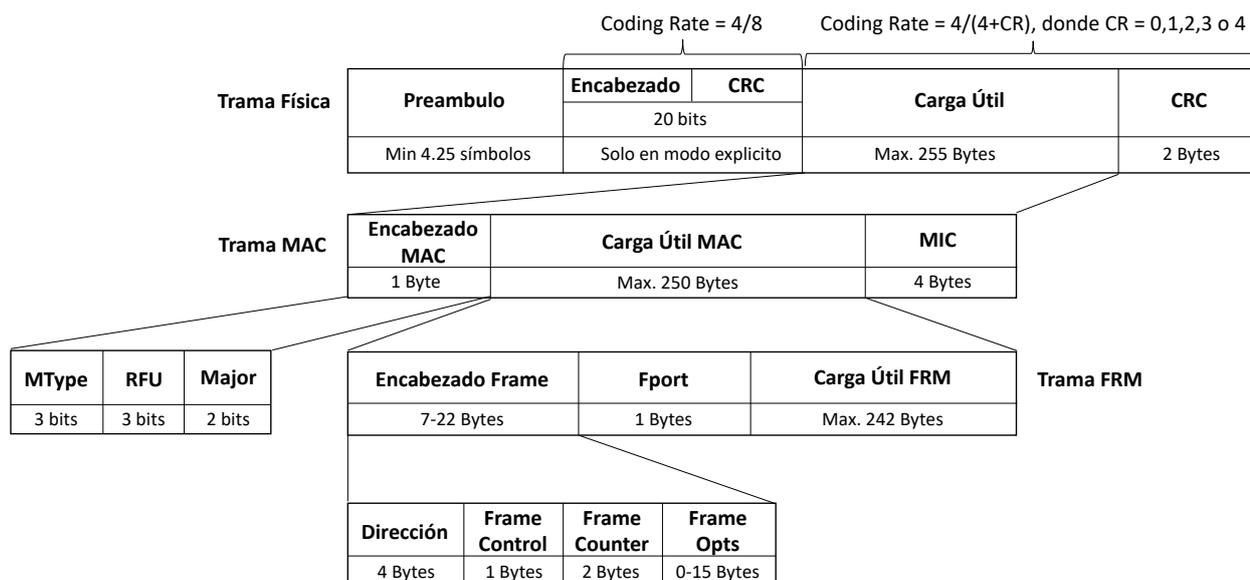


Figura 14. Estructura de paquete LoRaWAN. Elaboración propia

3.6. UTILIZACIÓN DEL TIEMPO DE AIRE EN LORA

El tiempo de aire de un paquete LoRaWAN viene dado por la Ecuación 9.

$$T_{Paquete} = T_{Preambulo} + T_{Carga Util} [seg]$$

Ecuación 9. Tiempo de Aire de LoRaWAN

El tiempo del preámbulo paquete LoRaWAN viene dado por la Ecuación 10

$$T_{Preambulo} = (n_{Preambulo} + 4.25) * T_s [seg]$$

Ecuación 10. Tiempo de Preámbulo de LoRaWAN

El tiempo de un símbolo de LoRaWAN viene dado por la Ecuación 11 y está relacionado con la tasa de símbolos

$$T_s = \frac{1}{R_s} [seg]$$

Protocolo LoRaWAN

Ecuación 11. Tiempo de Símbolo de LoRaWAN

La tasa de símbolos LoRaWAN viene dado por la Ecuación 12 y está relacionada con el ancho de banda del canal (BW) y el factor de dispersión (SF).

$$R_s = \frac{BW}{2^{SF}}$$

Ecuación 12. Tasa de Símbolo de LoRaWAN

El tiempo total de la carga útil de LoRaWAN viene dado por la Ecuación 13 y está relacionada con el ancho de banda del canal (BW) y el factor de dispersión (SF).

$$T_{Carga\ Útil} = PL_{Symb} * T_s$$

Ecuación 13. Tiempo de Carga Útil de LoRaWAN

La longitud en bytes de la carga útil de LoRaWAN viene dada por la Ecuación 14 donde:

CRC – 1 si el chequeo de redundancia cíclica esta activado

IH – 1 si los encabezados implícitos no están activados

DE – 1 si la optimización de bajo ancho de banda está habilitada

$$PL_{Symb} = 8 + \max \left(\text{ceil} \left[\frac{8PL - 4SF + 28 + 16CRC - 20IH}{4(SF - 2DE)} \right] (CR + 4), 0 \right)$$

Ecuación 14. Longitud de Carga Útil de LoRaWAN

El tiempo total de un paquete de LoRaWAN viene dado por la Ecuación 15 que deriva de la Ecuación 13, Ecuación 10, Ecuación 11 y Ecuación 12 y está relacionada con el ancho de banda del canal (BW) y el factor de dispersión (SF).

Se representa el comportamiento del tiempo total de aire de un paquete para los diferentes factores de dispersión en la Figura 15.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

$$T_{Paquete} = (n_{Preambulo} + 4.25 + PL_{Symb}) * \frac{2^{SF}}{BW}$$

Ecuación 15. Tiempo de paquete de LoRaWAN

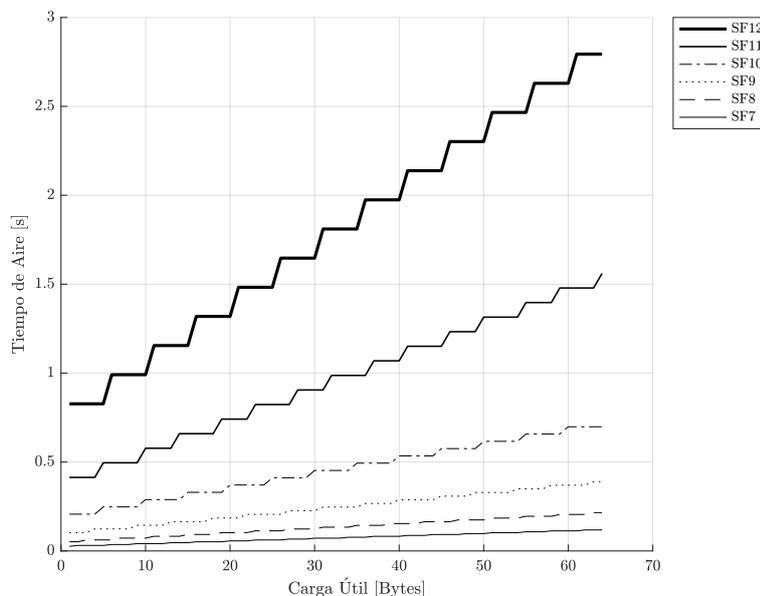


Figura 15. Tiempo de Aire vs Carga Útil en LoRaWAN. Elaboración Propia.

Considerando los puntos anteriores se puede resumir para un sistema LoRaWAN que el tiempo en el aire de un paquete se multiplica en forma exponencial para valores de factor de dispersión más altos según se ve en la Tabla 6.

SF	2^{SF}
7	128
8	256
9	512
10	1024
11	2048
12	4096

Tabla 6. Factor de Dispersión vs Longitud de Chip.

Así mismo utilizando en la Tabla 7 se visualiza el presupuesto de enlace para diferentes factores de dispersión, basados en la ganancia de 2.5 dB por cada paso de factor de dispersión. Para la

Protocolo LoRaWAN

elaboración de dicha tabla de utilizó una figura de ruido de receptor típica de 6 dB tal como se especifica en los manuales del receptor y una potencia de transmisión máxima de 14 dBm tal como se especifica para LoRaWAN 2.0. (Semtech, 2013)

SF	chips/symbol	SNR Límite [dB]	Figura de Ruido [dB]	BW [Hz]	Sensibilidad [dBm]	Potencia de TX [dBm]	Presupuesto de Enlace [dB]
7	128	-7.5	6	125000	-124.53	14.00	-138.53
8	256	-10	6	125000	-127.03	14.00	-141.03
9	512	-12.5	6	125000	-129.53	14.00	-143.53
10	1024	-15	6	125000	-132.03	14.00	-146.03
11	2048	-17.5	6	125000	-134.53	14.00	-148.53
12	4096	-20	6	125000	-137.03	14.00	-151.03

Tabla 7. Factor de dispersión vs Sensibilidad y Presupuesto de Enlace

3.7. CLASIFICACIÓN DE LOS DISPOSITIVOS LORAWAN

En LoRaWAN, el estudio del consumo de energía en dispositivos finales es muy importante. Hay tres tipos de clases de dispositivos utilizados para ampliar la duración de la batería: Clase A, Clase B y Clase C.

Los dispositivos utilizan acceso ALOHA, sin contención ni detección de portadora, para el enlace de subida. Después de enviar un paquete, un dispositivo clase A escucha esperando una respuesta desde el *gateway* durante dos ventanas recepción. Cada una de estas ventanas de recepción está definida por una duración, un tiempo de desplazamiento y una velocidad de datos. El valor típico para cada ventana de recepción es 1 seg. y 2 seg., respectivamente. Solo es posible transmitir datos desde el *gateway* al nodo después de una transmisión ascendente exitosa según se representa en la Figura 16.

La tasa de datos utilizado en la primera ventana de bajada desde el *gateway* al nodo se calcula como una función de la velocidad de datos de enlace. En la segunda ventana la tasa de datos es la mínima, 0.3 kbps.

En clase A no se puede transmitir tráfico de bajada hasta que no se realizó una transmisión de subida exitosa y la misma fue decodificada por el *gateway*.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

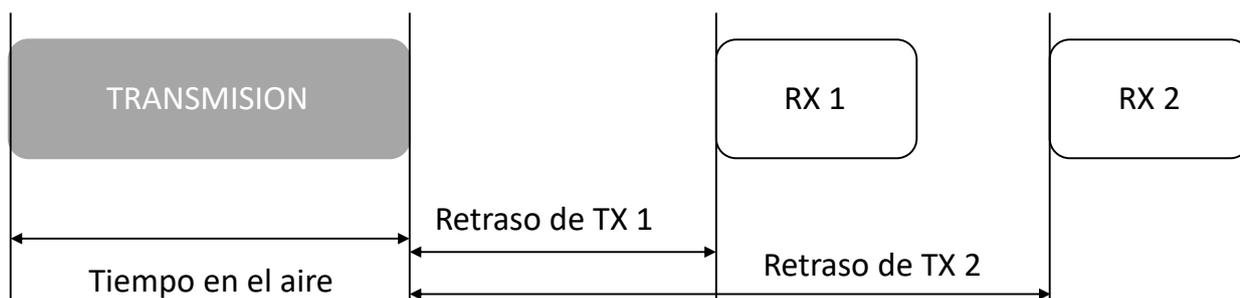


Figura 16. Contención en LoRaWAN. Elaboración Propia

La segunda ventana de recepción es deshabilitada cuando el tráfico de bajada es recibido por el nodo en la primera ventana exitosamente.

- Clase A. Los dispositivos de Clase A son dispositivos bidireccionales con la mayor eficiencia energética; la mayoría de las veces están en modo suspendido. La transmisión ascendente es seguida por dos ventanas descendentes predeterminadas en un corto período de tiempo. Se utilizan para aplicaciones que no requieren la recepción continua de datos y, de forma predeterminada, todos los dispositivos vienen configurados como Clase A. La Figura 17 muestra el tipo de transmisión de los dispositivos de Clase A.

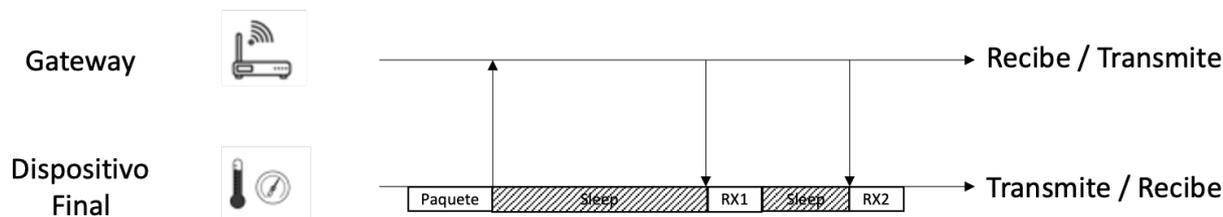


Figura 17. Transmisión de un dispositivo Clase A. Elaboración Propia

- Clase B. Los dispositivos de clase B son dispositivos bidireccionales con ventanas de recepción pre-programadas y que pueden abrir ventanas de enlace de recepción adicionales, la sincronización de ventas se produce por *beacons* transmitidos por la puerta de enlace. Estos dispositivos tienen un consumo de energía superior a los clase A. La Figura 18 muestra el tipo de transmisión de los dispositivos de Clase B.

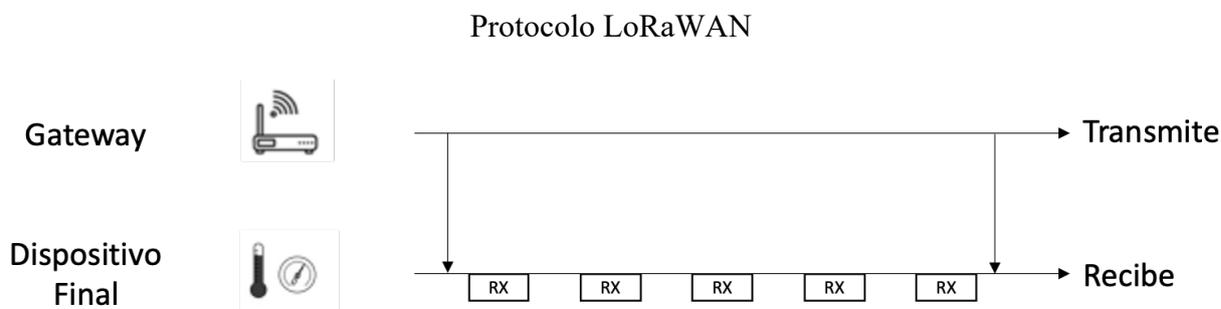


Figura 18. Transmisión de un dispositivo Clase B. Elaboración Propia

- Clase C. Los dispositivos clase C son dispositivos bidireccionales con un tiempo de recepción máximo, mantienen sus ventanas de recepción abiertas continuamente, cerrándola solo cuando transmiten. El consumo de energía de estos dispositivos es el máximo, y solo se recomienda que se utilice cuando la energía no esté limitada. Estos se emplean para aplicaciones que requieren baja latencia. La Figura 19 muestra el tipo de transmisión de los dispositivos de Clase C (LoRa Alliance, 2017)



Figura 19. Transmisión de un dispositivo Clase. Elaboración Propia

3.8. CONCLUSIONES

Se puede mencionar como principal conclusión que el protocolo LoRaWAN demuestra ser muy robusto ante las interferencias, principalmente a costa de sacrificar tasa de transferencia en pos de mayor radio de cobertura y tolerancia al ruido y, por esa razón, es de particular interés un análisis detallado de la capacidad de transferencia en un área determinada dado que los márgenes son bastante ajustados.

4. ANÁLISIS DE CAPACIDAD DE UNA RED LORAWAN

4.1. RESEÑA DEL CAPÍTULO

En este capítulo se realiza un análisis pormenorizado de la capacidad de un canal de comunicación LoRaWAN y de los factores físicos que influyen en la misma. Se discute la problemática relacionada con el tipo de acceso ALOHA del protocolo y cómo esa forma de acceso al medio limita de manera significativa la capacidad de los canales. Posteriormente, se presentan tablas de las capacidades de tráfico de los canales considerando las diferentes pérdidas de capacidad por señalización, para llegar a la realización de curvas de capacidad máxima y mostrando la necesidad de la utilización de múltiples canales para lograr incrementos de capacidad. A continuación, se comparan los planes de frecuencia US915 y AU915.

Finalmente se analizan los diferentes modelos de propagación de señales considerando el caso teórico de atenuación en el espacio libre versus modelos empíricos realizados con datos obtenidos en escenarios reales.

4.2. EFECTOS DE ALOHA EN UNA RED LORAWAN

Una red LoRaWAN opera en varios canales que están determinados por la configuración de la puerta de enlace (Gateway). El número de canales asignados depende de las restricciones del país y las configuraciones de red. Los canales pueden estar reservados para la transmisión de datos y se los llama canales principales, aparte se puede disponer que un canal reservado para las respuestas del Gateway a los nodos, así mismo como otros canales pueden ser utilizados por los nodos para transmitir solicitudes de unión al Gateway.

Debido a la naturaleza del protocolo ALOHA los nodos no realizan ningún chequeo del medio antes de transmitir y, por lo tanto, existe la probabilidad de colisión si otro nodo transmite al mismo tiempo. Esta probabilidad se incrementa significativamente a medida que la cantidad de nodos se incrementa en la red, además de producir un decrecimiento significativo del tráfico máximo posible (Haas & Deng, 2003).

Los problemas mencionados son especialmente importantes en las redes altamente cargadas, sin embargo, los desarrolladores y operadores de LoRaWAN han omitido este tipo de análisis. El crecimiento de la popularidad de IoT aumentará tanto la cantidad de dispositivos como la cantidad de redes de diferentes operadores que trabajan en la misma área en el espectro no licenciado, por lo que no se deben descuidar los potenciales problemas generados por este crecimiento (Bankov et al., 2016).

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Considerando la Ecuación 9, esto sitúa el alcance teórico máximo de LoRaWAN para SF=12 en 900 Km. El récord mundial actual con equipo estándar es de 702 km. (The Things Network, 2017).

En la Tabla 8, está claro que el impacto de usar una mayor dispersión es significativo en la tasa de bits total del canal.

$$R_b = \frac{4}{\frac{2^{SF}}{BW}} * SF \text{ [bits/seg]}$$

Ecuación 16. Tasa de Bits de LoRaWAN

		SF						
B/N	500000	6	7	8	9	10	11	12
RC	0	46875	27344	15625	8789	4883	2686	1465
	1	37500	21875	12500	7031	3906	2148	1172
	2	31250	18229	10417	5859	3255	1790	977
	3	26786	15625	8929	5022	2790	1535	837
	4	23438	13672	7813	4395	2441	1343	732
B/N	250000	6	7	8	9	10	11	12
RC	0	23438	13672	7813	4395	2441	1343	732
	1	18750	10938	6250	3516	1953	1074	586
	2	15625	9115	5208	2930	1628	895	488
	3	13393	7813	4464	2511	1395	767	419
	4	11719	6836	3906	2197	1221	671	366
B/N	125000	6	7	8	9	10	11	12
RC	0	11719	6836	3906	2197	1221	671	366
	1	9375	5469	3125	1758	977	537	293
	2	7813	4557	2604	1465	814	448	244
	3	6696	3906	2232	1256	698	384	209
	4	5859	3418	1953	1099	610	336	183

Tabla 8. Tasa de bits cruda del canal frente al ancho de banda del canal, factor de dispersión y tasa de codificación

Ahora bien, considerando la Figura 14, la carga útil se ve reducida por las sobrecargas de los campos de control mencionados. Ajustado para un canal de 125 kHz. Se obtienen las tasas de transferencia máximas listas en la Tabla 9, así como también una curva en la Figura 20 que muestra la tasa de bits por segundo en función del tamaño de paquete de la carga útil a ser entregada.

Análisis de capacidad de una red LoRaWAN

B/N	125000	6	7	8	9	10	11	12
RC	0	10381	5978	3373	1649	689	310	172
	1	8398	4845	2738	1359	584	261	146
	2	7051	4072	2304	1156	507	226	127
	3	6077	3513	1989	1006	448	199	112
	4	5339	3088	1750	890	401	178	100

Tabla 9. Tasa de bits máxima considerando encabezados MAC Y FRM

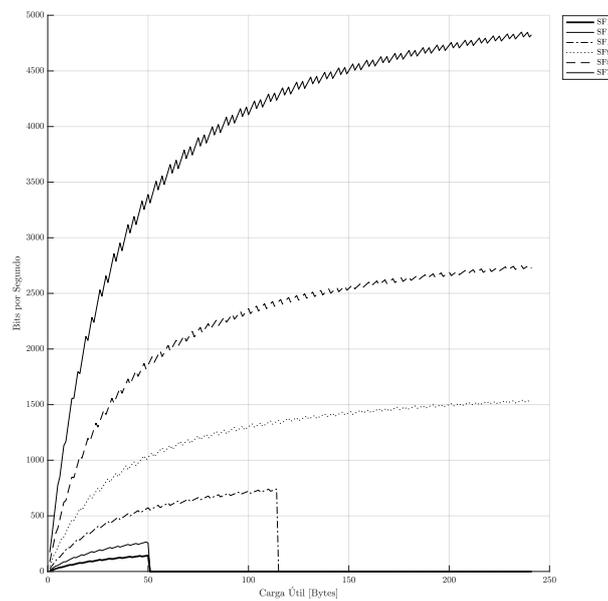


Figura 20. Curva de rendimiento de LoRaWAN considerando encabezados y corrección de errores. Elaboración propia.

El protocolo de *Media Access Control* (MAC) de LoRaWAN se basa en Pure ALOHA. Si definimos S como el promedio de paquetes generados por intervalo; la fuente de tráfico λ consta de un gran número de usuarios que forman una fuente de Poisson independiente con una tasa de paquetes agregada de X paquetes/s, el ancho de tiempo del paquete se fija, supuestamente, con un período de T segundos. Se puede considerar que cada usuario genera paquetes con poca frecuencia. S también se puede expresar como la tasa de rendimiento del canal. Un nodo retrasa la transmisión de un paquete previamente colisionado con un tiempo aleatorio. Entonces, el tráfico total no son solo paquetes nuevos sino repetición de retransmisión de paquetes colisionados.

$$S = \lambda T$$

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

$$G \geq S$$

$$G(n) = \lambda(n)T$$

$$S = G(n)P_{Suc} = \lambda(n)T * e^{-\lambda(n)2T}$$

Ecuación 15. Tiempo de paquete LoRaWAN

En ALOHA puro, una transmisión exitosa ocurre si el canal está libre durante el período de tiempo $2T$ (período de vulnerabilidad). La probabilidad de que no haya transmisiones en el periodo $2T$ es P_{suc} . El tráfico total del canal podría expresarse como se presenta en la Ecuación 15. Con estas restricciones, es posible mostrar que el rendimiento máximo del canal es del 18 % (Kleinrock & Lam, 1975) como se muestra en la Figura 21. (Polonelli et al., 2019).

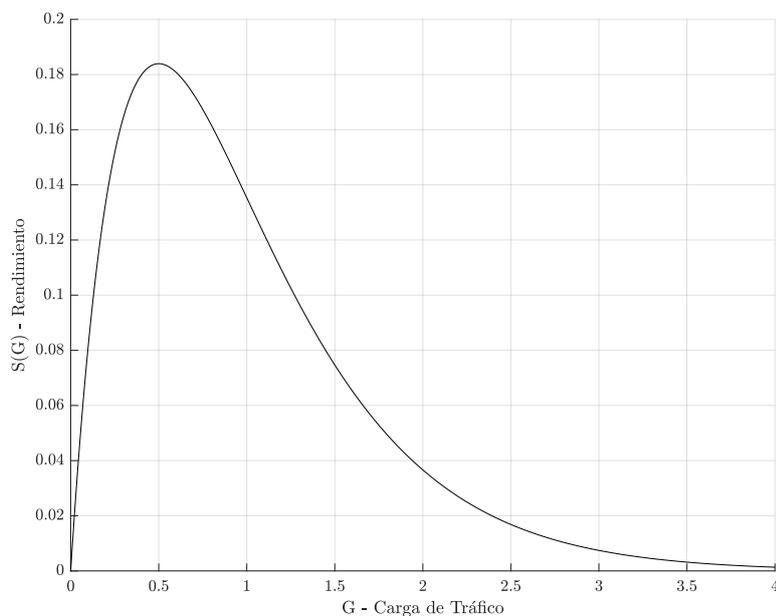


Figura 21. Curva de rendimiento del protocolo ALOHA vs carga del canal. (Polonelli et al., 2019).

Dicho esto, el ancho de banda total máximo ajustado por canal considerando ALOHA MAC para canales de 125 kHz se muestra en la Tabla 10.

Análisis de capacidad de una red LoRaWAN

0.18	125000	6	7	8	9	10	11	12
RC	0	1869	1076	607	297	124	56	31
	1	1512	872	493	245	105	47	26
	2	1269	733	415	208	91	41	23
	3	1094	632	358	181	81	36	20
	4	961	556	315	160	72	32	18

Tabla 10. Tasa de bits del canal frente al factor de dispersión y la tasa de codificación para ALOHA

Y considerando un *gateway* de 8 canales (125 KHz.), el ancho de banda total está de acuerdo con la Tabla 11.

8	125000	6	7	8	9	10	11	12
RC	0	14949	8609	4857	2374	993	447	248
	1	12093	6976	3943	1957	841	376	210
	2	10154	5864	3318	1665	730	325	182
	3	8751	5058	2864	1449	645	286	161
	4	7688	4447	2520	1282	577	256	144

Tabla 11. Tasa de bits total de la puerta de enlace frente al factor de dispersión y la tasa de codificación para Aloha en una puerta de enlace de 8 canales.

4.3. PLANES DE FRECUENCIA

4.3.1. PLAN DE FRECUENCIA EE. UU.

El espectro en los EE. UU. para LoRaWAN tiene 64 canales de enlace ascendente disponibles (125 kHz cada uno) (canales 0-63) a partir de 902,3 MHz que se incrementan cada 200 kHz hasta 914,9 MHz

Hay ocho canales de enlace ascendente superpuestos (500 kHz cada uno) (canales 64-71) desde 903 MHz que se incrementan cada 1,6 MHz hasta 914,2 MHz

Para la comunicación de puerta de enlace a nodo, hay 8 canales de enlace descendente (500 kHz cada uno) (canales 0-7) desde 923,3 MHz que aumentan cada 600 kHz hasta 927,5 MHz

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

<i>Uplink</i>		Canal							
125 kHz		0	1	2	3	4	5	6	7
Subbanda	0	902,3	902,5	902,7	902,9	903,1	903,3	903,5	903,7
	1	903,9	904,1	904,3	904,5	904,7	904,9	905,1	905,3
	2	905,5	905,7	905,9	906,1	906,3	906,5	906,7	906,9
	3	907,1	907,3	907,5	907,7	907,9	908,1	908,3	908,5
	4	908,7	908,9	909,1	909,3	909,5	909,7	909,9	910,1
	5	910,3	910,5	910,7	910,9	911,1	911,3	911,5	911,7
	6	911,9	912,1	912,3	912,5	912,7	912,9	913,1	913,3
	7	913,5	913,7	913,9	914,1	914,3	914,5	914,7	914,9
500 kHz		903,0	904,6	906,2	907,8	909,4	911,0	912,6	914,2
<i>Downlink</i>		923,3	923,9	924,5	925,1	925,7	926,3	926,9	927,5

Tabla 12. Bandas de espectro para LoRaWAN en EE. UU.

4.3.2. PLAN DE FRECUENCIA ARGENTINA Y AUSTRALIA

El espectro en Argentina para LoRaWAN tiene 64 canales de enlace ascendente disponibles (125 kHz cada uno) (canales 0-63) a partir de 915,2 MHz que se incrementan cada 200 kHz hasta 927,8 MHz.

Hay 8 canales de enlace ascendente superpuestos (500 kHz cada uno) (canales 64-71) desde 915,9 MHz que se incrementan cada 1,6 MHz hasta 927,1 MHz

Para la comunicación de puerta de enlace a nodo, hay 8 canales de enlace descendente (500 kHz cada uno) (canales 0-7) desde 923,3 MHz que aumentan cada 600 kHz hasta 927,5 MHz

Esto de ver representado en la Tabla 13 y el la Figura 22.

<i>Uplink</i>		Canal							
125 kHz		0	1	2	3	4	5	6	7
Subbanda	0	915,2	915,4	915,6	915,8	916,0	916,2	916,4	916,6
	1	916,8	917,0	917,2	917,4	917,6	917,8	918,0	918,2
	2	918,4	918,6	918,8	919,0	919,2	919,4	919,6	919,8
	3	920,0	920,2	920,4	920,6	920,8	921,0	921,2	921,4
	4	921,6	921,8	922,0	922,2	922,4	922,6	922,8	923,0
	5	923,2	923,4	923,6	923,8	924,0	924,2	924,4	924,6
	6	924,8	925,0	925,2	925,4	925,6	925,8	926,0	926,2
	7	926,4	926,6	926,8	927,0	927,2	927,4	927,6	927,8
500 kHz		915,9	917,5	919,1	920,7	922,3	923,9	925,5	927,1
<i>Downlink</i>		923,3	923,9	924,5	925,1	925,7	926,3	926,9	927,5

Análisis de capacidad de una red LoRaWAN

Tabla 13. Asignación de espectro para LoRaWAN en Argentina

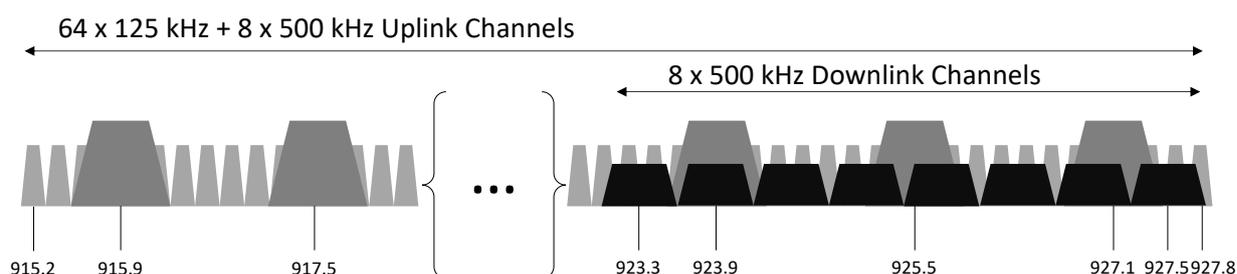


Figura 22. Mapa de canales del plan de frecuencias LoRa en Argentina. (Semtech, n.d.)

<i>DataRate</i>	Configuración	Tasa de bits física	<i>Downlink</i> DR	Carga Útil Máxima (400ms) [bytes]	Tiempo de Aire [ms]
<i>Uplink</i>					
0	LoRa: SF12 / 125 kHz	250	8	N/A	1248,8
1	LoRa: SF11 / 125 kHz	440	9	N/A	823,3
2	LoRa: SF10 / 125 kHz	980	10	11	370,7
3	LoRa: SF9 / 125 kHz	1760	11	53	205,8
4	LoRa: SF8 / 125 kHz	3125	12	125	113,2
5	LoRa: SF7 / 125 kHz	5470	13	222	61,7
6	LoRa: SF8 / 500 kHz	12500	13	222	28,3
7	Reservado				
<i>Downlink</i>					
8	LoRa: SF12 / 500 kHz	980		33	288,8
9	LoRa: SF11 / 500 kHz	1760		109	144,4
10	LoRa: SF10 / 500 kHz	3900		222	72,2
11	LoRa: SF9 / 500 kHz	7000		222	41,4
12	LoRa: SF8 / 500 kHz	12500		222	20,6
13	LoRa: SF7 / 500 kHz	2190		222	11,6
14	Reservado				
15	Reservado				

Tabla 14. Modos de operación LoRa para el plan AU915

4.4. MODELOS DE PROPAGACIÓN DE SEÑALES

El modelo teórico más simple de propagación de señales entre dos puntos es definido en (*IEEE Std 145-2013 (Revision of IEEE Std 145-1993)*, 2014) como Atenuación en el espacio libre (FSPL) y se define como “La pérdida entre dos radiadores isotrópicos en el espacio libre, expresada como

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

una relación de potencia, La pérdida de espacio libre no se debe a la disipación, sino más bien al hecho de que la densidad del flujo de potencia disminuye con el cuadrado de la distancia de separación y el área efectiva de la antena isotrópica receptora aumenta con el cuadrado de la longitud de onda". El espacio de pérdida libre generalmente se expresa en decibelios y está dado por la fórmula $20 * \log_{10}(\frac{4\pi r}{\lambda})$ donde r es la separación de las dos antenas y λ es la longitud de onda.

Otra forma de presentar la FSPL en función de la frecuencia de la señal y en unidades de decibelios se muestra en la Ecuación 17.

La atenuación es un factor que se considera en el presupuesto de enlace de un sistema de comunicación, para garantizar que llegue suficiente potencia de radio al receptor de modo que la señal transmitida se reciba de manera correcta.

$$FSPL=20 * \log_{10}(\frac{4\pi r f}{c})$$

Ecuación 17. Fórmula de la atenuación en el espacio libre

Una de las limitaciones del modelo de FSPL es que considera una línea directa sin obstrucciones ni reflexiones de señal entre el transmisor y receptor, a efectos de mitigar estas limitaciones se han desarrollado modelos empíricos más complejos, que después de una gran cantidad de mediciones en el campo, consideran valores de propagación vistos en situaciones reales y algunos de ellos serán revisados más adelante.

Existen algunos modelos de propagación para señales en el rango de frecuencia UHF, y el más conocido es el modelo Okumura-Hata. El modelo de Hata proporciona el valor de la pérdida de trayecto a una distancia dada entre una estación base y un usuario móvil. Considera varios factores como frecuencia, alturas de antena y otros (Hata, 1980).

En áreas urbanas:

$$L_{PU} = 69.55 + 26.16 \log(f) - 13.82 \log(h_B) - a(h_M) + [44.9 - 6.55 \log(h_B)] \log(d)$$

Para una ciudad pequeña mediana

$$a(h_M) = (1.1 \log(f) - 0.7)h_M - (1.56 \log(f) - 0.8)$$

Análisis de capacidad de una red LoRaWAN

L_P =Atenuación en áreas urbanas. Unidad: decibelio [dB]

h_B = Altura de la estación base sobre el suelo [metros]

h_M = Altura de la estación móvil sobre el suelo [metros]

f = Frecuencia de transmisión [MHz]

$a(h_M)$ = Factor de corrección de la altura de la Antena = Distancia desde la puerta de enlace al dispositivo [km]

En áreas suburbanas

$$L_{PSU} = L_{PU} - 2 \left(\log \left(\frac{f}{28} \right) \right)^2 - 5.4$$

En áreas rurales o abiertas

$$L_{PSU} = L_{PU} - 4.78 (\log(f))^2 + 18.33 \log(f) - 40.94$$

En la Figura 23 se comparan los resultados de los modelos de FSPL con el de Hata para espacio abierto para diferentes distancias.

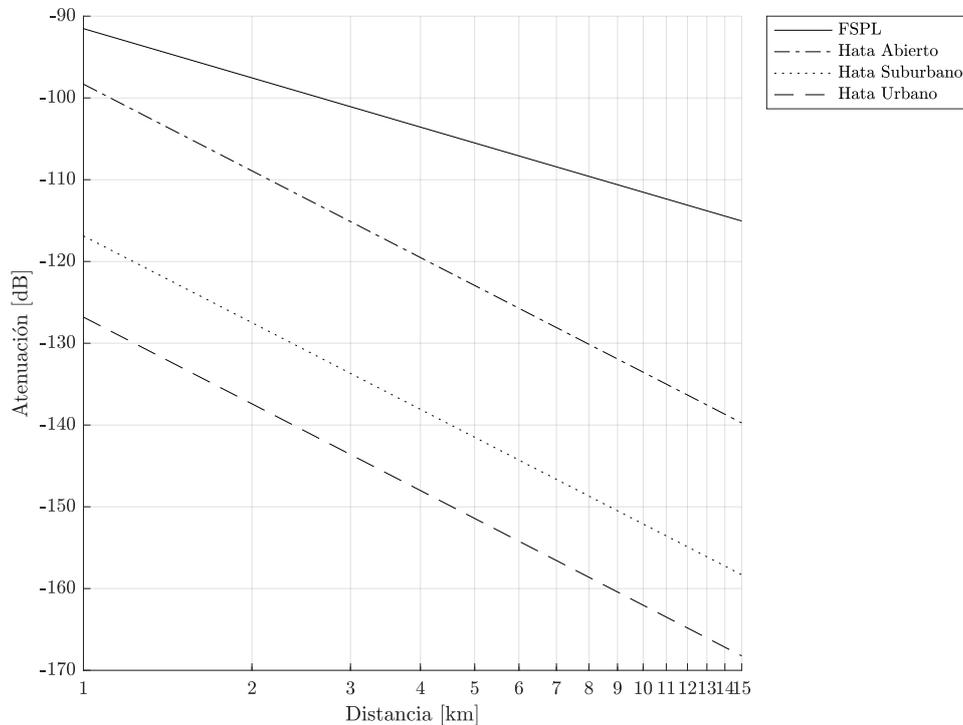


Figura 23. Comparación de FSPL vs Hata. Elaboración Propia

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Este modelo de Hata funciona muy bien para la mayoría de las aplicaciones de IoT; sin embargo, su precisión está limitada por una altura mínima de la estación base de 30 m. En esta tesis algunos análisis fueron realizados sobre estaciones base que estarán a alturas potencialmente menores entre 8 y 12 m de altura, por lo que este modelo no es directamente aplicable.

Un modelo más preciso para antenas de baja altura se describe en (Villard et al., 2012) y se aplica mejor a algunos de los escenarios estudiados en esta tesis. Este modelo considera modelos urbanos y suburbanos y también aplicaciones exteriores e interiores en comerciales (muros de hormigón) y residenciales (estructura de madera y paredes de ladrillo).

Este modelo se ha realizado en una banda de 900 MHz, por lo que se aplica perfectamente a este estudio.

$$L_P = 20 \log(h_B) + 20 \log(h_M) - 43.36 \log(d) - A - B - C$$

L_P = Atenuación en áreas urbanas. Unidad: decibelio [dB]

h_B = Altura de la estación base sobre el suelo [metros]

h_M = Altura de la estación móvil sobre el suelo [metros]

d = Distancia de la puerta de enlace al dispositivo [m]

A = Constante – Tipo de área - 24,3 dB en suburbana y 29,3 dB en urbana

B = Constante – Atenuación del edificio – 0 dB en exteriores, 17,7 dB para edificios comerciales (concreto) y 5,4 dB para casas suburbanas (estructura de madera y ladrillo)

C = Constante - Efecto de sombreado - 0 dB en exteriores, 9,3 dB para edificios comerciales y 6,4 dB para viviendas

4.5. MODELADO DE ESCENARIOS

Se modelaron dos tipos de escenarios, uno enfocado en un entorno urbano denso con edificios de hormigón y sensores instalados internamente y otro suburbano disperso también con sensores interiores.

4.5.1. URBANO DENSO CON EDIFICIOS DE HORMIGÓN Y SENSORES INTERIORES

Análisis de capacidad de una red LoRaWAN

Para una red LoRaWAN típica con una puerta de enlace que se ejecuta en un nodo de fibra en el cable portante, supongamos que la altura de la estación base sobre el suelo $h_B = 9m$ y que la altura de la estación móvil sobre el suelo $h_M = 1.5m = 4 ft$

Condiciones de contorno del modelo

$$h_B = 9m$$

$$h_M = 1.5m$$

$$A = 29.3dB$$

$$B = 17.7dB$$

$$C = 9.3dB$$

$$L_P = 20 \log(9) + 20 \log(1.5) - 43.36 \log(d) - 29.3 - 17.7 - 9.3$$

$$L_P = 19.08 + 3.52 - 43.36 \log(d) - 29.3 - 17.7 - 9.3$$

$$L_P = -43.36 \log(d) - 33.7$$

Graficando la atenuación en función de la distancia como en la Figura 24, se marcan los umbrales para la operación en los factores de dispersión más resistente por un lado y rápido por el otro, SF=12 y SF=7.

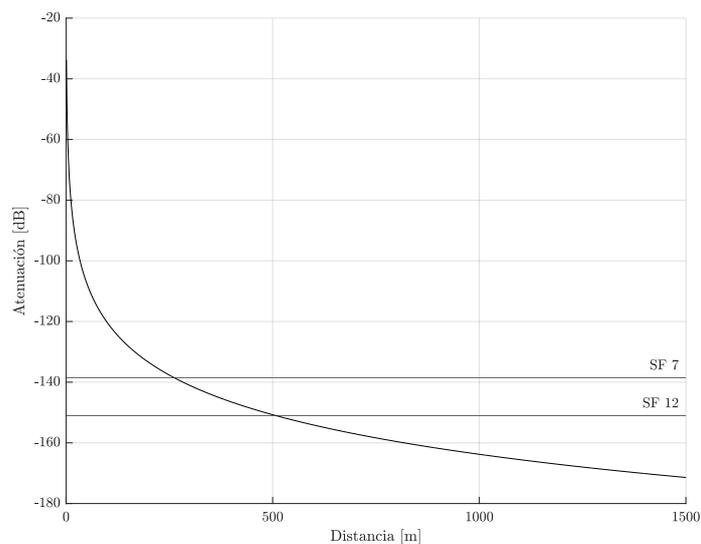


Figura 24. Atenuación vs distancia para escenario urbano denso. Elaboración Propia

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

La distancia máxima para diferentes factores de dispersión se puede calcular en función de su presupuesto de enlace, como se muestra en la tabla 4.

$$d = 10^{\left(\frac{L_B+33.7}{-43.36}\right)}$$

Para SF=12, $L_B = -151.08$ dB

$$d = 10^{\left(\frac{-151.08+33.7}{-43.36}\right)}$$

$$d = 509.5m$$

Para SF=7, $L_B = -138.53$ dB

$$d = 10^{\left(\frac{-138.53+33.7}{-43.36}\right)}$$

$$d = 261.6m$$

La conclusión es que, en un escenario urbano denso con edificios de concreto, una red LoRaWAN puede operar con el modo de transmisión menos eficiente hasta 509,5 m y el límite para el perfil más eficiente es 261,1 m

4.5.2. SUBURBANO DISPERSO CON CASAS Y SENSORES INTERIORES

Para una red LoRaWAN típica con una puerta de enlace que se ejecuta en un nodo de fibra en el cable, supongamos $h_B = 9m = 30 ft$ y $h_M = 1.5m = 4 ft$

Condiciones de contorno del modelo

$$h_B = 9m$$

$$h_M = 1.5m$$

$$A = 24.3dB$$

$$B = 5.4dB$$

$$C = 6.4dB$$

Análisis de capacidad de una red LoRaWAN

$$L_P = 19.08 + 3.52 - 43.36 \log(d) - 24.3 - 5.4 - 6.4$$

$$L_P = -43.36 \log(d) - 13.5$$

Graficando la atenuación en función de la distancia como en la Figura 25, se marcan los umbrales para la operación en los factores de dispersión más resistentes por un lado y más rápidos por el otro, SF=12 y SF=7.

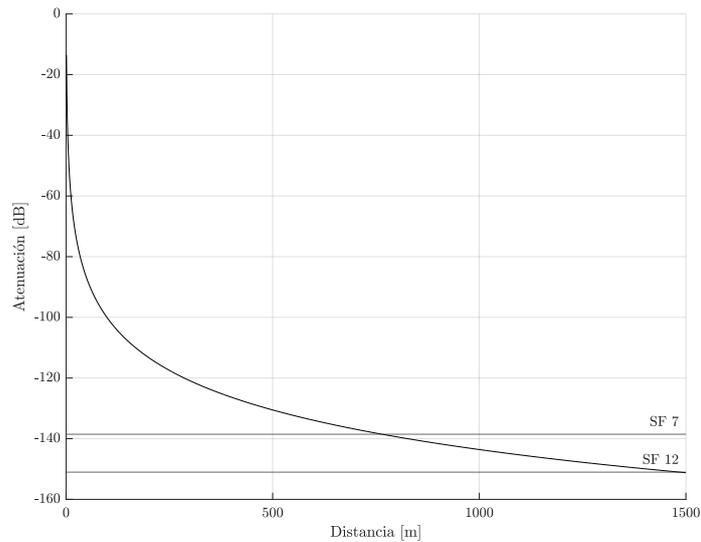


Figura 25. Atenuación vs distancia para escenario suburbano. Elaboración Propia

$$d = 10^{\left(\frac{L_B + 13.5}{-43.36}\right)}$$

Para SF=12, $L_B = -151.08$ dB

$$d = 10^{\left(\frac{-151.08 + 13.5}{-43.36}\right)}$$

$$d = 1489.25m$$

Para SF=7, $L_B = -138.53$ dB

$$d = 10^{\left(\frac{-138.53 + 13.5}{-43.36}\right)}$$

$$d = 764.7m$$

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

La conclusión es que, en un escenario suburbano disperso con casas y sensores interiores, una red LoRaWAN puede operar con el modo de transmisión menos eficiente hasta 1489,25 m y el límite para el perfil más eficiente es 796,7 m.

4.6. CONCLUSIONES

De este capítulo se pueden extraer varias conclusiones, primero que la naturaleza ALOHA del protocolo LoRaWAN impacta severamente en la capacidad en bits por segundo de cada canal, por otro lado, el comportamiento del protocolo está fuertemente influenciado por el medio donde esta aplicado, es decir se registra una variación significativa de escenarios rurales a urbanos en distancia de cobertura y capacidad. Esto hace necesario la utilización de gateways de canales múltiples para poder expandir la capacidad. Por último, se ve claramente como la restricción de espectro actual en la República Argentina permite solamente el uso de 8x4 canales bidireccionales, dado que la banda alta de la asignación se encuentra compartida con los canales de bajada, en comparación con la disponibilidad de 64 canales en el plan de frecuencias US915.

5. MATERIALES DE LABORATORIO Y TRABAJO DE CAMPO

5.1. RESEÑA DEL CAPÍTULO

En este capítulo se presenta un detalle de todos los materiales utilizados tanto en el trabajo de laboratorio como en el de campo tanto en el campo del software como en el hardware. Se abordan los beneficios de la utilización del lenguaje Python en microcontroladores y su aplicación en la plataforma de transmisores *gateways* llamada Pycom. Posteriormente se discute el uso de herramientas de radio definidas por software con el objeto de realizar mediciones flexibles y de bajo costo del protocolo LoRaWAN en los ambientes de laboratorio y de campo. Por último, se investiga en detalle una suite de servidores de control red y de extracción de datos y se presentó en detalle los beneficios y flexibilidad de esta aplicación tanto para laboratorio como para potenciales redes reales, así como también su integración con sistemas de bases de datos y de analítica.

5.2. MICROPYTHON

5.2.1. UTILIZACIÓN DE PYTHON

Python es un lenguaje de scripting de alto nivel, interpretado y orientado a objetos. Una de las ventajas más grandes de Python es tener una sintaxis clara y fácil de entender que se lee lo más cercana posible al idioma inglés como sea posible.

Estas son algunas de las características clave de Python.

- Un intérprete procesa Python en tiempo de ejecución. No se requiere el uso de ningún compilador.
- Python soporta construcciones de programación orientada a objetos a través de clases y métodos.
- Python es un gran lenguaje para los programadores de nivel principiante y apoya el desarrollo de una amplia gama de aplicaciones.
- Python es un lenguaje de scripting, pero se puede utilizar para una amplia gama de aplicaciones.
- Es muy popular y utilizado en todo el mundo dándole un enorme base de sustentación y una gran cantidad de bibliotecas disponibles

5.2.2. UTILIZACIÓN DE MICROPYTHON

Hasta hace algunos años solo se podía correr aplicaciones en Python en computadoras personales o servidores, puesto que se requería de un sistema operativo para poder ejecutar, el entorno de ejecución y las bibliotecas requeridas.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

En la actualidad se lo utiliza para programar dispositivos con hardware mucho más limitado, como algunas computadoras compactas (Raspberry Pi, Beaglebone, etc.) y sistemas embebidos (ESP32, ESP8266, etc.).

Por último, existe MicroPython, este fue creado y es mantenido por Damien P. George, Paul Sokolovsky y otros colaboradores. Fue diseñado para ser una versión eficiente del lenguaje Python 3 para que pueda ser instalado en un pequeño microcontrolador. Dado que Python es un lenguaje interpretado y, por lo tanto, más lento (en general) que un lenguaje compilado, MicroPython fue diseñado para ser lo más eficiente posible para que pueda ejecutarse en microcontroladores que normalmente son más lentos y tienen mucha menos memoria que una computadora personal (Charles Bell, 2017).

Otro aspecto es que las placas de microcontroladores como Arduino requieren un paso de compilación que debe realizar en la computadora para después cargar el ejecutable binario en la placa. Por el contrario, dado que MicroPython tiene su intérprete ejecutándose directamente en el hardware, no es necesario el paso intermedio para preparar el código.

Esto permite a los fabricantes de hardware construir placas pequeñas y económicas que incluyen MicroPython en el mismo chip que el microprocesador, esto brinda la capacidad de conectarse a la placa, escribir código y ejecutarlo sin ningún paso adicional.

MicroPython es una implementación completa de las características principales de Python 3, incluido un tiempo de ejecución compacto e intérprete interactivo.

5.2.3. VENTAJAS DE USAR PYTHON EN IOT

Intérprete interactivo: Las placas MicroPython tienen incorporada una consola interactiva especial a la que se puede acceder conectándose a la placa con un cable USB o vía WiFi. Esta consola (REPL Console) permite escribir el código y ejecutarlo una línea a la vez. Es una excelente manera de crear prototipos de código o ejecutar un proyecto a medida que se lo desarrolla.

Bibliotecas estándar de Python: MicroPython también es compatible con la mayoría de las bibliotecas estándar de Python. En general, puede esperar encontrar que MicroPython admite más del 80% de las bibliotecas más utilizadas. Estos incluyen el análisis y generación de JSON, programación de sockets, manipulación de cadenas, entrada/salida de archivos y soporte de expresiones regulares.

Bibliotecas a nivel de hardware: MicroPython tiene bibliotecas integradas que le permiten acceder al hardware directamente para encender o apagar pines analógicos, leer datos analógicos, leer datos digitales e incluso controlar hardware con modulación de ancho de pulso (PWM).

5.3. EQUIPAMIENTO LORAWAN

5.3.1. FIPY

Es una placa de desarrollo para proyectos de IoT muy utilizada dado que es compatible con la mayoría de los sistemas de comunicación IoT: LoRa, Sigfox, WiFi, Bluetooth y LTE-CAT M1/NB1 móvil, de ahí se deriva su nombre (Five Python). Se programa en MicroPython, lo cual como se mencionó anteriormente provee muchos beneficios en el tiempo de programación y de despliegue de prototipos.

La característica más notable de la placa es su pequeño tamaño como se puede ver en la Figura 26. La placa mide aproximadamente 55 mm. x 20 mm. x 12 mm. Sus características técnicas se muestran la Tabla 15. Se puede usar la placa en una placa experimental, así como también puede comprar una placa especial que funciona como adaptador. Pycom lo llama placa de expansión y proporciona una unidad microSD, un conector micro USB, LED, replica todos los pines e incluye un conector de batería. La placa de expansión hace que el uso del FiPy sea considerablemente más fácil.

La Figura 27 muestra la placa de expansión Pycom. Sus principales características incluyen: La unidad microSD, el conector micro USB y el conector de batería LIPO también es un cargador, por lo que mientras su placa está conectada a una fuente de alimentación USB, se carga la batería LIPO.

La placa FiPy también es compatible con placas de sensado remoto como las Pysense y Pytrack que serán descritas más adelante. En el ámbito de esta investigación los FiPy se utilizaron como dispositivos remotos recolectores de datos junto con placas de sensado Pysense y Pytrack, las placas de expansión se utilizaron para la programación inicial de los FiPy.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

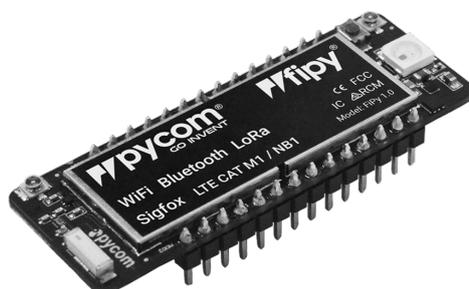


Figura 26. Imagen de Pycom FiPy

<ul style="list-style-type: none"> - Five Networks: WiFi, BLE, cellular LTE-CAT M1/NB1, LoRa and Sigfox - Powerful CPU. - Can also double up as a Nano LoRa gateway - MicroPython enabled - Fits in a standard breadboard (with headers) - Ultra-low power usage: a fraction compared to other connected micro controllers - World ready, one product covers all LTE-M bands
<p style="text-align: center;">CPU</p> <ul style="list-style-type: none"> - Xtensa® dual-core 32-bit LX6 microprocessor(s), up to 600 DMIPS - Hardware floating point acceleration - Python multi-threading - An extra ULP-coprocessor that can monitor GPIOs, the ADC channels and control most of the internal peripherals during deep-sleep mode while only consuming 25uA.
<p style="text-align: center;">Memory</p> <ul style="list-style-type: none"> - RAM: 520KB + 4MB - External flash: 8MB
<p style="text-align: center;">WiFi</p> <ul style="list-style-type: none"> - 802.11b/g/n 16mbps
<p style="text-align: center;">Bluetooth</p> <ul style="list-style-type: none"> - Low energy and classic
<p style="text-align: center;">LoRa</p> <ul style="list-style-type: none"> - LoRaWAN stack - Class A and C devices - Node range: Up to 40km - Nano-gateway: Up to 22km (Capacity up to 100 nodes)

Tabla 15. Especificaciones técnicas de FiPy

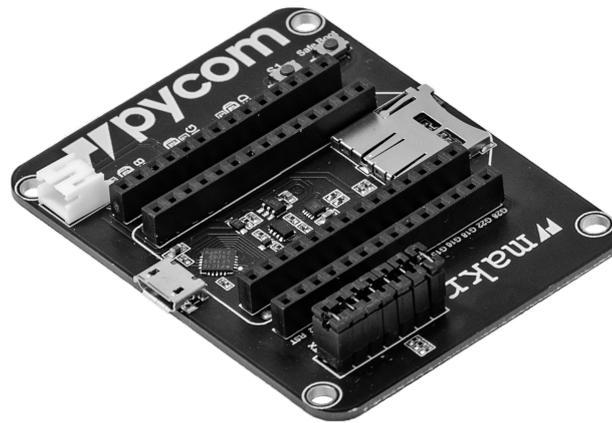


Figura 27. Imagen de Pycom *Expansion Board*

5.3.2. PYSENSE

PySense es un adaptador de sensor que se puede utilizar con cualquier módulo Pycom y contiene una serie de sensores, como luz ambiental, presión y humedad. En la Figura 28 se visualiza mientras que en la Tabla 16 se listan sus características técnicas.

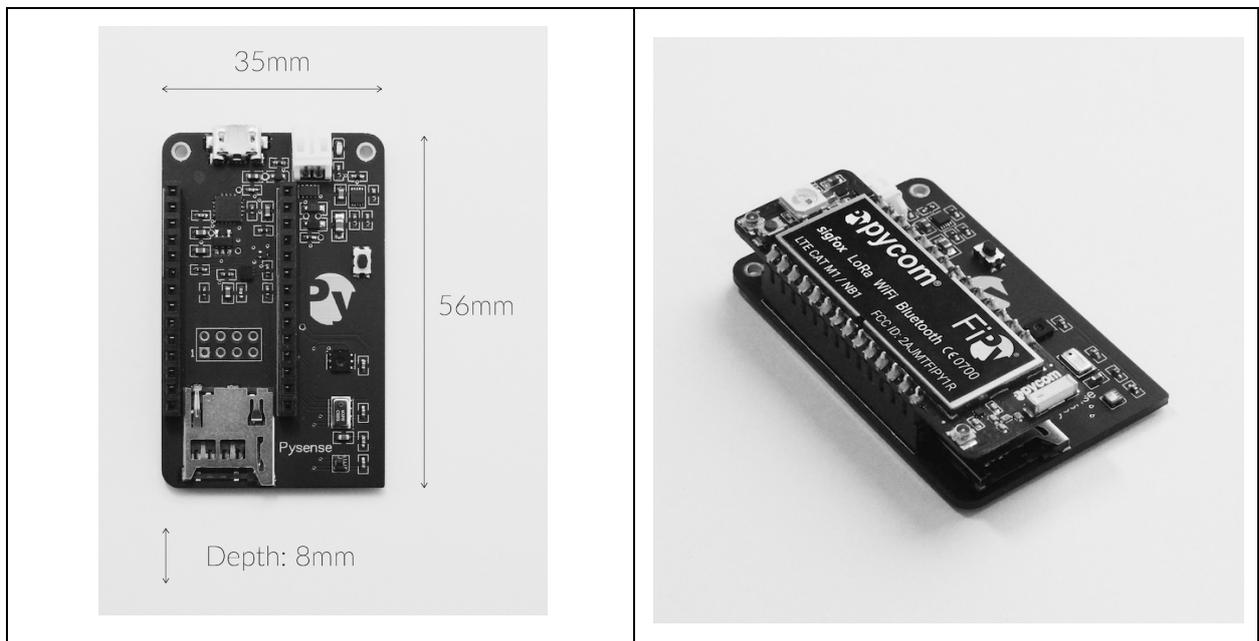


Figura 28. Imagen de Pycom PySense

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Sensors
- Ambient light sensor
- Barometric pressure sensor
- Humidity
- 3 axis 12-bit accelerometer
- Temperature sensor
- A USB port with serial access
- LiPo battery charger
- MicroSD card compatibility
- Ultra low power operation (~1uA in deep sleep)
Mechanical
Size: 55mm x 35mm x 10mm
Operating Temperature: -40 to +85 Celsius
Weight: 50 g

Tabla 16. Especificaciones técnicas de PySense

5.3.3. PYTRACK

PyTrack es un adaptador de sensor que se puede utilizar con cualquier módulo Pycom. y contiene una serie de sensores, como acelerómetro y GPS. En la Figura 29 se visualiza mientras que en la Tabla 17 se listan sus características técnicas.

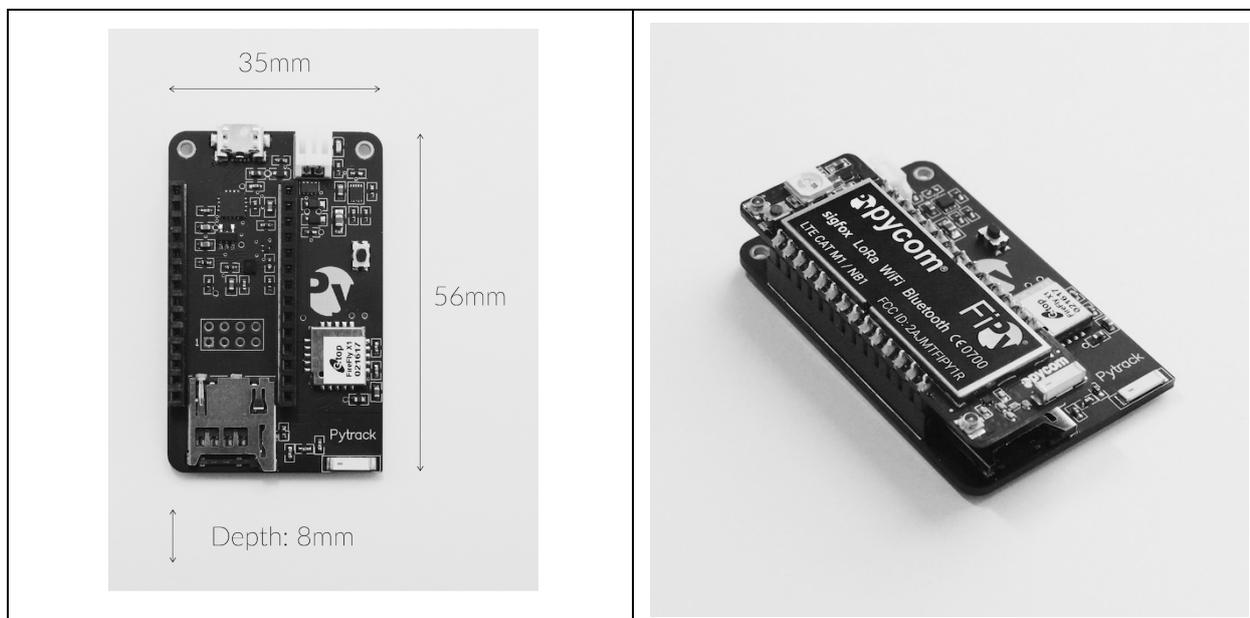


Figura 29. Imagen de Pycom PyTrack

Materiales de laboratorio y trabajo de campo

Sensors
- A super accurate GNSS + Glonass GPS
- 3 axis 12-bit accelerometer
- A USB port with serial access
- LiPo battery charger
- MicroSD card compatibility
- Ultra low power operation (~1uA in deep sleep)
Location services supported GPS, GLONASS, Galileo and QZSS.
Mechanical
Size: 55mm x 35mm x 10mm
Operating Temperature: -40 to +85 Celsius
Weight: 11g

Tabla 17. Especificaciones técnicas de PyTrack

5.3.4. PYGATE

Es un gateway LoRaWAN de ocho canales de bajo costo que viene en forma de adaptador. Tiene el mismo factor de forma que los adaptadores compatibles con Pycom y puede conectarse a placas WiPy3, LoPy4 o un GPy.

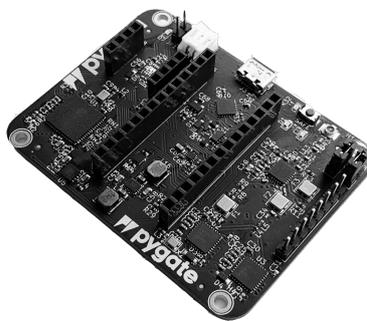


Figura 30. Imagen de Pycom Pygate

Dual SX1257 transceivers for a total of 8 channels support
Frequencies: 863-870 MHz or 902-928 MHz
Baseband processor: SX1308
Compatible with WiPy3.0, GPy and LoP4
Ultra-low power standby mode
Ethernet with PoE via the optional daughter board
Powered via USB, LiPo Battery and Ethernet via the optional daughter board

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

LiPo battery charging via the USB port
USB to serial connection for the Pycom module with auto-programming features
Dimensions: 65 (W) x 50 (L) x 8 (H) mm
(Same small form factor as the Expansion Board v3.0/v3.1)
Fits the universal IP67 Case for outdoor use

Tabla 18. Especificaciones técnicas de Pygate

<p>Procesamiento: Espressif ESP32 SoC Sistema de radio WiFi con doble procesador en chip</p> <p>El procesador de red maneja la conectividad WiFi y la pila IPv6 El procesador principal es totalmente gratuito para ejecutar la aplicación del usuario.</p> <p>Un coprocesador ULP adicional que puede monitorear GPIO, los canales ADC y controlar la mayoría de los periféricos internos durante el modo de reposo profundo mientras solo consume 25uA.</p>
<p>Memoria: RAM de 4MB con memoria flash 8MB</p>
<p>Conectividad: WiFi 802.1b/g/n 16mbps Bluetooth clásico y de bajo consumo Frecuencias Operativas Sigfox RCZ4 – 920-922MHz (ANZ, Latinoamérica y S-E Asia)</p>
<p>Especificación LoRa Transceptor Semtech LoRa SX1276</p> <p>LoRaWAN</p> <p>Dispositivos de clase A y C</p> <p>Frecuencias de operación LoRa:</p> <p>868 MHz (Europa) a 14dBm máximo</p> <p>915 MHz (América del Norte y del Sur, Australia y Nueva Zelanda) a 20dBm máximo</p> <p>433 MHz (Europa) a 10dBm máximo</p> <p>470 – 510 MHz (China) a 14dBm máximo</p> <p>Especificación de rango LoRa Rango de nodos: hasta 40 km</p> <p>Nano-gateway: hasta 22 km</p> <p>Capacidad de Nano-gateway: hasta 100 nodos</p>
<p>Interface – 2 x UART, SPI, 2 x I2C, I2S, tarjeta microSD</p> <p>Canales analógicos: ADC de 8_12 bits</p> <p>Temporizadores: 4_16 bit con PWM y captura de entrada</p> <p>DMA en todos los periféricos</p> <p>GPIO: hasta 24</p>
<p>Alimentación: Voltaje de entrada: 3.3V o 5.5V</p>
<p>Salida: de 3.3V con capacidad hasta de 400mA</p>
<p>WiFi: 12mA en modo activo, 5uA en espera</p>
<p>LoRa: 15 mA en modo activo, 1-uA en espera Sigfox (Australia, Nueva Zelanda y Sudamérica): 12 mA en modo Rx, 120 mA en modo Tx y 0,5 uA en modo de espera</p>

Seguridad y Certificación: SSL/TLS soporta hasta 1.2 WPA seguridad empresarial FCC 2AJMTLOPY4R CE 0700
Hash/Encriptación: SHA, MD5, DES, AES
Notas: RTC hasta 32kHz MicroPython habilitado

Tabla 19. Especificaciones técnicas de LoPy4

5.4. EQUIPAMIENTO DE MEDICIÓN DE RADIOFRECUENCIA

5.4.1. SOFTWARE DEFINED RADIO (SDR)

Históricamente, los sistemas de comunicación han sido conformados por una combinación de procesamiento digital y parte de recepción de radiofrecuencia analógica. En la actualidad, en los sistemas modernos, el procesamiento de señales ha progresado hasta tal punto que la mayoría de las funcionalidades de banda base están siendo implementadas en software. La flexibilidad del hardware de RF para ser reutilizado y reconfigurado ha llevado a tener un *front-end* de radio que maneja la mayoría de los sistemas de RF. Normalmente, el *front-end* de RF era controlado por software, pero no definido totalmente por software. Esta combinación moderna de *front-ends* de RF flexibles y procesamiento de señales en software ha llevado al nacimiento de la radio definida por software (Travis F. Collins et al., 2018).

La SDR es un sistema de comunicación muy flexible que es capaz de configurar el hardware para realizar varias funciones diferentes. Los componentes de un SDR se implementan o definen en un medio programable y proporcionan un enlace flexible entre los dispositivos de hardware y el software que lo controla (X. Cai, M. Zhou, and X. Huang, 2017).

Un sistema SDR es un dispositivo que realiza varias tareas complejas simultáneamente para permitir la recepción y opcionalmente transmisión de datos. En general, un sistema de comunicaciones digitales consiste en una secuencia interdependiente de operaciones responsables de tomar algún tipo de información, ya sea la voz, música, imágenes de video, o datos y transmitirlos por aire a un receptor para procesarla y decodificarla en una versión reconstruida de la señal original.

Si la información original es analógica (como el audio), primero debe digitalizarse utilizando técnicas como la cuantización para que se pueda obtener una representación binaria de esta información. Una vez en formato binario, el transmisor procesa digitalmente esta información y

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

la convierte en una forma de onda sinusoidal electromagnética que se define por sus características físicas, como la amplitud, frecuencia de la portadora y fase. En el otro extremo del enlace de comunicaciones, el receptor tiene la tarea de identificar correctamente las características físicas de la forma de onda modulada que fue transmitida a través de un canal potencialmente ruidoso y con distorsión y, finalmente, convertir la señal recibida a la representación binaria correcta. Los componentes básicos de un sistema de comunicación digital se muestran en la Figura 31.

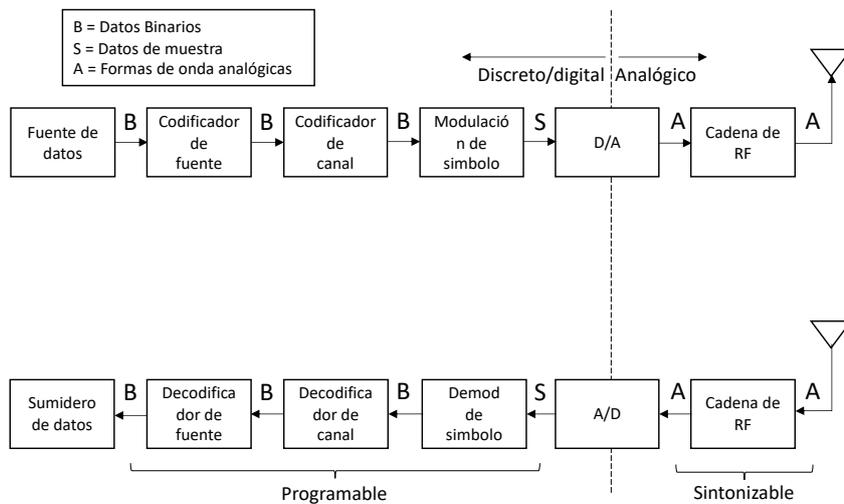


Figura 31. Diagrama de bloques de un transceptor SDR

Dada la complejidad de una plataforma SDR y sus componentes, es importante comprender las limitaciones de una plataforma SDR específica y cómo su diseño puede afectar el rendimiento del prototipo resultante. Por ejemplo, es muy deseable tener un procesamiento de banda base en tiempo real para la detección de espectro y operaciones de transmisión y recepción rápidas con alto rendimiento computacional y baja latencia.

Sin embargo, si el microprocesador empleado por la plataforma SDR no es lo suficientemente potente como para soportar las operaciones del sistema de comunicación digital, la implementación de SDR no funcionará correctamente, produciendo errores de recepción o transmisión y un rendimiento deficiente

Materiales de laboratorio y trabajo de campo

Los sistemas SDR actuales usan en todo o alguna parte integrados del tipo FPGA (Field Programmable Gate Array) para mitigar los efectos de límites de procesamiento de los microprocesadores.

Un FPGA es un dispositivo electrónico basado en una matriz reconfigurable de bloques lógicos. Estos dispositivos, cuando se integran en un SDR, son extremadamente potentes y pueden ofrecer un alto rendimiento de procesamiento dado que proporcionan tanto el ancho de banda de entrada/salida como las capacidades de procesamiento para implementar tasas de muestreo en el orden de miles de millones de muestras por segundo y anchos de banda en el rango de gigahertz (D. F. Macedo, D. Guedes, L. F. M. Vieira, M. A. M. Vieira, and M. Nogueira, 2018).

Los FPGA se pueden programar para realizar tareas de procesamiento de señales de alta velocidad, lo que permite varias características de los SDR de alto rendimiento. Esto incluye capacidades de canalización, así como también de procesamiento (filtrado, decimación, interpolación, etc.) lo que permite una trans-recepción extremadamente rápida.

Además, permiten la modulación a una frecuencia intermedia (IF) antes de pasar a través de enlaces digitales al convertidor digital-analógico (DAC). El lado del receptor también se puede programar de manera similar para una funcionalidad de alto rendimiento. Por otro lado, al ser reconfigurables se puede asignar recursos para usar partes del FPGA adicionales para crear procesadores de señal digital (DSP) adaptados a la señal a ser procesada.

Los FPGA proporcionan características computacionalmente intensivas y son mucho más potentes que los CPU para realizar el procesamiento de señales en tiempo real. Son dispositivos de baja latencia debido a la arquitectura paralela del hardware y la naturaleza determinista, lo que los hace ideales para enlaces de baja latencia o para recepción de banda ancha tal como se ejemplifica en la Figura 32

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

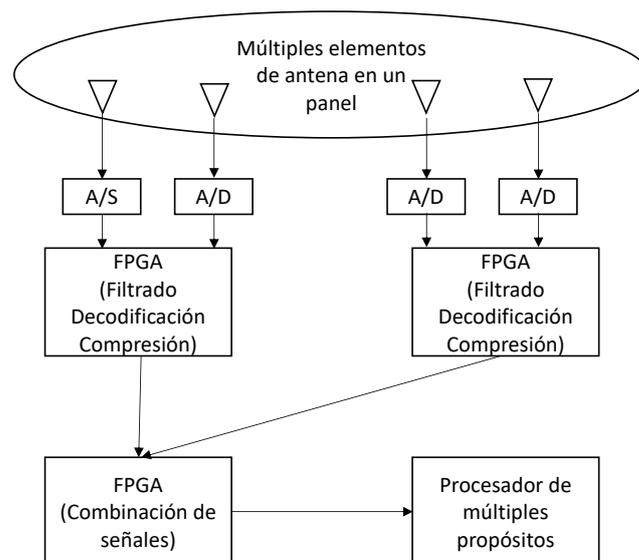


Figura 32. Diagrama de bloques de un receptor SDR basado en FPGA

5.4.2. LIMESDR

La placa LimeSDR es un transceptor SDR full-dúplex para el rango de frecuencias de 100 kHz a 3800 MHz (3,8 GHz). La placa se utiliza para el desarrollo y prueba de software para comunicaciones de radio de banda ancha. Las principales aplicaciones residen en métodos de transmisión como GSM, UMTS, LTE, 5G, IoT, WiFi o radar también son posibles gracias a su capacidad full-duplex y MIMO. El LimeSDR se utiliza mucho en la investigación y la educación, pero también es utilizado por los radioaficionados como base para la operación de SSB o DATV. La potencia de transmisión máxima es de 10 dBm (10 mW CW), el ancho máximo de captura es de 61,44 MHz, los convertidores analógicos digitales integrados tienen una resolución de 12 bits.

El LimeSDR es una plataforma de código abierto para la tecnología SDR. Todos los diagramas de circuitos, planos de perforación y montaje, etc. están disponibles bajo licencia abierta. El chip transceptor utilizado es un LMS7002M de Myriad HF, que implementa dos transceptores independientes. Esto permite la operación dúplex completa y MIMO 2x2. Las características del chip, como la frecuencia, el ancho de banda, la modulación, etc., se pueden cambiar fácilmente mediante la programación. La placa LimeSDR utiliza un FPGA Altera MAX10 que admite el procesamiento del flujo de datos, y cuenta con 256 MB de memoria. El LimeSDR ofrece una plataforma de desarrollo muy potente para el SDR (LimeSDR, 2022)

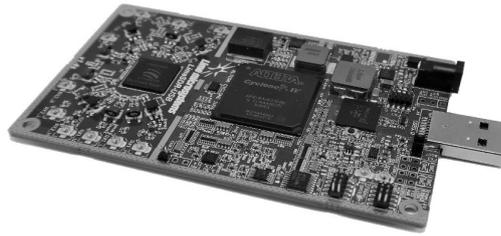


Figura 33. Imagen del Lime SDR

RF Transceiver: Lime Microsystems LMS7002M MIMO FPRF (Datasheet)
FPGA: Altera Cyclone IV EP4CE40F23 – also compatible with EP4CE30F23
Memory: 256 MBytes DDR2 SDRAM
USB 3.0 controller: Cypress USB 3.0 CYUSB3014-BZXC
Oscillator: Rakon RPT7050A @30.72MHz (Datasheet)
Continuous frequency range: 100 kHz – 3.8 GHz
Bandwidth: 61.44 MHz
RF connection: 10 U. FL connectors (6 RX, 4 TX)
Power Output (CW): up to 10 dBm
Multiplexing: 2×2 MIMO
Power: micro USB connector or optional external power supply
Status indicators: programmable LEDs
Dimensions: 100 mm x 60 mm

Tabla 20. Especificaciones técnicas del LimeSDR

5.5. HERRAMIENTAS DE SOFTWARE

5.5.1. CHIRPSTACK

ChirpStack es una suite de software basada código abierto que proporciona los componentes necesarios para soporte a redes LoRaWAN. Sus componentes forman una solución que incluye una interfaz web para la administración de dispositivos y API para la integración. La arquitectura modular permite la integración dentro de las infraestructuras existentes. Todos los componentes están licenciados bajo la licencia MIT y pueden ser utilizados con fines comerciales (*ChirpStack Architecture - ChirpStack Open-Source LoRaWAN® Network Server*, 2022)

5.5.1.1 FUNCIONALIDADES DE CHIRPSTACK

Soporte de clase A

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Los dispositivos finales de clase A permiten comunicaciones bidireccionales mediante las cuales la transmisión ascendente de cada dispositivo final es continuada por dos ventanas de recepción de enlace descendente. La transmisión por el dispositivo final se basa en las necesidades de comunicación del mismo con una pequeña variación en tiempo basada en una base de tiempo aleatorio (ALOHA).

Soporte de clase B

Los dispositivos finales de Clase B permiten más espacios de recepción. Además de las ventanas de recepción aleatoria de Clase A, los dispositivos de Clase B abren ventanas de recepción adicionales a tiempos programados. El dispositivo final abre su ventana de recepción a la hora programada si recibe un *beacon* sincronizado en tiempo desde la puerta de enlace.

Compatibilidad con Clase C

Los dispositivos finales de clase C tienen una ventana de recepción abierta casi continuamente, que solo se cierra cuando transmiten. El dispositivo final de Clase C usa más energía para operar que el de Clase A o Clase B, pero ofrecen la latencia más baja para la comunicación de servidor a dispositivo final.

Velocidad de datos adaptable

Cuando un dispositivo final tiene ADR habilitado, el servidor de red de ChirpStack se asegura de que el dispositivo funcione utilizando la velocidad de datos y la potencia transmisión más eficientes. Esto no solo ahorra energía en el lado del dispositivo, sino que también optimiza el uso del espectro radioeléctrico, reduciendo el riesgo de colisiones.

Registro de transacciones en vivo

El servidor de aplicaciones de ChirpStack proporciona registro de paquetes en tiempo real segmentado por gateway y dispositivo. Muestra todos los metadatos de transmisión y recepción, junto con los datos brutos LoRaWAN en un formato legible.

Reconfiguración de canales

Ya sea que desee utilizar un subconjunto de los canales definidos por LoRaWAN (por ejemplo, para la banda de EE. UU.) o desee configurar canales adicionales (por ejemplo, para la banda de la UE), el servidor de red ChirpStack se asegurará de que el dispositivo permanezca siempre

sincronizado con los canales configurados por la red (utilizando el campo CFList y / o mac-commands).

API e integración

Todos los componentes de ChirpStack proporcionan API gRPC y/o REST para la integración con servicios externos. De forma predeterminada, todos los datos de la aplicación se publican en un agente MQTT, sin embargo, las integraciones están disponibles para varios proveedores de nube, bases de datos y plataformas de visualización.

Compatible con LoRaWAN 1.0 y 1.1

El servidor de red ChirpStack admite simultáneamente los dispositivos LoRaWAN 1.0 y LoRaWAN 1.1, incluidas todas las revisiones y bandas de la especificación de parámetros regionales de LoRaWAN.

5.5.1.2 ARQUITECTURA DE CHIRPSTACK

El software se visualiza un diagrama de la arquitectura de los componentes de ChirpStack, cada uno de ellos puede ser ejecutado como un contenedor

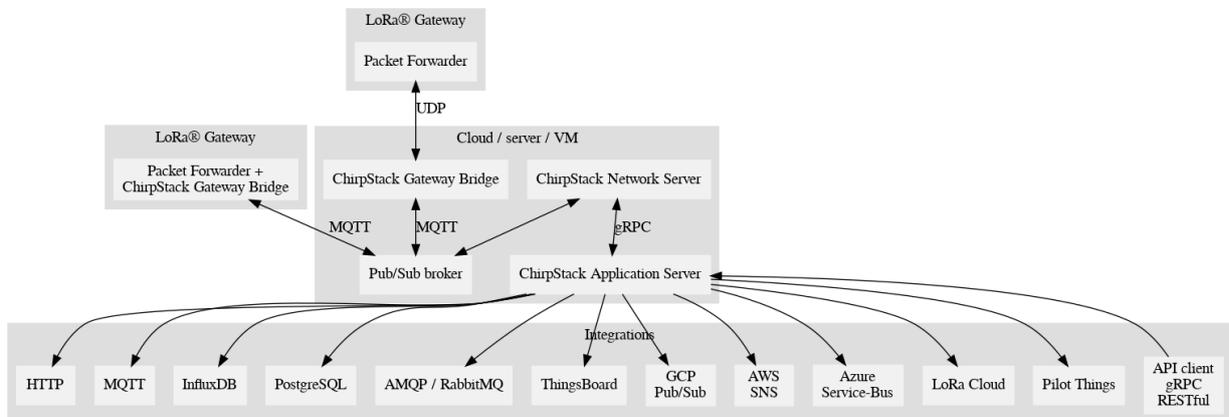


Figura 34. Arquitectura de ChirpStack

Puente de puerta de enlace ChirpStack

El puente de puerta de enlace ChirpStack se encuentra entre el gateway y el agente MQTT. Transforma el formato *Packet Forwarder* (como el protocolo Semtech UDP *Packet Forwarder*) en un formato de datos utilizado por los componentes de ChirpStack. También proporciona integraciones con varias plataformas en la nube como GCP Cloud IoT Core y Azure IoT Hub.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Servidor de red ChirpStack

El servidor de red ChirpStack es un servidor de red LoRaWAN, responsable de administrar el estado de la red. Tiene conocimiento de las activaciones de dispositivos en la red y es capaz de manejar solicitudes de unión cuando los dispositivos desean unirse a la red.

Cuando los datos son recibidos por varias puertas de enlace, el servidor de red ChirpStack deduplicará estos datos y los reenviará como una carga útil al servidor de aplicaciones ChirpStack. Cuando un servidor de aplicaciones necesita enviar datos a un dispositivo, el servidor de red mantendrá estos elementos en cola, hasta que pueda enviar a una de las puertas de enlace.

Servidor de aplicaciones ChirpStack

El servidor de aplicaciones ChirpStack es un servidor de aplicaciones LoRaWAN, compatible con el servidor de red ChirpStack. Proporciona una interfaz web y API para la gestión de usuarios, organizaciones, aplicaciones, puertas de enlace y dispositivos.

Los datos de enlace ascendente recibidos se reenvían a una o varias integraciones configuradas.

Aplicación final

La aplicación final recibe los datos del dispositivo a través de una de las integraciones configuradas. Puede usar la API de ChirpStack Application Server para programar una carga útil de enlace descendente a los dispositivos. El propósito de una aplicación final podría ser análisis, alertas, visualización de datos, acciones desencadenantes.

En el caso de esta tesis se utilizó la base de datos influxDB como aplicación final

5.5.2. INFLUXDB (BASE DE DATOS)

Influx DB es una base de datos de series temporales desarrollada en código abierto y escrita en lenguaje Go que es desarrollada por InfluxData. Está optimizada para la recuperación de datos con alta disponibilidad y manera mucho más rápida que bases de datos tradicionales optimizando el almacenamiento de datos de series temporales. Es usada principalmente en áreas tales como monitoreo de operaciones de redes, métricas de aplicaciones, datos de sensores de IoT y análisis en tiempo real (InfluxData, 2022)

InfluxDB es una base de datos de series temporales de alto rendimiento que puede almacenar datos que van desde cientos de miles de puntos por segundo. InfluxDB es un lenguaje de consulta de tipo SQL que se creó específicamente para datos de series temporales.

5.5.2.1 SERIE TEMPORAL

Una serie temporal es una colección de observaciones de datos a través de mediciones repetidas a lo largo del tiempo. Los datos de series temporales se indexan en orden temporal, que es una secuencia de puntos de datos.

5.5.2.2 BASE DE DATOS DE SERIES TEMPORALES

Una base de datos de series temporales (TSDB) es un sistema de base de datos que está optimizado para proporcionar datos de series temporales.

La base de datos de series temporales consiste en mediciones o eventos que se monitorean, graban, y procesan los datos, es decir, muestreos correlativos y agregados a lo largo del tiempo. Pueden ser datos de análisis de monitoreo de aplicaciones, métricas de servidores, datos sobre sensores, datos de mercado, datos de bolsa de valores en todos los mercados.

La base de datos de series temporales es capaz de ingerir millones de puntos de datos por segundo proporcionando un rendimiento de alto nivel.

El ejemplo clásico del mundo real de una serie temporal son los datos de precios de divisas de la bolsa de valores.

Una serie temporal es un conjunto de valores con una marca de tiempo para cada registro donde el tiempo es un componente significativo de los datos.

5.5.3. GRAFANA

Grafana es una herramienta que muestra datos de series temporales, es una herramienta de código abierto, con licencia Apache 2.0, diseñada por Torkel Ödegaard y creada en enero de 2014. Grafana está escrito en el lenguaje de programación Go (creado por Google) y Node.js.

Puede dar una vista general gráfica de la situación de una empresa u organización a partir de los datos recopilados, así como también permite a los usuarios ver sus datos a través de tablas y gráficos que se unifican en un panel o múltiples para una interpretación y comprensión más fáciles. También puede consultar y establecer alertas sobre su información y métricas desde cualquier lugar donde se almacene esa información, ya sean entornos de servidores tradicionales, clústeres de Kubernetes o varios servicios en la nube, etc.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Es capaz de permitir analizar más fácilmente los datos, identificar tendencias e inconsistencias y, en última instancia, hacer que los procesos sean más eficientes.

Los paneles de Grafana permiten ordenar los datos recopilados de diversas fuentes. Esos paneles se pueden compartir con otros miembros del equipo y otros equipos, lo que permite la colaboración y una exploración más extensa de los datos y sus implicaciones. Cree paneles específicamente para usted y su equipo y personalice sus paneles para crear las visualizaciones que desee, utilizando capacidades avanzadas de consulta y transformación.

Comprender todos los datos relevantes y las relaciones de datos es vital cuando se trata de encontrar la causa de un incidente o comportamiento inesperado del sistema lo más rápido posible. Grafana permite una visualización y movimiento de datos sin problemas entre los equipos y los miembros del equipo para que puedan llegar a la raíz de un problema rápidamente y resolverlo.

5.5.4. GNURADIO

GNU Radio es un kit de herramientas de desarrollo de código abierto que proporciona bloques de procesamiento de señales para implementar radios vía software. Se puede utilizar con hardware de RF externo de bajo costo para crear radios definidas por software, o directamente sin hardware en un entorno similar a la simulación. Es ampliamente utilizado en entornos, académicos y comerciales para apoyar tanto la investigación de comunicaciones inalámbricas como los sistemas de radio del mundo real (GNURadio, 2012).

GNU Radio realiza todo el procesamiento de las señales. Puede utilizarse para escribir aplicaciones para recibir datos o para insertar datos en flujos digitales, que luego se transmiten mediante hardware dedicado. GNU Radio tiene filtros, códigos de canal, elementos de sincronización, ecualizadores, demoduladores, vocoders, decodificadores y muchos otros elementos (elementos llamado bloques) que se encuentran típicamente en los sistemas de radio. Más importante aún, incluye un método para conectar estos bloques y luego administra cómo se pasan los datos de un bloque a otro, como se muestra en la Figura 35. Extender GNU Radio es muy simple; si encuentra un bloque específico que falta, puede crearlo y agregarlo rápidamente.

Dado que GNU Radio es un software, solo puede manejar datos digitales. Por lo general, las muestras de banda base complejas son el tipo de datos de entrada para los receptores y el tipo de datos de salida para los transmisores. El hardware analógico se utiliza para cambiar la señal a la

frecuencia central deseada. Dejando de lado ese requisito, cualquier tipo de datos se puede pasar de un bloque a otro, ya sean bits, bytes, vectores, o tipos de datos más complejos.

Las aplicaciones de GNU Radio se escriben principalmente utilizando el lenguaje de programación Python.

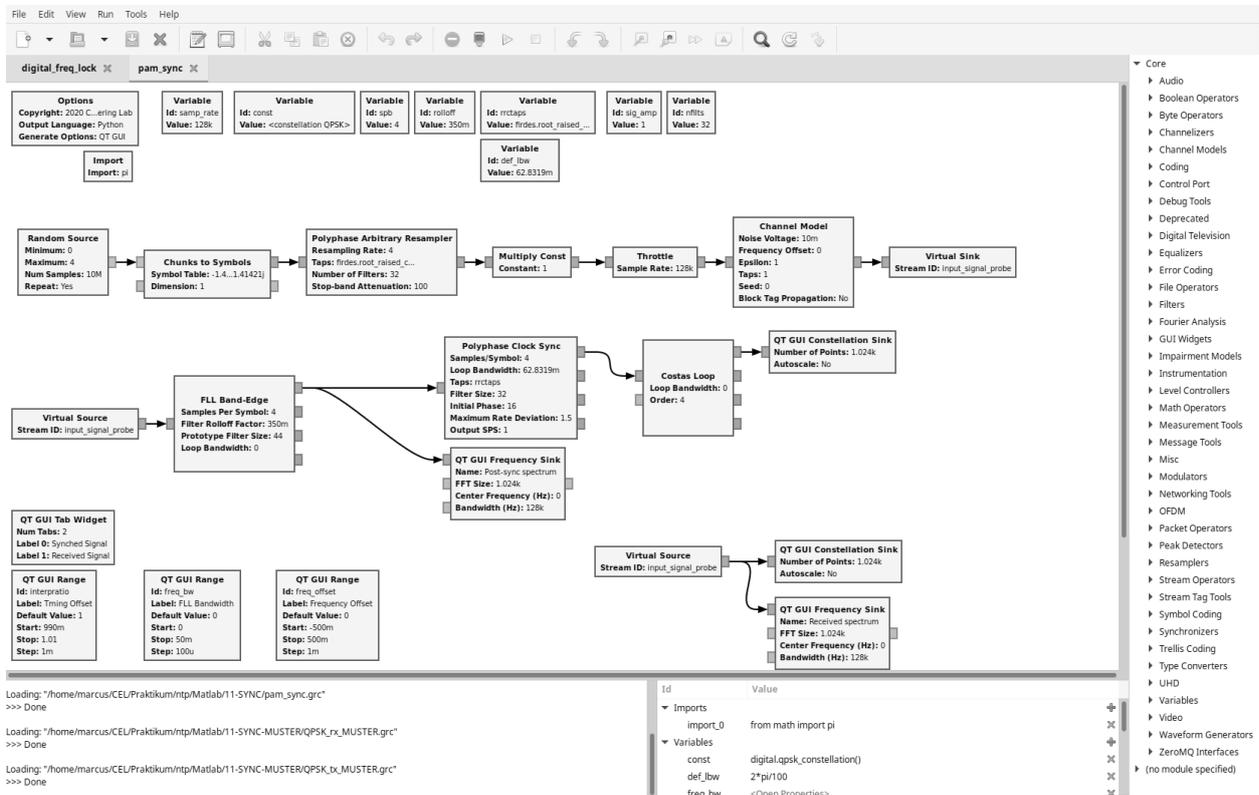


Figura 35. Interfaz gráfica de GNU Radio

5.6. CONCLUSIONES

Se puede concluir que la variedad de herramientas de software libre disponibles junto con los desarrollos de hardware de diseño abierto permite la implementación completa a muy bajo costo tanto de una suite de pruebas de una red LoRaWAN como también de una red productiva. Por otro lado, la utilización de *software defined radio* como analizador de espectro y de protocolos ofrece un gran avance en términos de agilidad comparado con herramientas dedicadas tradicionales.

6. MÉTODOS Y RESULTADOS DE ENSAYOS

6.1. RESEÑA DEL CAPÍTULO

En este capítulo se presenta un detalle de los diferentes métodos y procedimientos utilizados tanto en los ambientes de laboratorio como los de campo. Se detalla la realización del montaje y configuración de un entorno de servidores de red, así como de una base de datos para minado de información y una herramienta de analítica y gráficos. Posteriormente se efectúa una presentación detallada de las diferentes configuraciones realizadas en los dispositivos con el objetivo de obtener un entorno de red estable y que permita realizar los experimentos necesarios. A continuación, se presenta un detalle de los experimentos a ser realizados, primero para obtener valores de calibración y después para validar en el campo modelos teóricos y empíricos de propagación realizados en otros lugares del mundo. Se definen los escenarios de prueba para dos diferentes entornos, uno urbano con casas contiguas y edificios y otro rural. Los escenarios principalmente se definen para comprobar las hipótesis de las variables que más afectan al comportamiento de las redes tal como distancias o variaciones del terreno.

Finalmente se presentan los resultados de laboratorio donde se realizó la comprobación funcional de las redes LoRaWAN y los valores de calibración de los equipos de esta. Para después mostrar los resultados de las pruebas de campo con la correspondiente validación de los modelos aplicados y la correlación entre los escenarios y los valores obtenidos y así permitir utilizar algunos de estos valores como entradas a las simulaciones realizadas en el capítulo 7

6.2. INTRODUCCIÓN

La primera etapa del experimento es garantizar que los componentes necesarios estén configurados correctamente, esto a su vez permitirá garantizar la validez y precisión de los datos recopilados durante la investigación, por lo tanto, esta primera parte consistió en realizar ensayos de laboratorio de comunicaciones LoRaWAN en condiciones controladas para lo que se diseñó un entorno de experimentos tal cual se lo representa en la Figura 36.

Los principales objetivos de los ensayos de laboratorio son:

- Validación de estabilidad de recepción de datos en condiciones controladas a diferentes distancias y tasas de datos.
- Medición de parámetros de RF (SNR y RSSI) en condiciones controladas a diferentes distancias y tasas de datos.
- Diseño de un bloque de recepción de protocolo LoRa en GNURadio para el análisis de las comunicaciones.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

- Análisis de la trama LoRaWAN mediante SDR.
- Minado de datos recibidos en una base de datos temporal.
- Minado de datos de parámetros de RF en una base de datos temporal.
- Visualización de datos para reporte y análisis.

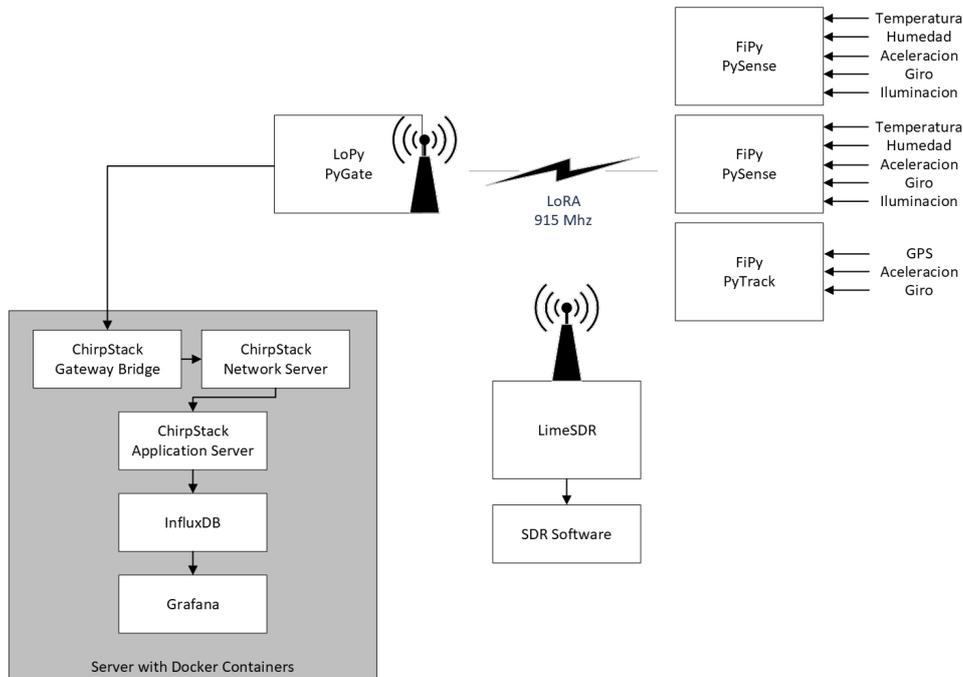


Figura 36. Diagrama de entorno de Laboratorio

6.3. CONFIGURACIÓN DE LOS COMPONENTES

6.3.1. CHIRPSTACK

Para realizar la instalación de la plataforma Chirpstack se corrieron los comandos en Código 1. Al mismo tiempo se actualiza la configuración para correr en la banda AU915 bloque 1 (916.8 -918.4 MHz)

```
mkdir ~/LoRaWAN
cd ~/LoRaWAN
git clone https://github.com/brocaar/chirpstack-docker.git
cd chirpstack-docker/
sed -i "s|eu868|au915_1|" configuration/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml
sed -i '/ "us915_1"/, /a\ "au915_1", ' configuration/chirpstack/chirpstack.toml
sudo docker-compose up -detach
```

Código 1. Instalación de Chirpstack

Métodos y resultados de ensayos

Una vez instalada la plataforma se puede acceder a la misma en la dirección IP del servidor en el Puerto 8080 con el usuario admin y la contraseña admin, tal como se ve en la Figura 37.

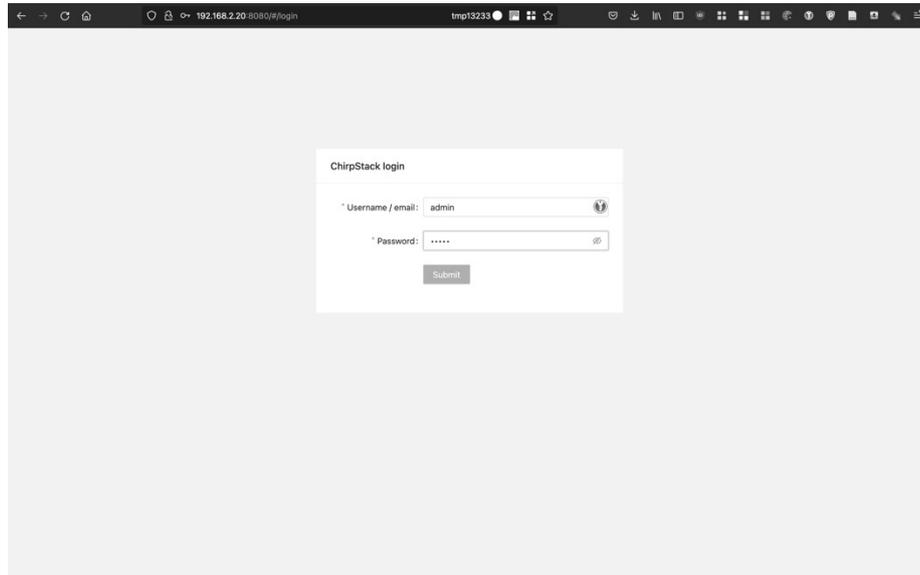


Figura 37. Chirpstack - Pantalla de acceso

Una vez instalada la plataforma se deben crear los perfiles de dispositivo, para los experimentos se utilizaron dos perfiles según el tipo de activación en la red LoRaWAN.

6.3.1.1 ABP (ACTIVATION BY PERSONALIZATION) VS OTAA (OVER THE AIR ACTIVATION)

Es importante hablar de las diferencias entre los modos de activación ABP y OTAA para los dispositivos LoRaWAN, y entender por qué en general se recomienda el uso de OTAA, aunque ABP es útil en algunos casos.

El identificador principal de un dispositivo en la red LoRaWAN es el DevEUI que es un identificador único de 64 bits asignado por el fabricante.

A diferencia del DevEUI, que identifica a un dispositivo final globalmente, un DevAddr de 32 bits identifica el dispositivo final dentro de la red actual y toda la comunicación después de unirse a la red se realiza con él. Un valor DevAddr consiste en NwkAddr (dirección del dispositivo final dentro de la red) precedido por un NwkID (identificador de red).

El DevAddr y las claves de sesión son asignadas a un dispositivo final durante un procedimiento denominado activación. LoRaWAN admite dos modos de activación de un dispositivo final: ABP (activación por personalización) y OTAA (activación por el aire).

6.3.1.2 ABP

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

En la activación de ABP, se asigna un DevAddr fijo y las claves de sesión se pre-cargan en el dispositivo final y permanecen fijos durante toda la vida útil del dispositivo final. En este modo, el dispositivo final evita el procedimiento de unión lo que resulta más simple, pero a costa de bastantes limitaciones que se explican a continuación.

6.3.1.3 OTAA

En la activación OTAA, el dispositivo final realiza un procedimiento de unión a la red LoRaWAN, durante el cual se asigna un DevAddr dinámico y se utilizan claves maestras para derivar claves de sesión. Por lo tanto, el DevAddr y las claves de sesión cambian cada vez que se realiza cada nueva sesión.

6.3.1.4 VENTAJAS DE OTAA VS ABP

Los inconvenientes de ABP surgen como consecuencia de sus principales características.

a) Los dispositivos ABP utilizan un DevAddr fijo.

Teniendo en cuenta la estructura de DevAddr y el hecho de que, con ABP, el NwkID es fijo, del dispositivo puede funcionar correctamente solo en su red predefinida. Incluso si una red permite registrar dispositivos finales con valores devAddr diferentes a los que se asignaron a esa red, el agente de paquetes no enrutará el tráfico desde esos dispositivos finales a la red correcta.

A los dispositivos OTAA se les asigna un nuevo DevAddr al establecer cada sesión. Esto les permite moverse a diferentes redes fácilmente.

b) Los dispositivos ABP utilizan claves de sesión fijas.

Los contadores de tramas en los dispositivos ABP no deben restablecerse durante su vida útil, de lo contrario, los mensajes de esos dispositivos finales pueden terminar perdiéndose. Cuando un dispositivo final ABP se queda sin contador de tramas, se acaba su vida útil.

Las claves de sesión están fijas en un dispositivo ABP, lo que significa que, si su seguridad es comprometida, no se pueden rotar, lo que es una grave amenaza para la seguridad.

A diferencia de ABP, los dispositivos finales OTAA renegocian los contadores de tramas y las claves de sesión al establecer cada sesión. Por lo tanto, la vida útil de un dispositivo OTAA no está condicionada por el contador de tramas.

c) Los dispositivos ABP utilizan parámetros de red fijos.

Métodos y resultados de ensayos

Al registrar un dispositivo ABP deberá proporcionar el retardo RX1, el desplazamiento de velocidad de datos RX1, el índice de velocidad de datos RX2, la frecuencia RX2 y una lista de frecuencias preestablecidas de fábrica. Si estos valores no están configurados correctamente, es posible que los enlaces ascendentes y/o descendentes no funcionen.

Los dispositivos finales OTAA renegocian los parámetros de red al establecer cada sesión, lo que significa que no hay configurarlos manualmente (The Things Industries, 2022).

En estos experimentos se utilizó ABP para poder forzar un dispositivo a una frecuencia y tasa de datos fija y predeterminada de tal manera de poder realizar mediciones específicas.

El perfil de dispositivo se define en la pestaña de *devices profiles* según la Figura 38.

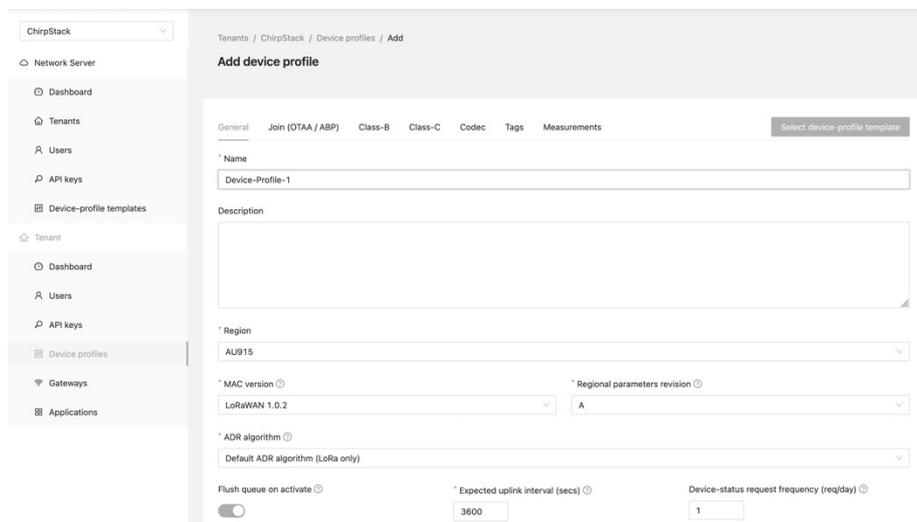


Figura 38. Chirpstack - Agregar un Perfil de dispositivo

Con el objetivo de no tener que definir un protocolo de transmisión de los diferentes tipos de datos sobre el protocolo LoRaWAN, se utiliza un protocolo que predefine la mayoría de los tipos de datos y estandariza la transmisión entre dispositivos finales y el servidor de aplicaciones. Este protocolo se llama CayenneLPP (*Low Power Payload*). Cayenne ofrece una forma fácil de enviar datos a través de LoRaWAN. Cayenne LPP cumple con las restricciones de tamaño de carga útil, que se pueden reducir a 11 bytes, y permite que el dispositivo final envíe múltiples tipos de datos de diferentes sensores a la vez. Existen librerías de CayenneLPP para micropython y es soportado en Chirpstack en la definición del dispositivo Figura 39.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

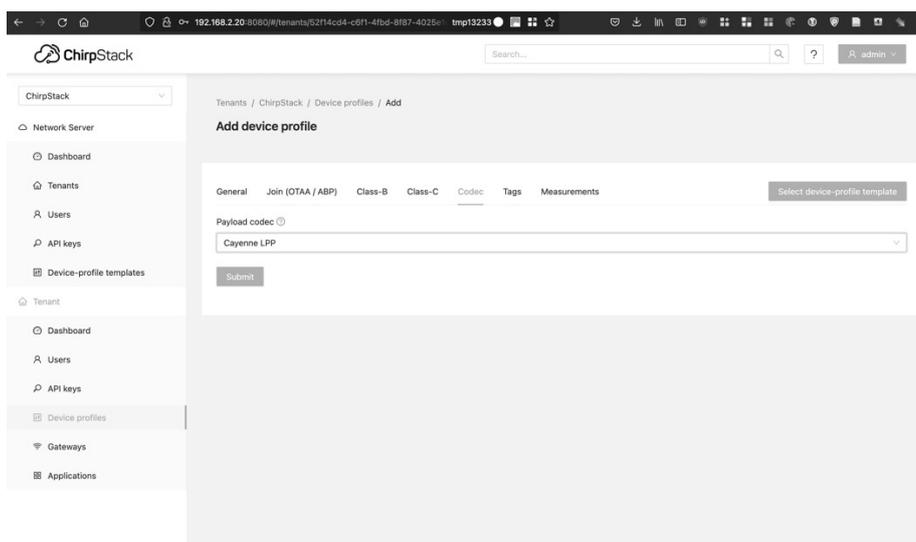


Figura 39. Chirpstack - Agregar un perfil de dispositivo – Codificador

Dada la naturaleza experimental de esta red, se definen los parámetros relevantes de los dispositivos de la red tal como el DevEUI, las claves de aplicación, etc. que son listados en la Tabla 21.

Dispositivo	DevEUI	
Pygate	00010203FFFFFFFF0	
OTAA		
PySense-1	00010203FFFFFFE0	app_key:D48982641B53A98874E3816EEFF218AB
PyTrack-1	00010203FFFFFFE1	app_key:E73954605A2B98801E712B308DF3F56F
PySense-2	00010203FFFFFFE2	app_key:C76D2A01A544AB6CD1A2F7F5458CC63C
ABP		
PySense-2b	00010203FFFFFFE3	dev_add:0054d85d nwk_key:f6e0b4cc24482cfc19d49e12e14c42ca app_key:00b44872b0d662a5038f581f5161af8a

Tabla 21. Definiciones de parámetros de dispositivos

Se define el Gateway en la pestaña apropiada, según el DevEUI determinado tal como se muestra en la Figura 40.

Métodos y resultados de ensayos

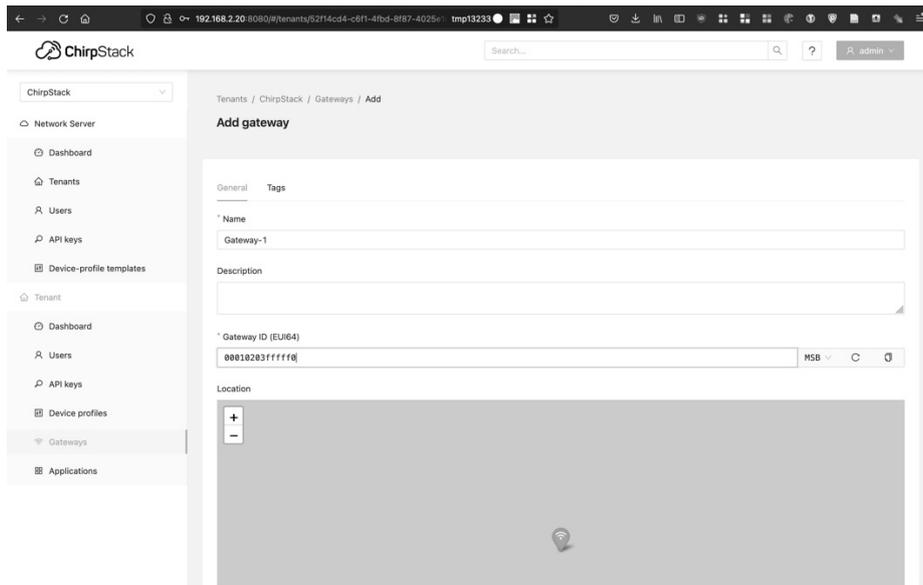


Figura 40. Chirpstack - Agregar un Gateway

Se define una nueva aplicación en la pestaña apropiada, tal como se muestra en la Figura 41.

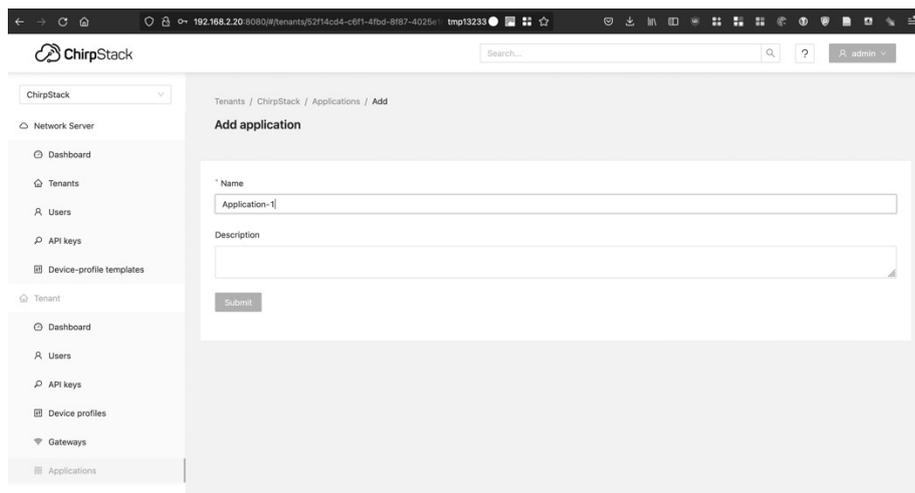


Figura 41. Chirpstack - Agregar una Aplicación

Se define una nueva integración en la aplicación, en este caso se agrega una integración con influxDB, a la cual se reenvían todos los datos recibidos desde los dispositivos, tal como se muestra en la Figura 42.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

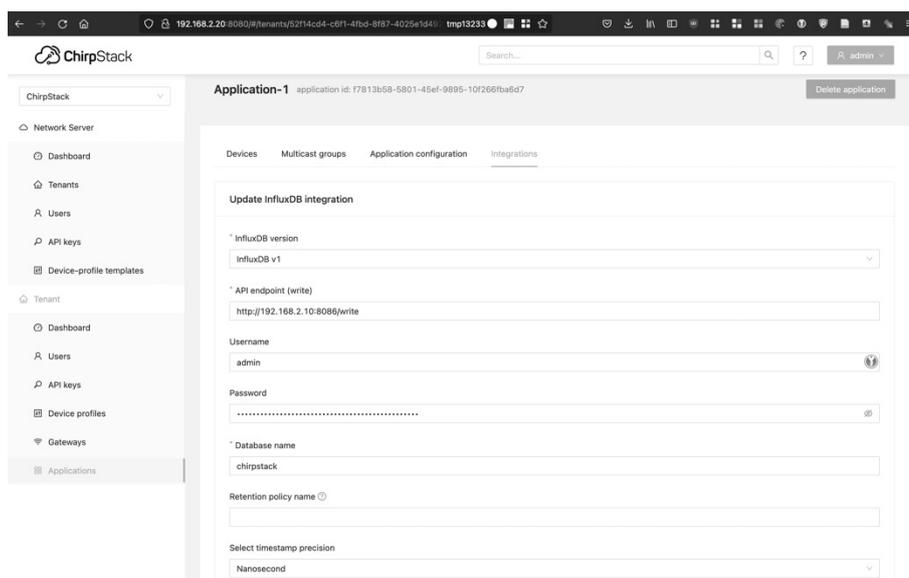


Figura 42. Chirpstack - Configurar la integración con la base de datos

Por último, se agregan los dispositivos finales dentro de la aplicación. Estos dispositivos se configuración con los parámetros definidos en la Tabla 21, según se muestra en las Figura 43 y Figura 44.

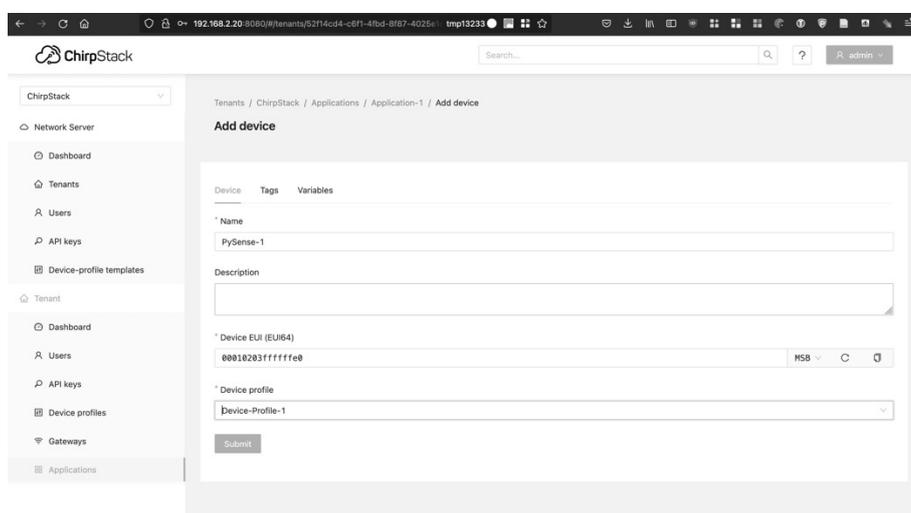


Figura 43. Chirpstack - Agregar un Dispositivo a la aplicación

Métodos y resultados de ensayos

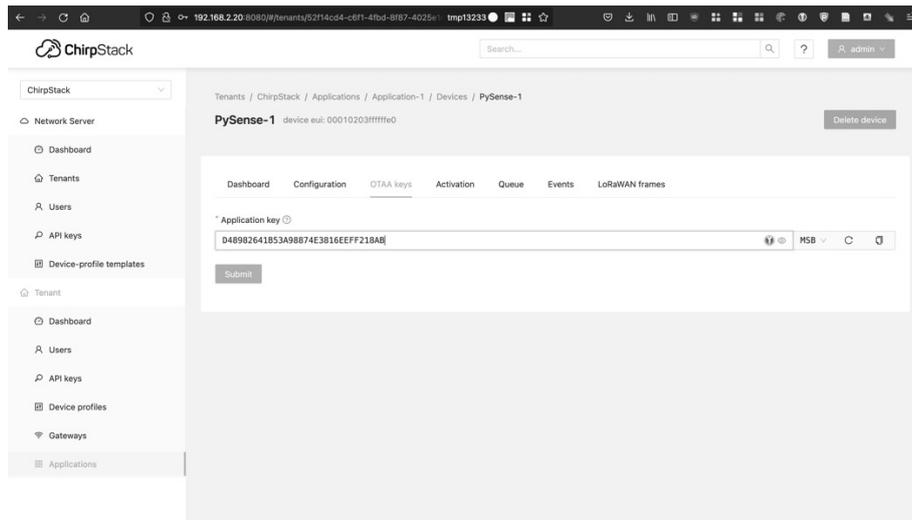


Figura 44. Chirpstack - Configurar el Application Key en el dispositivo

6.3.2. GATEWAY – PYGATE

Tal como se describió en la sección 5.3.4 se utilizó como Gateway un Pycom Pygate con una placa controladora LoPy. En la configuración de Laboratorio se conectó a una antena $\frac{1}{2}$ onda de 868 a 930 MHz con 2 dBi de ganancia. El gateway fue alimentado vía USB desde un adaptador USB AC. Una fotografía del gateway utilizado se muestra en la Figura 45.

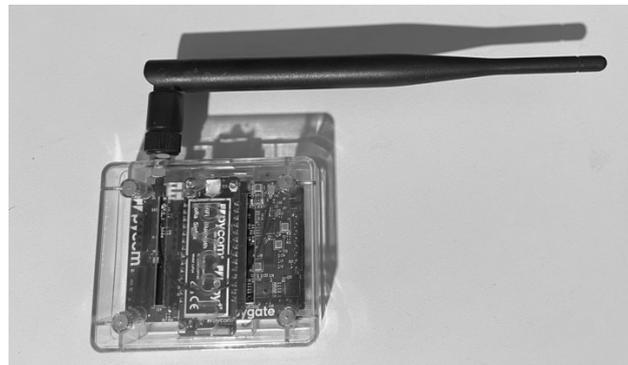


Figura 45. Fotografía de Gateway de Laboratorio

El gateway fue configurado con el script de código mostrado en Código 2 donde se modificó el DevEUI acorde a de la Tabla 21 que también fue configurado en Chirpstack, las coordenadas geográficas del lugar y la dirección IP del servidor de red. Los archivos completos de configuración son listados en el Anexo 1.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
cd ~/LoRaWAN
git clone https://github.com/peterjbrammer/Pycom_PyGate
cd Pycom_PyGate
# Cambia IP de LoRaWAN server
sed -i 's|router.au.thethings.network|192.168.2.20|' config_new.json
# Cambia DevEUI del Gateway
sed -i 's|"gateway_ID".*|"gateway_ID": "00010203ffffff0",|' config_new.json
# Cambia coordenadas GPS de referencia
sed -i 's|"ref_latitude".*|"ref_latitude": -35.187659,|' config_new.json
sed -i 's|"ref_longitude".*|"ref_longitude": -59.095730,|' config_new.json
sed -i 's|"ref_altitude".*|"ref_altitude": 34|' config_new.json
# Sube archivos a Pygate
ampy -p /dev/pygate put main.py
ampy -p /dev/pygate put config_new.json
# programa DevEUI en el pygate
screen /dev/pygate 115200
fo = open("/flash/sys/lpwan.mac", "wb")
mac_write=bytes([0x00,0x01,0x02,0x03,0xff,0xff,0xff,0xf0])
fo.write(mac_write)
fo.close()
<CTRL+A><CTRL+K>
```

Código 2. Script de configuración del Gateway

Todos los experimentos se llevaron a cabo utilizando los siguientes parámetros para el dispositivo:

- Subbanda Utilizada: 916.8 - 918.4 MHz
- Tamaño del canal: 8 canales de 125kHz y 1 canal de 500 kHz (OTAA Join)
- Factor de propagación (envío y recepción), variable = 7 a 12:
- Tasa de codificación (envío y recepción), fija = 4/5:
- Potencia de transmisión: 20dBm

El gateway se configuró con ocho canales de 125 kHz y un canal de 500 kHz tal como soporta la subbanda 1 del plan AU915.

Canal	Frecuencia	BW	SF
8	916800000	125 kHz.	7-12
9	917000000	125 kHz.	7-12
10	917200000	125 kHz.	7-12
11	917400000	125 kHz.	7-12
12	917600000	125 kHz.	7-12
13	917800000	125 kHz.	7-12
14	918000000	125 kHz.	7-12
15	918200000	125 kHz.	7-12
65	917500000	500 kHz.	8

Tabla 22. Tabla de frecuencias del gateway

6.3.3. SENSOR – PYSENSE

Métodos y resultados de ensayos

Tal como se describió en la sección 5.3.2 se utilizaron como sensores dos Pycom PySense con una placa controladora FiPy. En la configuración de Laboratorio se conectó a una antena $\frac{1}{2}$ onda de 868 a 930 MHz con 2 dBi de ganancia. Los sensores fueron alimentados vía USB desde un adaptador USB AC. Una fotografía del sensor utilizado se muestra en la Figura 46.

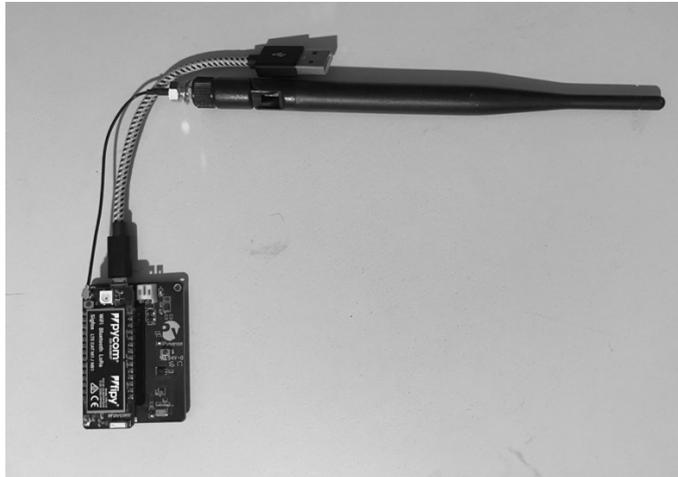


Figura 46. Fotografía del sensor PySense

El sensor fue configurado con el script de código mostrado en Código 3, donde se modificó el DevEUI acorde a de la Tabla 21 que también fue configurado en Chirpstack. Los archivos completos de configuración son listados en el Anexo 1. Con el objetivo de tener un flujo constante de datos se configuró la transmisión de datos recolectados cada 35 segundos.

Todos los experimentos se llevaron a cabo utilizando los siguientes parámetros para el dispositivo:

- Subbanda Utilizada: 916.8 - 918.4 MHz
- Tamaño del canal: 125 kHz
- Factor de propagación (envío y recepción), variable = 7 a 12:
- Tasa de codificación (envío y recepción), fija = 4/5:
- Potencia de transmisión: 20dBm

```
cd ~/LoRaWAN
mkdir pysense1
cd pysense1
wget https://raw.githubusercontent.com/brocaar/pycom-
examples/master/pysense-example/main.py
wget https://raw.githubusercontent.com/brocaar/pycom-
examples/master/pysense-example/lib/CayenneLPP.py
cp ../pycom-libraries/shields/lib/LIS2HH12.py .
cp ../pycom-libraries/shields/lib/LTR329ALS01.py .
cp ../pycom-libraries/shields/lib/MPL3115A2.py .
cp ../pycom-libraries/shields/lib/SI7006A20.py .
```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
cp ../pycom-libraries/shields/lib/pycoproc_1.py .
# Cambia a plan de frecuencia AU915y habilita ADR
sed -i "s|lora = LoRa(mode=LoRa.LORAWAN)|lora = LoRa(mode=LoRa.LORAWAN,
region=LoRa.AU915,adr=True)|" main.py
sed -i "s|app_eui = binascii.unhexlify('0101010101010101')|app_eui =
binascii.unhexlify('0000000000000000')|" main.py
# Cambia App Key según tabla
sed -i "s|app_key =
binascii.unhexlify('11B0282A189B75B0B4D2D8C7FA38548B')|app_key =
binascii.unhexlify('E73954605A2B98801E712B308DF3F56F')|" main.py
# Cambia Data Rate a DR6
sed -i "s|socket.SO_DR, 5|socket.SO_DR, 6|" main.py
sed -i "s|from pysense import Pysense|from pycoproc_1 import Pycoproc|"
main.py
sed -i "s|py = Pysense()|py = Pycoproc(Pycoproc.PYSENSE)|" main.py
# Sube Archivos al pysense
ampy -p /dev/pysense1 put CayenneLPP.py
ampy -p /dev/pysense1 put LIS2HH12.py
ampy -p /dev/pysense1 put LTR329ALS01.py
ampy -p /dev/pysense1 put MPL3115A2.py
ampy -p /dev/pysense1 put SI7006A20.py
ampy -p /dev/pysense1 put pycoproc_1.py
ampy -p /dev/pysense1 put main.py
ampy -p /dev/pysense1 reset
# programa DevEUI en el pygsense
screen /dev/pysense1 115200
fo = open("/flash/sys/lpwan.mac", "wb")
mac_write=bytes([0x00,0x01,0x02,0x03,0xff,0xff,0xff,0xe0])
fo.write(mac_write)
fo.close()
<CTRL+A><CTRL+K>
```

Código 3. Script de configuración del PySense 1

6.3.4. SENSOR – PYTRACK

Tal como se describió en la sección 5.3.3 se utilizó como sensor un Pycom PyTrack con una placa controladora FiPy. En la configuración de Laboratorio se conectó a una antena $\frac{1}{2}$ onda de 868 a 930 MHz con 2 dBi de ganancia. Los sensores fueron alimentados vía USB desde un adaptador USB AC. Una fotografía del gateway utilizado se muestra en la Figura 47.

Métodos y resultados de ensayos

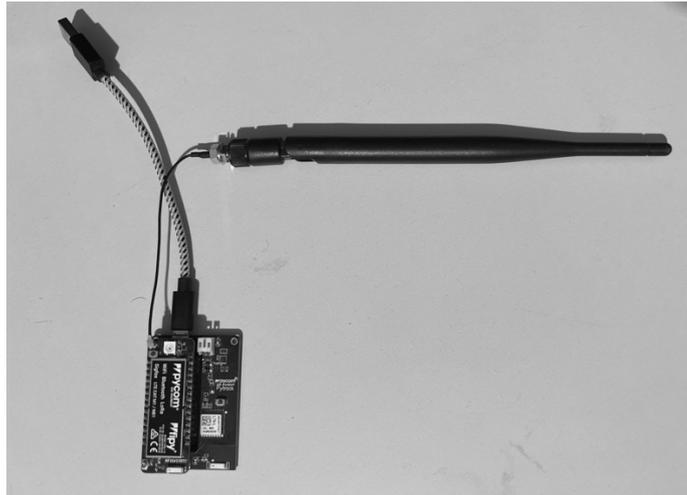


Figura 47. Fotografía del sensor PyTrack

El sensor fue configurado con el script de código mostrado en Código 4, donde se modificó el DevEUI acorde a de la Tabla 21 que también fue configurado en Chirpstack. Los archivos completos de configuración son listados en el Anexo 1. Con el objetivo de tener un flujo constante de datos se configuro la transmisión de datos recolectados cada 35 segundos.

Todos los experimentos se llevaron a cabo utilizando los siguientes parámetros para el dispositivo:

- Subbanda Utilizada: 916.8 - 918.4 MHz
- Tamaño del canal: 125 kHz
- Factor de propagación (envío y recepción), variable = 7 a 12:
- Tasa de codificación (envío y recepción), fija = 4/5:
- Potencia de transmisión: 20dBm

```
cd ~/LoRaWAN
mkdir pytrack
cd pytrack
wget https://raw.githubusercontent.com/brocaar/pycom-
examples/master/pytrack-example/main.py
wget https://raw.githubusercontent.com/brocaar/pycom-
examples/master/pysense-example/lib/CayenneLPP.py
cp ../pycom-libraries/shields/lib/LIS2HH12.py .
cp ../pycom-libraries/shields/lib/L76GNSS.py .
cp ../pycom-libraries/shields/lib/pycoproc_1.py .
# Cambia a plan de frecuencia AU915y habilita ADR
sed -i "s|lora = LoRa(mode=LoRa.LORAWAN)|lora = LoRa(mode=LoRa.LORAWAN,
region=LoRa.AU915,adr=True)|" main.py
sed -i "s|app_eui = binascii.unhexlify('0101010101010101')|app_eui =
binascii.unhexlify('0000000000000000')|" main.py
# Cambia App Key según tabla
sed -i "s|app_key =
binascii.unhexlify('11B0282A189B75B0B4D2D8C7FA38548B')|app_key =
binascii.unhexlify('D48982641B53A98874E3816EEFF218AB')|" main.py
# Cambia Data Rate a DR6
sed -i "s|socket.SO DR, 5|socket.SO DR, 6|" main.py
```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
sed -i "s|from pytrack import Pytrack|from pycoproc_1 import Pycoproc|"
main.py
sed -i "s|py = Pytrack()|py = Pycoproc(Pycoproc.PYTRACK)|" main.py# Sube
Archivos al pysense
ampy -p /dev/pytrack put CayenneLPP.py
ampy -p /dev/pytrack put LIS2HH12.py
ampy -p /dev/pytrack put L76GNSS.py
ampy -p /dev/pytrack put pycoproc_1.py
ampy -p /dev/pytrack put main.py
# programa DevEUI en el pytrack
screen /dev/pygate 115200
fo = open("/flash/sys/lpwan.mac", "wb")
mac_write=bytes([0x00,0x01,0x02,0x03,0xff,0xff,0xff,0xe1])
fo.write(mac_write)
fo.close()
<CTRL+A><CTRL+K>
```

Código 4. Script de configuración del PyTrack

6.3.5. LIMESDR

Tal como se describió en la sección 5.4.2, se utilizó el receptor SDR LimeSDR. En la configuración de Laboratorio se conectaron cuatro mini antenas optimizadas para 800-960 MHz, 1710-2170 MHz y 2400-2700 MHz. El receptor fue energizado y transfiere sus datos a través de una conexión USB 3.0 a una computadora personal. A efectos de estandarizar las mediciones de radio frecuencia, se colocó a exactamente un metro de las antenas de los sensores. Una fotografía del SDR utilizado se muestra en la Figura 48.

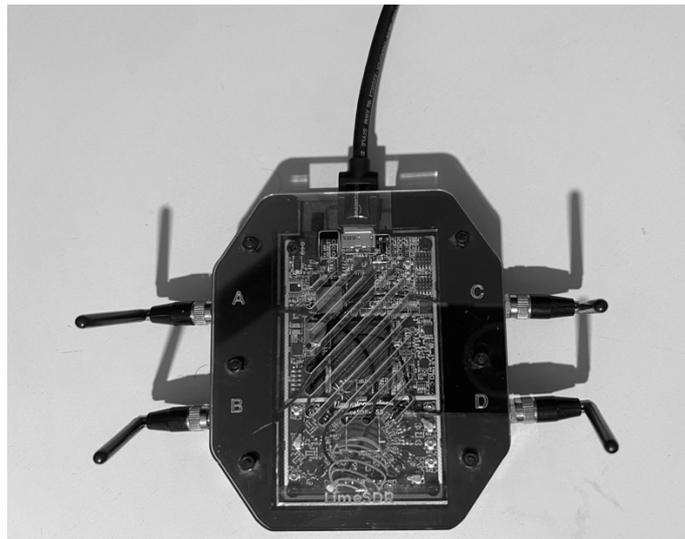


Figura 48. Fotografía del receptor LimeSDR

6.3.6. GNURADIO

Una parte muy importante del trabajo experimental de esta tesis fue la investigación de herramientas de radio definida por software (SDR), dicha investigación generó como trabajo relacionado un desarrollo de software que permitió la obtención de una patente y la generación de un proyecto de desarrollo de hardware. En la Figura 49 se puede ver una captura de pantalla de la aplicación gnu-radio-companion que incluye la configuración del receptor SDR, acoplada a un receptor de decodificador de LoRa, y en paralelo dos visualizaciones, una del tipo espectrógrafo y otra de análisis de espectro.

El receptor decodificador de LoRA es un proyecto de código abierto (gr-lora) (rpp0, 2016/2022) y permitió validar a nivel de radiofrecuencia las transmisiones del protocolo dado que implementa todas las características del esquema de modulación de capa física LoRa.

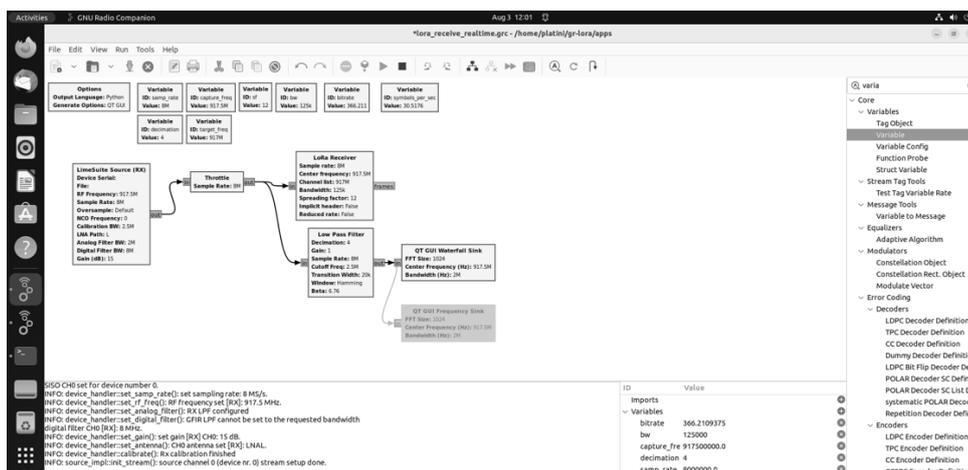


Figura 49. Diagrama en bloques de un proyecto GNURadio

6.3.7. GRAFANA

Se desarrolló un panel de visualización (*dashboard*) en Grafana que permitió visualizar gráficamente los parámetros más importantes de radiofrecuencia y de conectividad de cada uno de los sensores, así como también los datos de aplicación de cada uno de ellos. Estos datos son tomados en tiempo real desde la base de datos en tiempo real (InfluxDB). En las Figura 50 y Figura 51 se presenta una captura donde se visualizan los sensores PySense y PyTrack con sus mediciones de RSSI, SNR, acelerómetro, girómetro, barómetro, humedad, temperatura y GPS.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

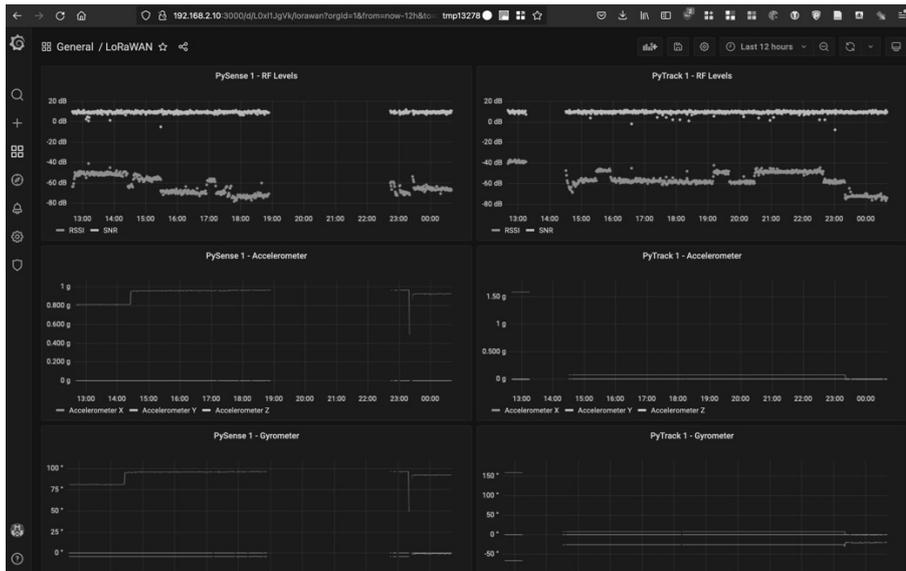


Figura 50. Panel de visualización de datos en Grafana

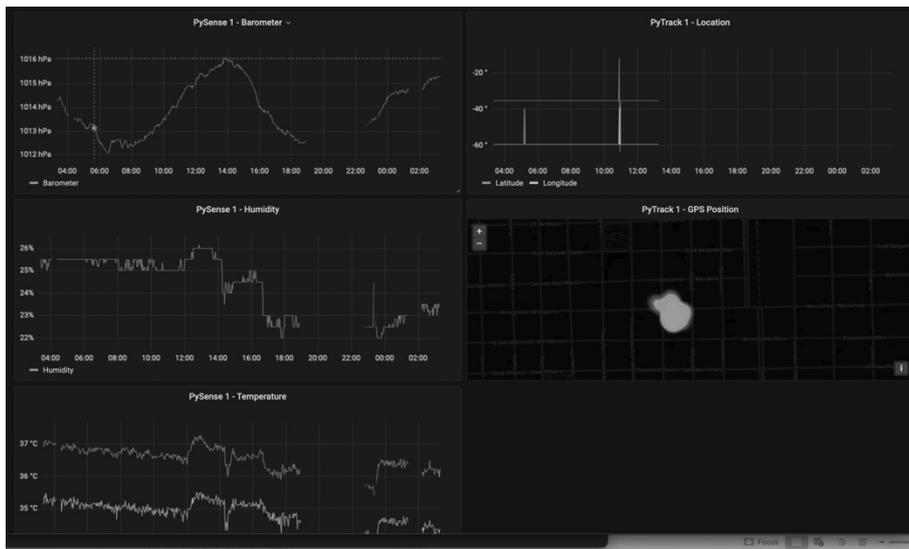


Figura 51. Panel de visualización de datos en Grafana

6.4. EXPERIMENTOS

El método de investigación de campo consistió en realizar experimentos en una red LoRaWAN en vivo que estuvo conformada por un gateway y tres dispositivos finales al mismo tiempo. Todos los componentes de hardware fueron alimentados por batería y en una configuración móvil para poder desplegar cualquiera de los componentes en un punto remoto.

Métodos y resultados de ensayos

Al igual en la referencia de laboratorio, de estos experimentos se extrajeron los valores RSSI y SNR junto con la pérdida de paquetes, y los parámetros de transmisión utilizados.

En todos los experimentos, el dispositivo de nodo final transmite un paquete con una carga útil fija determinada por el tipo de sensor y los datos transmitidos según Tabla 23 a intervalos de cada 35 segundos con una potencia de transmisión de 20dBm.

El factor de propagación puede variar de SF7 a SF12. El valor SF se elige automáticamente en función de la distancia (señal).

Carga Útil	Bytes
Actitud + Temperatura + Humedad + Iluminación	31
Actitud + GPS	27
Actitud	16
Status	8

Tabla 23. Carga útil de pruebas de campo

Se realizaron dos veces cada uno de los experimentos según las siguientes variantes

- a) Con las antenas alineadas, lo que significa que ambas están orientadas verticales
- b) Antenas desalineadas, una vertical (gateway) y la otra horizontal

Cada experimento constó de 50 transmisiones consecutivas para cada uno de los casos de prueba en cada uno de los entornos.

Se utilizó el valor medio de las 50 mediciones para generar un valor de referencia, en el anexo 2 se presentan el resumen de valor promedio y desvío para cada una de las mediciones.

6.4.1. LABORATORIO

En esta prueba se realizaron las mediciones dentro de entorno controlado, se colocó gateway en un trípode con la antena principal en posición vertical y un sensor en otro trípode también con la antena en forma vertical. Ambos dispositivos se colocaron a 1.5 metros del suelo.

Se definieron tres puntos de prueba definidos en la Tabla 24. El gateway se colocó en el punto 0 de forma estacionaria y con conexión a internet y a la infraestructura de colección de datos, mientras que los diferentes dispositivos finales se colocaron a diferentes distancias.

Punto de medición	Distancia al gateway
A	1m
B	2m
C	4m

Tabla 24. Distancias de puntos de medición en laboratorio

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

6.4.2. RURAL – ABIERTO

Se configuro el equipo en un lugar donde era posible tener al menos dos kilómetros con una línea de visión libre. En función de esto, se definieron cuatro puntos de prueba definidos en la Tabla 25 y reflejados geográficamente en la Figura 52 donde se puede ver la ubicación y las distancias. El gateway se colocó en el punto 0 de forma estacionaria y con conexión a internet y a la infraestructura de colección de datos, mientras que los diferentes dispositivos finales se colocaron a diferentes distancias.

Punto de medición	Distancia al gateway
A	100m
B	500m
C	1000m
D	2000m

Tabla 25. Distancias de puntos de medición en escenario rural



Figura 52. Geolocalización de puntos de medición en escenario rural

6.4.3. URBANO - SEMIDENSO

En esta prueba se realizaron las mediciones dentro de una ciudad que tiene casas hechas de diferentes materiales, aunque todas se encuentran al nivel de la superficie.

Se definieron tres puntos de prueba definidos en la Tabla 26 y reflejados geográficamente en la Figura 53 donde se puede ver la ubicación y las distancias. El gateway se colocó en el punto 0 de

Métodos y resultados de ensayos

forma estacionaria y con conexión a internet y a la infraestructura de colección de datos, mientras que los diferentes dispositivos finales se colocaron a diferentes distancias.

Punto de medición	Distancia al gateway
A	150m
B	300m
C	500m

Tabla 26. Distancias de puntos de medición en escenario urbano



Figura 53. Geolocalización de puntos de medición en escenario urbano

6.5. RESULTADOS

6.5.1. LABORATORIO

Se validaron los parámetros de RF según la configuración realizada en el gateway y representada por la Tabla 22.

6.5.1.1 ESPECTROGRAMA O DIAGRAMA DE CASCADA

El espectrograma es un diagrama de tiempo-frecuencia, también conocido como diagrama de cascada.

El eje X del espectrograma muestra las frecuencias de izquierda a derecha. El eje Y el tiempo desde arriba hacia abajo.

Los colores en el espectrograma representan las intensidades de señal de una frecuencia en un momento determinado. Un color intenso significa que se registró una fuerte intensidad de esa frecuencia en ese momento.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

El mapeo de color es personalizable seleccionando un ajuste preestablecido de mapa de color, a menudo el azul representa baja intensidad, el rojo es la intensidad más alta registrada. En el caso de estos ensayos de uso gris oscuro como alta intensidad y gris claro como baja intensidad,

A efectos de poder utilizar un patrón conocido en el espectrograma, se configuró uno de los sensores para transmitir información de telemetría de 4 bytes en un canal específico (917,0 MHz.) de 125 kHz de ancho de banda, se configuraron los diferentes valores de factores de dispersión en las diferentes pruebas y de validaron ancho banda y duración de los símbolos

En la Figura 54 se muestra la captura que más fácilmente permite visualizar las características de la señal, la elección de este factor de dispersión se realizó debido a que los símbolos son más largos temporalmente y por lo tanto permiten una captura más detallada por el SDR.

Según la Ecuación 18, la tasa de símbolos es igual a 30.51 símbolos/se es decir cada símbolo con SF12 debe durar 32 mseg.

$$T_S = \frac{BW}{2^{SF}}$$

Ecuación 18. Periodo de símbolos LoRaWAN

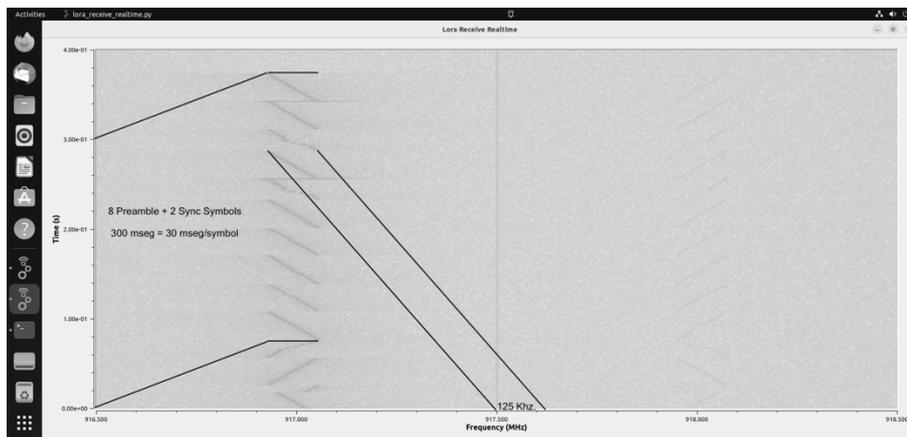


Figura 54. Espectrograma o diagrama de intensidad de señal con respecto a frecuencia y tiempo de una transmisión LoRaWAN - SF12 - 125 kHz

Se validó el tiempo de duración de los símbolos y el ancho de banda utilizado para cada una de las señales.

6.5.1.2 ANÁLISIS DE ESPECTRO

Se realizaron mediciones de la pureza espectral de las señales en cada uno de los canales con respecto a las adyacentes. Así como también del piso de ruido y la ausencia de interferentes en las condiciones de laboratorio.

En la Figura 55 se visualiza un max-hold de 1 minuto de múltiples transmisiones en la frecuencia de 917.0 MHz y una transmisión en la de 118 donde se evidencia claramente la forma de campana del espectro y un piso de ruido en los -110 dBm. No se observaron interferentes ni armónicas.

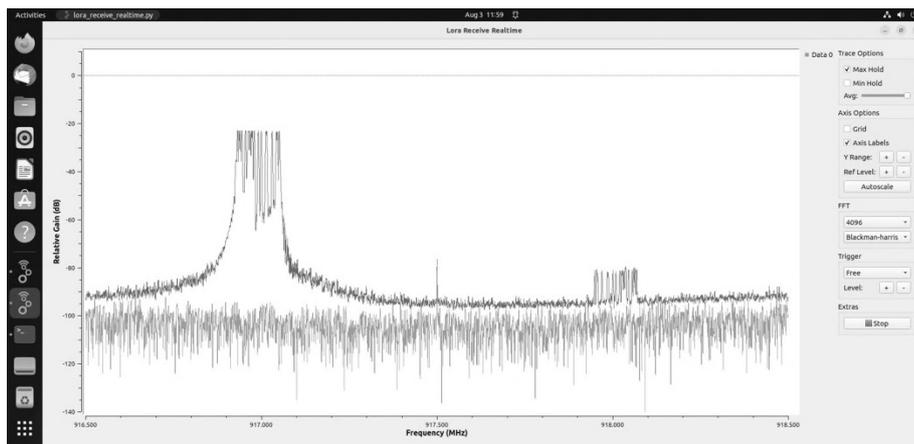


Figura 55. Análisis de espectro de un canal de 125 kHz en 917.0 MHz

En la Figura 56 se visualiza un max-hold de 10s minuto de múltiples transmisiones en los ocho canales de la subbanda 1 se evidencia claramente la forma de campana de los espectros y un piso de ruido en los -110 dBm. No se observaron interferentes ni armónicas.

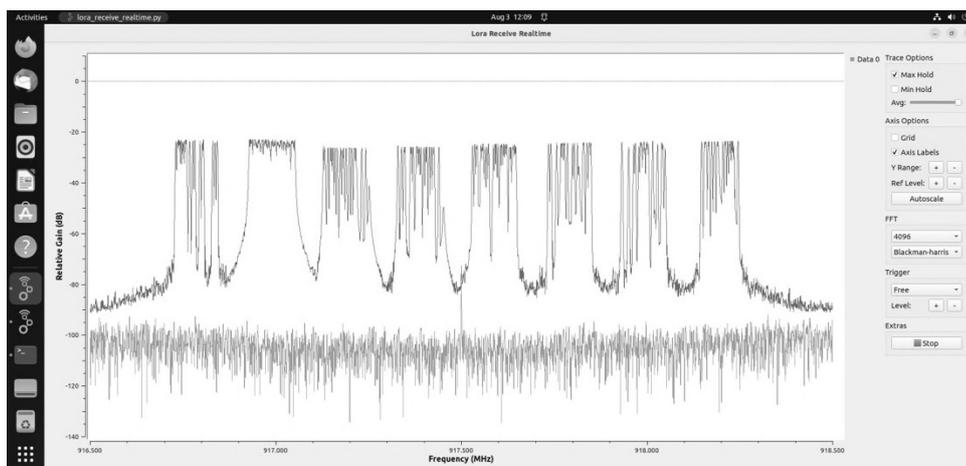


Figura 56. Análisis de espectro de los ocho canales de 125 kHz en la subbanda 1

6.5.1.3 MEDICIONES DE RSSI Y CALIBRACIÓN

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Tal como se mencionó en 6.4.1, se realizaron 50 transmisiones en cada uno de los puntos de medición establecidos. Una vez realizadas estas mediciones se tomaron los valores de RSSI medidos por el gateway de datos de la base de dato temporal y se graficaron en la Figura 57 de la forma de *boxplot*. En el anexo 2 se presenta una tabla con los valores resumidos para cada uno de los puntos.

Un punto muy importante de este experimento fue obtener los valores de calibración del sistema gateway / sensor.

$$RSSI = P_T - PL(d)$$

Ecuación 19. Potencia de señal recibida en función de transmitida y atenuación

La Ecuación 19 marca la relación entre el RSSI, la potencia de transmisión P_T que incluye las ganancias y atenuaciones de los componentes del sistema y la atenuación del camino $PL(d)$. (Huda Ali Hashim et al., 2018)

$$PL(d) = PL_0 + 10n \log \left(\frac{d}{d_0} \right) + X_\sigma$$

Ecuación 20. Modelo log-normal atenuación

Para calcular la atenuación del camino se utilizó el modelo log-normal de atenuación (Theodore Rappaport, 2002). En este modelo se define una atenuación de referencia PL_0 , calculada a una distancia de referencia según el modelo de atenuación en el espacio libre (FSPL) definido en Ecuación 17, un valor n que depende del entorno de propagación y que por ejemplo en el espacio libre tiene un valor de 2, en un entorno con obstrucciones tiene valores más grandes y en entornos interiores con reflexiones de señal más pequeños según se ve en la Tabla 27.

Entorno	Exponente de Atenuación n
Espacio Libre	2
Línea de vista en espacio cerrado	1,6 a 1,8
Espacio cerrado obstruido	4 a 6
Espacio cerrado fabricas	2 a 3
Espacio urbano	2,7 a 3,5
Espacio exterior obstruido	3 a 5

Tabla 27. Exponentes de atenuación en diferentes entornos

$$RSSI = P_T - PL_0 - 10n \log \left(\frac{d}{d_0} \right) - X_\sigma$$

Ecuación 21. Modelo log-normal atenuación

Un punto importante a evaluar el error del amplificador de potencia de los módulos FiPy dado que según (Vázquez-Rodas et al., 2020) puede ser significativo cuando el módulo transmite a máxima potencia $P_T = 20 \text{ dBm}$.

Realizando un mapeo polinómico de las mediciones de RSSI obtenidas a las diferentes distancias se obtiene un valor reflejado en Ecuación 22.

$$RSSI = -24.81 - 1.61 \left(10 \log_{10} \left(\frac{d}{d_0} \right) \right) - X_\sigma$$

Ecuación 22. RSSI modelado de datos

Como primera conclusión se puede decir que $n = 1.61$ esta dentro de los valores esperados en la Tabla 27.

Ahora bien, aplicando el valor de $n = 1.61$ a la y considerando $P_T = 20 \text{ dBm}$. La RSSI teórica resultará la reflejada en la Ecuación 23.

$$RSSI = -11.53 - 1.61 \left(10 \log_{10} \left(\frac{d}{d_0} \right) \right) - X_\sigma$$

Ecuación 23. RSSI teórico con $n=1.61$

De comparar las dos expresiones se identifica una diferencia de -13.28 dB. que corresponden al error del transmisor del FiPy cuando transmite a máxima potencia. Este valor de corrección será utilizado en los análisis de las mediciones de campo para validar los modelos teóricos de propagación.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

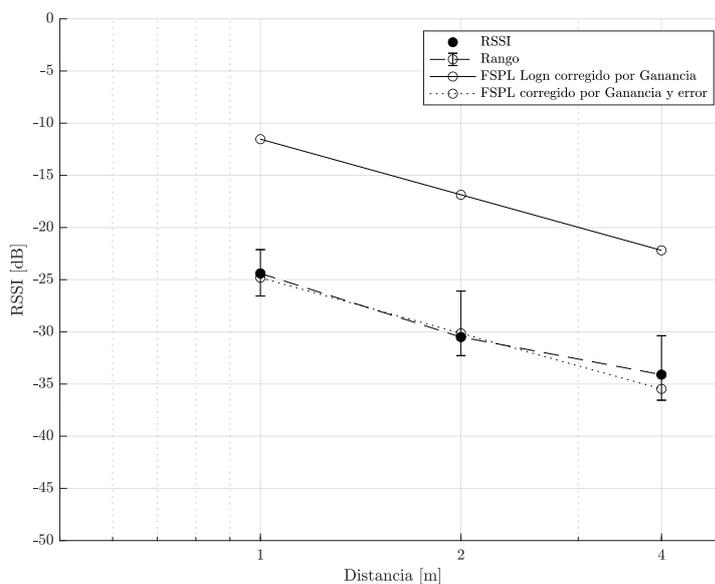


Figura 57. Mediciones de RSSI vs Distancia – Laboratorio

6.5.2. MEDICIONES DE RSSI, SNR Y PERDIDA DE PAQUETES - RURAL

Tal como se mencionó en 6.4.2, se realizaron 50 transmisiones en cada uno de los puntos de medición establecidos. Una vez realizadas estas mediciones se tomaron los valores de RSSI, SNR y paquetes entregados, medidos por el gateway y registrados en la base de datos temporal.

Los datos de RSSI se graficaron en la Figura 57 de la forma de *boxplot* y en el anexo 2 se presenta una tabla con los valores resumidos para cada uno de los puntos. En las mediciones de RSSI para las dos orientaciones de antenas, los resultados reflejaron un comportamiento que permitió concluir que los resultados se acercan a los esperados por el modelo de atenuación Hata para escenario de campo abierto.

A forma de referencia, en el gráfico se representó la expresión teórica de dicho modelo como línea sólida y se la ajustó con la potencia salida del transmisor de los sensores (+20 dBm) y con las pérdidas del sistema calculadas experimentalmente en el laboratorio.

Los datos de relación señal a ruido se graficaron en la Figura 59 de la forma de *boxplot* y en el anexo 2 se presenta una tabla con los valores resumidos para cada uno de los puntos. En las mediciones de SNR para las dos orientaciones de antenas, los resultados reflejaron un comportamiento acorde al esperado con respecto a la relación señal a ruido en función de las

Métodos y resultados de ensayos

distancias para un escenario en campo abierto donde a modo de referencia se representaron los valores de umbral para los factores de dispersión SF 7 y SF8, que proveen los rendimientos más altos y donde incluso a la distancia máxima medida fueron preponderantes.

Por otro lado, los resultados de porcentaje de entrega exitosa de paquetes representados en la Figura 60, claramente se correlacionan con los muy buenos valores de relación señal a ruido observados.

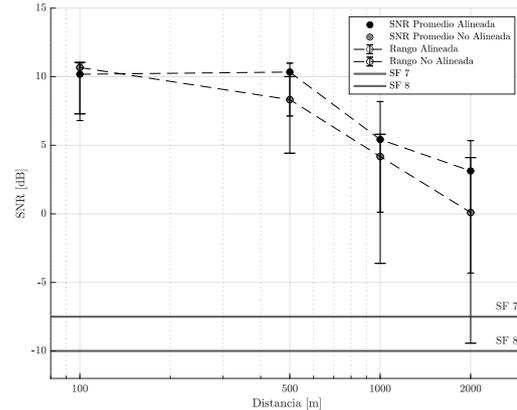
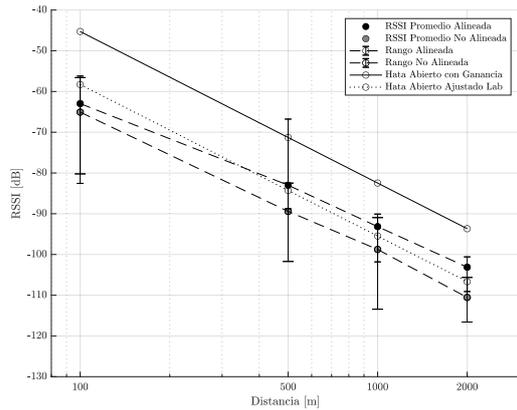


Figura 58. Mediciones de RSSI vs Distancia - Rural

Figura 59. Mediciones de SNR vs Distancia - Rural

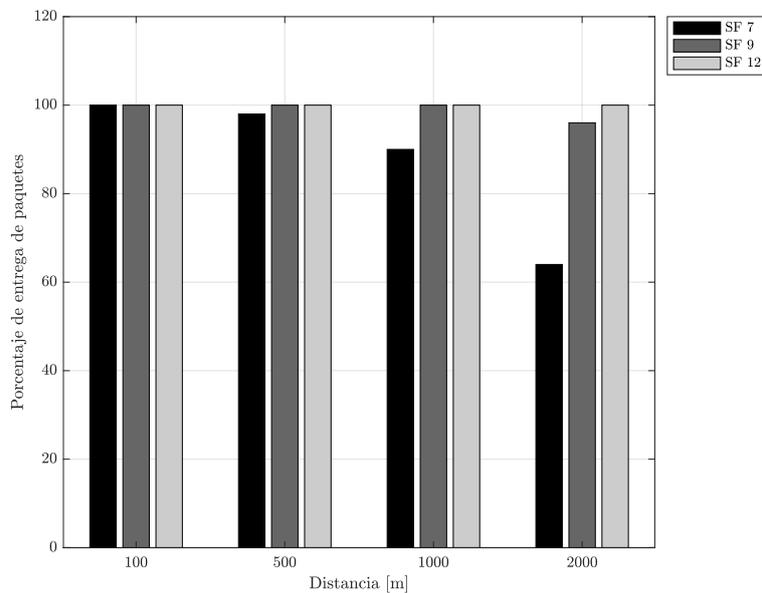


Figura 60. Porcentaje de entrega de paquetes - Rural

6.5.3. MEDICIONES DE RSSI, SNR Y PERDIDA DE PAQUETES - URBANO

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Tal como se mencionó en 6.4.3, se realizaron 50 transmisiones en cada uno de los puntos de medición establecidos. Una vez realizadas estas mediciones se tomaron los valores de RSSI, SNR y paquetes entregados, medidos por el gateway y registrados en la base de datos temporal.

Los datos de RSSI se graficaron en la Figura 61 de la forma de *boxplot* y en el anexo 2 se presenta una tabla con los valores sumariados para cada uno de los puntos. En las mediciones de RSSI para las dos orientaciones de antenas, los resultados reflejaron un comportamiento que permitió concluir que los resultados se acercan a los esperados por el modelo de atenuación Hata para escenario urbano y al de Villardi.

A forma de referencia, en el grafico se representó la expresión teórica de dichos modelos como línea sólida y se la ajusto con la potencia salida del transmisor de los sensores (+20 dBm) y con las pérdidas del sistema calculadas experimentalmente en el laboratorio.

Los datos de relación señal a ruido se graficaron en la Figura 59 de la forma de *boxplot* y en el anexo 2 se presenta una tabla con los valores sumariados para cada uno de los puntos. En las mediciones de SNR para las dos orientaciones de antenas, los resultados reflejaron comportamientos muy variables debido a la atenuación variable de los caminos en los escenarios urbanos, el comportamiento acorde al esperado con respecto a la relación señal a ruido en función de las distancias para un escenario urbano donde a modo de referencia se representaron los valores de umbral para los factores de dispersión SF 7 a SF11, cabe destacar que a 500 metros la mayoría de las transmisiones resultaron no recibidas por el gateway,.

Por otro lado, los resultados de porcentaje de entrega exitosa de paquetes representados en la Figura 62, se corresponden con los valores de RSSI que fuertemente descendentes y a las relaciones señal a ruido muy pobres para el punto más lejano.

Métodos y resultados de ensayos

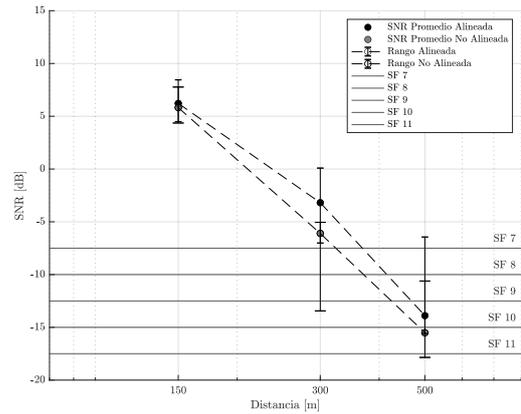
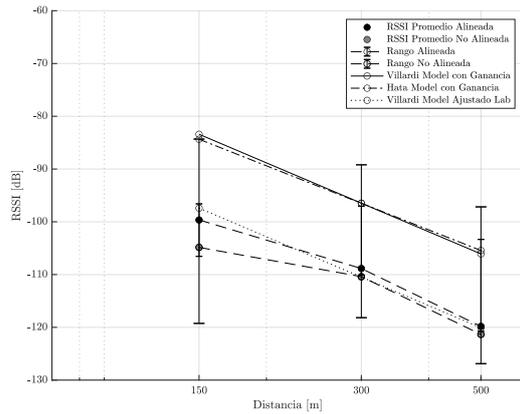


Figura 61. Mediciones de RSSI vs Distancia - Urbano

Figura 62. Mediciones de SNR vs Distancia - Urbano

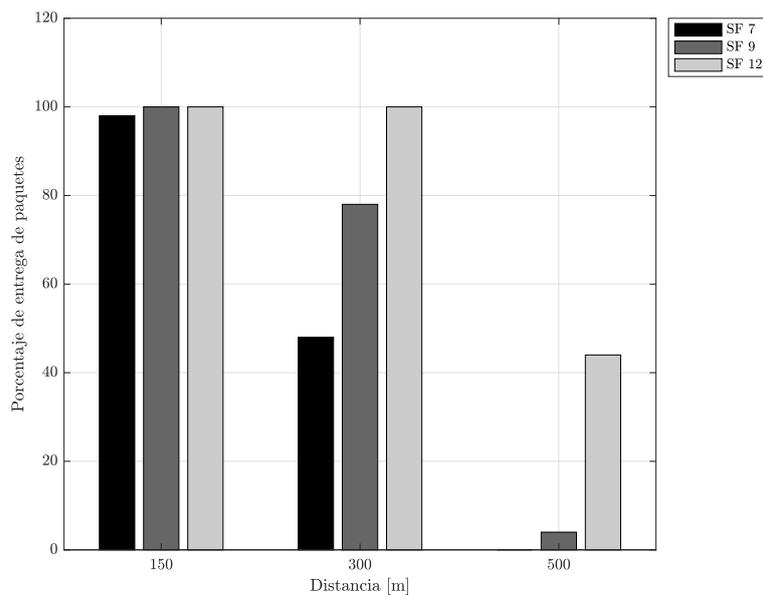


Figura 63. Porcentaje de entrega de paquetes – Urbano

6.6. CONCLUSIONES

Este capítulo permitió llegar a las siguientes conclusiones: Los modelos de propagación de Hakamura-Hata y Villardi aproximan de manera muy precisa el comportamiento de una red LoRaWAN en las condiciones de los casos de estudio tanto rural como urbano propuestos. En condiciones urbanas fue muy difícil alcanzar los 500 metros de cobertura sin una pérdida de paquetes considerable algo que se correlaciona con los análisis teóricos realizados en el capítulo

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

4. En el escenario rural la cobertura fue mucho más amplia y se lograron distancias mayores de 2000 metros sin una pérdida de datos significativa. Se comprobó que la orientación de las antenas tiene influencia en la cobertura y se vuelve más relevantes cuando las condiciones de señal se vuelven marginales.

Por último, se verificó que la activación con el modelo OTAA provee un mecanismo mucho más sencillo y estable de operación en una red productiva con condiciones variables, mientras que el ABP es mucho más útil para pruebas de laboratorio, pero mucho menos escalable en grandes despliegues.

7. SIMULACIONES

7.1. RESEÑA DEL CAPÍTULO

En este capítulo en primer lugar se presenta el porqué de la necesidad de realizar un análisis de simulación y posteriormente se realizó un análisis de diferentes herramientas de simulación de redes LoRaWAN junto con la definición de un conjunto de indicadores que permitirían validar la hipótesis que la cantidad de dispositivos por área geográfica tiene una influencia en la performance de la red utilizando como entrada los modelos de propagación validados en las mediciones de campo. A continuación, se presentan los resultados de cada una de las pruebas de simulación para los escenarios rural y urbano mostrando los efectos del aumento de la cantidad de dispositivos en la cantidad de información entregada y en la eficiencia total de la red

7.2. INTRODUCCIÓN

En el capítulo 6 se realizaron múltiples ensayos donde se validó la aplicación de los modelos de Hata y de Villardi en los escenarios de prueba definidos como laboratorio, urbano y rural.

Si bien esas pruebas permitieron medir y validar valores de intensidad de señal, factores de dispersión y relación señal ruido, las mismas se realizaron con una cantidad muy limitada de sensores (tres como máximo simultáneamente) y por lo tanto ellas no permiten analizar el impacto de una cantidad significativa de dispositivos en los escenarios de prueba.

Al mismo tiempo es muy complejo obtener y desplegar, a efectos de estas pruebas, una cantidad significativa de sensores por lo que se consideró un modelo de simulación basado en los parámetros obtenidos de las pruebas reales.

Como se mencionó anteriormente la Internet de las Cosas (IoT) es una de las tecnologías emergentes más importantes, que abarca una enorme variedad de posibles aplicaciones con un creciente número y variedad de dispositivos. Por lo tanto, fue fundamental simular el trabajo de dichos sistemas y aplicaciones antes y durante de las implementaciones reales. Se han desarrollado varias herramientas de simulación de red con enfoques muy diferentes y utilizadas en muchos campos de investigación (Idris et al., 2022). Durante el proceso de esta investigación se analizaron varias herramientas de simulación como lo es ns-3, aunque se decidió por el uso de dos llamados LoRaSim y FloRa.

7.3. SOFTWARE DE SIMULACIÓN

7.3.1. LORASIM

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

LoRaSim es un simulador de eventos discretos basado en Python y diseñado para analizar la escalabilidad de una red LoRa (Bor et al., 2016). LoRaSim permite el despliegue de nodos y gateways en una cuadrícula bidimensional o espacio aleatorio.

El modelo de canal en LoRaSim se implementa en función de la atenuación LogNormal en función de la distancia y a efectos de este trabajo se implemento en ese código el modelo de Hata.

Aunque LoRaSim es un simulador simple que proporciona bastante información en términos del rendimiento de la red, hay algunas funcionalidades que no implementa como principalmente las confirmaciones de mensajes. Por lo tanto, no se puede utilizar para investigar algunos aspectos del rendimiento de las redes, particularmente cuando los nodos cambian su factor de dispersión con retroalimentación de la puerta de enlace por los que no se puede utilizar para evaluar el mecanismo *Adaptive Data Rate* (ADR), que es esencial para optimizar el rendimiento de la red (Reynders et al., 2018).

LoRaSim ofrece la posibilidad de ejecutar redes con múltiples *gateways* y múltiples canales ajustando el SF y ajustando la potencia del nodo final en función de su distancia desde el gateway.

Para que LoRaSim funcione sin problemas, se instalaron las bibliotecas SimPy, matplotlib y NumPy. También ofrece una implementación de visualización de las de los resultados, pero sin interfaz gráfica.

LoRaSim es muy versátil y muchos investigadores lo han ampliado o mejorado para satisfacer sus necesidades y tal ha sido el caso de esta investigación donde se le realizaron modificaciones de los algoritmos de propagación.

7.3.2. FLORA

FLoRa es un entorno de simulación basado en el simulador OMNeT++ y en modulo INET que permite llevar a cabo simulaciones de extremo a extremo para redes LoRa. Permite la simulación completa de la red LoRa/LoRaWAN con sus componentes principales. FLoRa implementa la especificación LoRaWAN clase A con modo de transmisión no confirmado. A través del mecanismo ADR, el servidor de red y los nodos admiten la gestión dinámica de los parámetros de configuración. El mecanismo ADR controla los parámetros SF, BW y TP de los DE.

A diferencia de LoraSIM, FLoRa tiene una interfaz de usuario gráfica.

Además, FLoRa ofrece un modelo de capa física LoRa preciso y una simulación de extremo a extremo con una (o más) gateways. La comunicación entre los gateway(s) y el(los) servidor(es) de red es a través del Protocolo de Internet (IP). La capa física entre la(s) gateways y el servidor de red se puede realizar con los módulos de marco INET existentes.

7.4. METRICAS DE SIMULACIÓN

Se definieron las siguientes métricas de simulación:

7.4.1. RELACIÓN DE ENTREGA DE PAQUETES (PDR)

Relación de entrega de paquetes (PDR): se define como el número total de paquetes recibidos por el servidor de red dividido por el número total de paquetes enviados por los nodos finales. El PDR se puede calcular por nodo o para toda la red. Es una de las métricas de rendimiento más utilizadas en la literatura de redes de LoRaWAN. Para toda la red, esto se puede calcular como se muestra en la Ecuación 24.

$$PDR = \frac{\sum \text{Número de paquetes recibidos}}{\sum \text{Número de paquetes enviados}}$$

Ecuación 24. Tasa de entrega de paquetes (PDR)

7.4.2. NÚMERO DE COLISIONES

Se define como colisión al fenómeno que ocurre cuando dos o más dispositivos intentan transmitir un paquete simultáneamente, resultando en la posible pérdida de datos transmitidos de uno o los dos dispositivos. El concepto de colisión o cómo se detecta puede variar dependiendo de los criterios de cómo se define una colisión.

7.4.3. DISTRIBUCIÓN ESPACIAL DE LOS NODOS

La distribución espacial de nodos se refiere a una representación de las coordenadas X, Y de cada uno de los nodos en un eje de coordenadas donde se representa con diferentes colores alguno de los parámetros a ser visualizado.

7.4.4. HISTOGRAMA DE FACTORES DE DISPERSIÓN

El histograma de los factores representa la cantidad de sensores que utilizan cada uno de los factores de dispersión.

7.5. PARÁMETROS DE SIMULACIÓN

Topología de Red	Estrella de estrellas
Número de <i>gateways</i>	1
Frecuencia	1 canal – 916,8 MHz 2 canales – 916,8 - 917,0 MHz 4 canales – 916,8 - 917,0 - 917,2 - 917,2 MHz 8 canales – 916,8 - 917,0 - 917,2 - 917,2 - 917,4 - 917,6 - 917,8 - 918,0 MHz
Factor de dispersión	7 a 12
Ancho de banda	125 kHz
Potencia de transmisión	20 dBm
<i>Coding rate</i>	4/5
Tamaño de paquete (Carga Útil Phy)	14 bytes – 1 Byte de Aplicación 20 bytes – 6 Bytes de Aplicación 40 bytes – 27 Bytes de Aplicación
Número de dispositivos	10-800
Tiempo entre paquetes	30 segundos 2 minutos 1 hora
Tiempo de simulación	10000 segundos
Dimensiones del área	Urbano – 500 metros (radial) Rural – 5000 metros (radial)

7.6. RESULTADOS DE SIMULACIÓN

7.6.1. RESULTADOS DE SIMULACIÓN RURAL

7.6.1.1 TASA DE ENTREGA DE PAQUETES CONTRA CANTIDAD DE SENSORES Y NÚMERO DE CANALES

Esta simulación muestra el impacto del crecimiento de la cantidad de sensores sobre la tasa de entrega exitosa de paquetes, en este escenario se deja fijo el tamaño de la carga útil en 20 bytes y el intervalo de entrega de paquetes a 30 segundos por sensor.

En la Figura 64 se puede observar que para el caso de un canal, la tasa de entrega baja a menos del 50 % por sobre 250 sensores y continúa bajando a menos del 20 % para 800 sensores. Por otro lado, se puede ver claramente cómo el aumento de la cantidad de canales aumenta la tasa de entrega exitosa y, por lo tanto, permite tener más sensores en la misma área.

Simulaciones

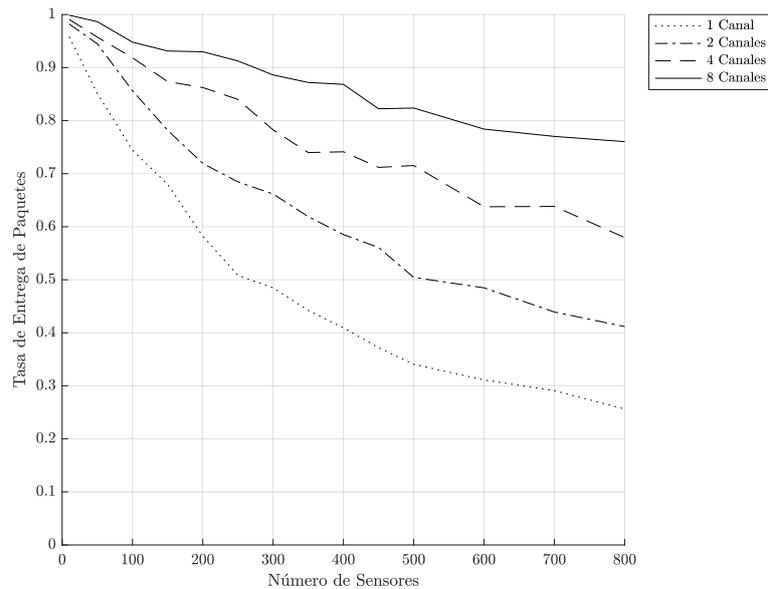


Figura 64. Simulación – Rural - Tasa de entrega de paquetes vs cantidad de sensores vs canales

7.6.1.2 TASA DE ENTREGA DE PAQUETES CONTRA CANTIDAD DE SENSORES Y TAMAÑO DE LA CARGA ÚTIL

Esta simulación muestra el impacto del crecimiento de la cantidad de sensores sobre la tasa de entrega exitosa de paquetes, en este escenario se deja fijo el número de canales en 8 y el intervalo de entrega de paquetes a 30 segundos por sensor.

En la Figura 65 se puede observar que para el caso de 14 bytes de carga útil física o sea 1 byte de carga útil de aplicación, la tasa de entrega baja a alrededor de 80 % en el nivel de 800 sensores y, por otro, se puede ver claramente como el aumento de la cantidad de tamaño de la carga útil la tasa de entrega exitosa baja y por lo tanto es posible tener menos sensores si se aumenta la misma.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

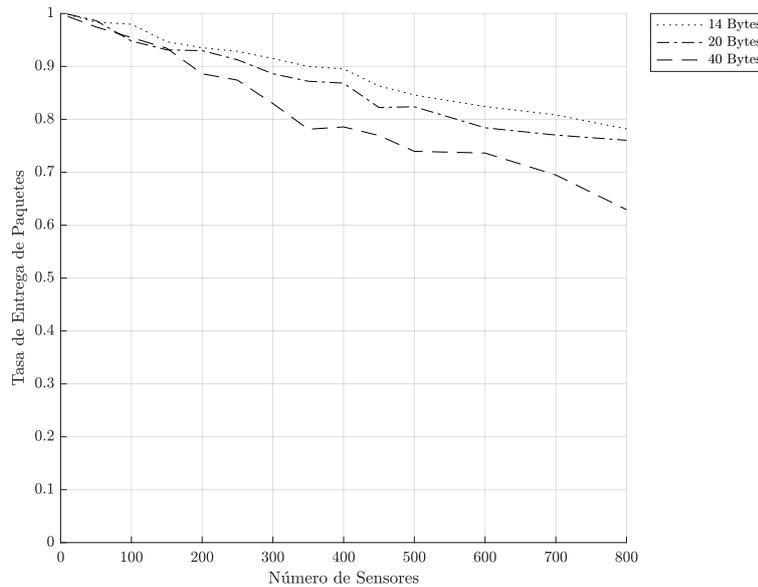


Figura 65. Simulación – Rural - Tasa de entrega de paquetes vs cantidad de sensores vs carga útil

7.6.1.3 TASA DE ENTREGA DE PAQUETES CONTRA CANTIDAD DE SENSORES E INTERVALO ENTRE PAQUETES

Esta simulación muestra el impacto del crecimiento de la cantidad de sensores sobre la tasa de entrega exitosa de paquetes, en este escenario se deja fijo el número de canales en ocho y el tamaño de la carga útil en 20 bytes por sensor.

En la Figura 66 se puede observar que, para el caso de 30 segundos de intervalo, la tasa de entrega baja a alrededor de 75 % en el nivel de 800 sensores. Por otro se puede ver claramente como el aumento del intervalo hace que la tasa de entrega exitosa suba significativamente a más del 99 % para 1 hora y por lo tanto es posible tener muchos más sensores si se aumenta dicho intervalo, esta es una consideración muy importante para no sobre-medir en los sensores dado que el impacto en la estabilidad de la red es significativo.

Simulaciones

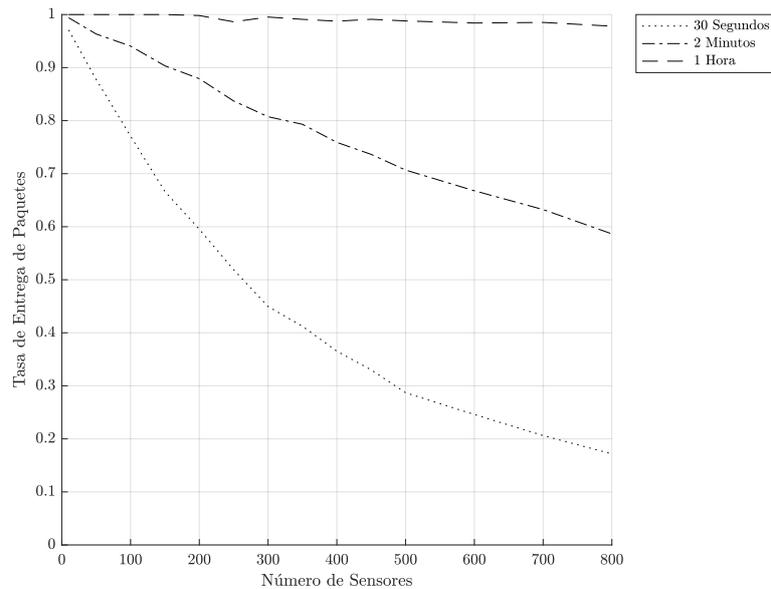


Figura 66. Simulación – Rural - Tasa de entrega de paquetes vs cantidad de sensores vs intervalo

7.6.1.4 COLISIONES CONTRA CANTIDAD DE SENSORES Y NÚMERO DE CANALES

Esta simulación muestra el impacto del crecimiento de la cantidad de sensores sobre la cantidad de colisiones en este escenario se deja fijo el tamaño de la carga útil en 20 bytes y el intervalo de entrega de paquetes a 30 segundos por sensor.

En la Figura 67 se puede observar que la cantidad de colisiones sube relativamente de manera proporcional a la cantidad de dispositivos para todos los casos, para el caso de menos canales la cantidad de colisiones es mayor debido a la mayor cantidad de transmisiones compartiendo el mismo medio pero que al agregar más canales la cantidad de baja.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

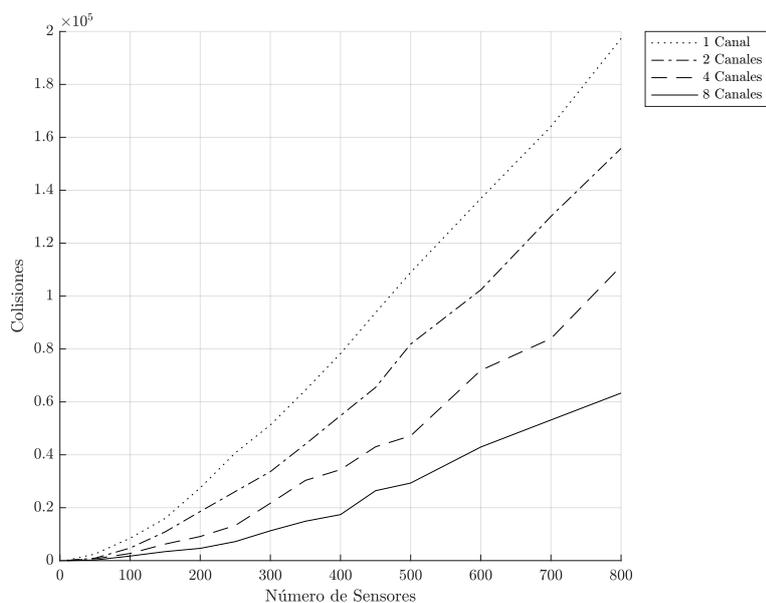


Figura 67. Simulación – Rural - Colisiones vs cantidad de sensores vs canales

7.6.1.5 DISTRIBUCION ESPACIAL DE SENSORES Y FACTORES DE DISPERSIÓN

La representación de la Figura 68 permite observar la distribución aleatoria de los sensores, para cada uno de los casos de prueba. Para este ejemplo se eligieron los casos de 50, 100, 200 y 400 sensores.

En los resultados se puede observar cómo la distribución alcanza distancias de 5000 metros con factores de dispersión no mayores a SF10 y valores de SF7 para distancias de 2000 metros lo cual se corresponde con los resultados experimentales en el campo y con los cálculos teóricos del modelo de Hata para campo abierto.

Simulaciones

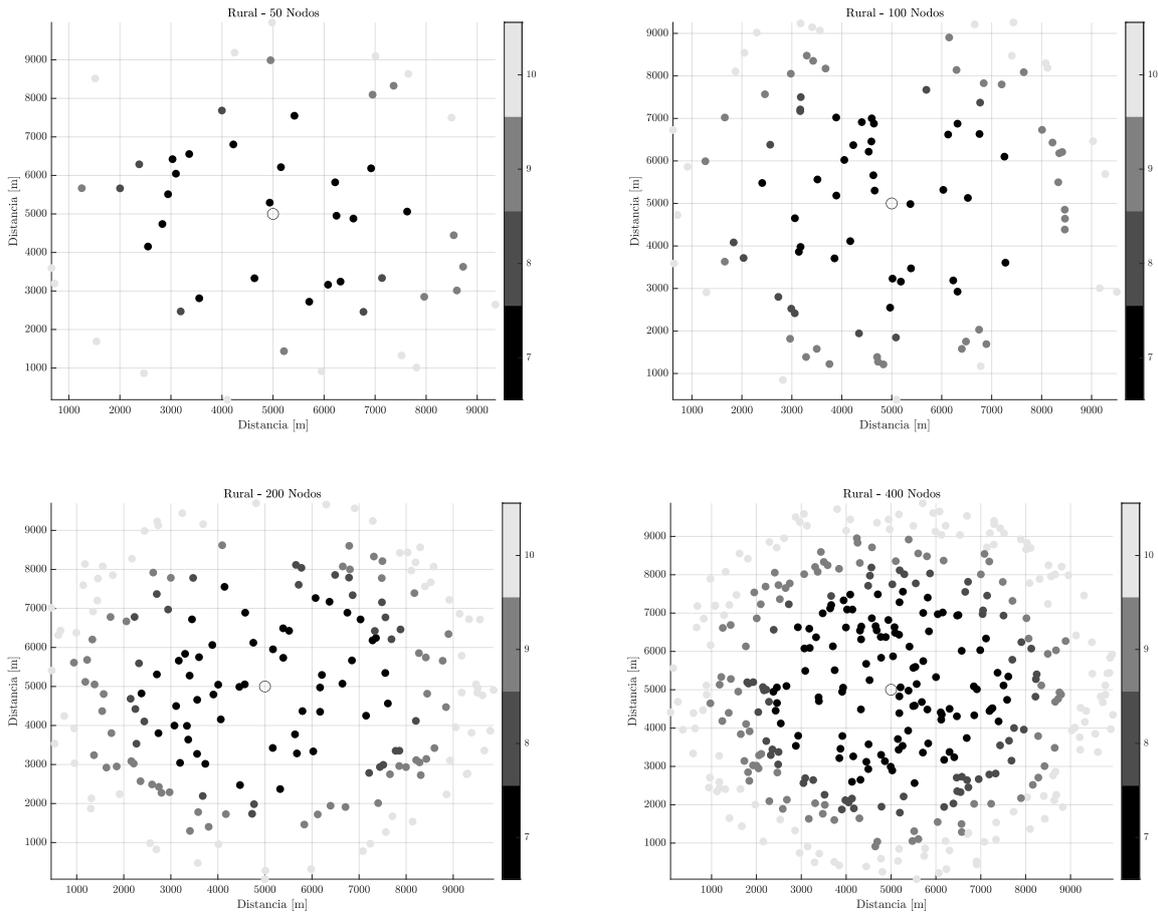


Figura 68. Simulación – Rural – Visualización de factor de dispersión en función de diferentes distancias al gateway para una distribución aleatoria de sensores

7.6.1.6 HISTOGRAMA DE DISTRIBUCION DE FACTORES DE DISPERSIÓN

La representación de la Figura 69 permite visualizar la cantidad total de sensores por cada factor de dispersión. Como se revisó anteriormente es muy importante conocer esto dado que cuánto más alto es el factor de dispersión, más alto es el tiempo de aire para la misma cantidad de información transmitida (Figura 15) y por lo tanto menor será la capacidad total del canal (Figura 20).

Asumiendo una distribución equiprobable de sensores transmitiendo la misma carga útil se puede estimar la capacidad máxima real de este canal basado en los máximos teóricos de la Tabla 9 y la Ecuación 25

$$Capacidad\ Máxima = \sum_{SF=7}^{12} \% \text{ dispositivos } (SF) * Capacidad\ Máxima (SF)$$

Ecuación 25. Capacidad de canal con factores mixtos

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

En este caso particular, el canal tendrá una capacidad máxima estimada de 2341 bits por segundo según lo calculado en la Tabla 28

SF	SF7	SF8	SF9	SF10	SF11	SF12	Total
Capacidad Máxima [bps]	4845	2738	1359	584	261	146	2290
Dispositivos [n]	116	67	81	136	0	0	400
Porcentaje [%]	0.28	0.16	0.22	0.34	0	0	1

Tabla 28. Cálculo de capacidad máxima de canal

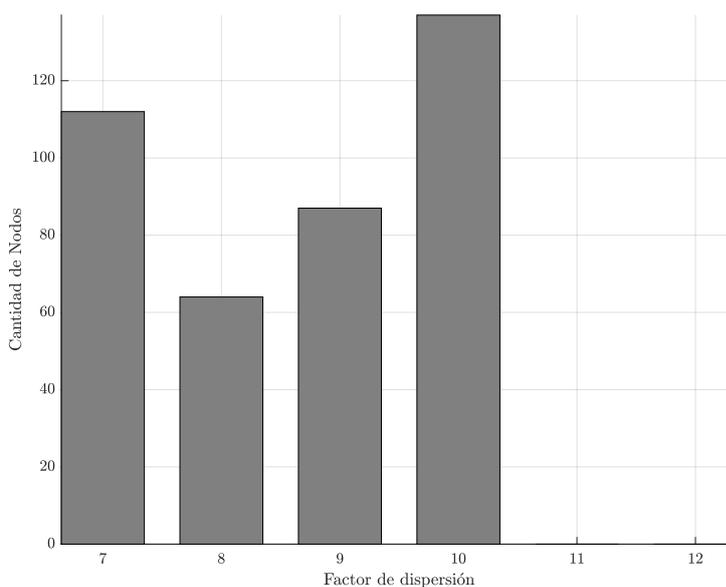


Figura 69. Simulación – Rural – Distribución de sensores vs. factor de dispersión

7.6.2. RESULTADOS DE SIMULACIÓN URBANA

7.6.2.1 TASA DE ENTREGA DE PAQUETES CONTRA CANTIDAD DE SENSORES Y NÚMERO DE CANALES

Esta simulación muestra el impacto del crecimiento de la cantidad de sensores sobre la tasa de entrega exitosa de paquetes, en este escenario se deja fijo el tamaño de la carga útil en 20 bytes y el intervalo de entrega de paquetes a 30 segundos por sensor.

En la Figura 70 se puede observar que para el caso de un canal, la tasa de entrega baja a menos del 50 % por sobre 400 sensores y continúa bajando a al 30 % para 800 sensores. Por otro, se puede

Simulaciones

ver claramente cómo el aumento de la cantidad de canales aumenta la tasa de entrega exitosa y por lo tanto permite tener más sensores en la misma área.

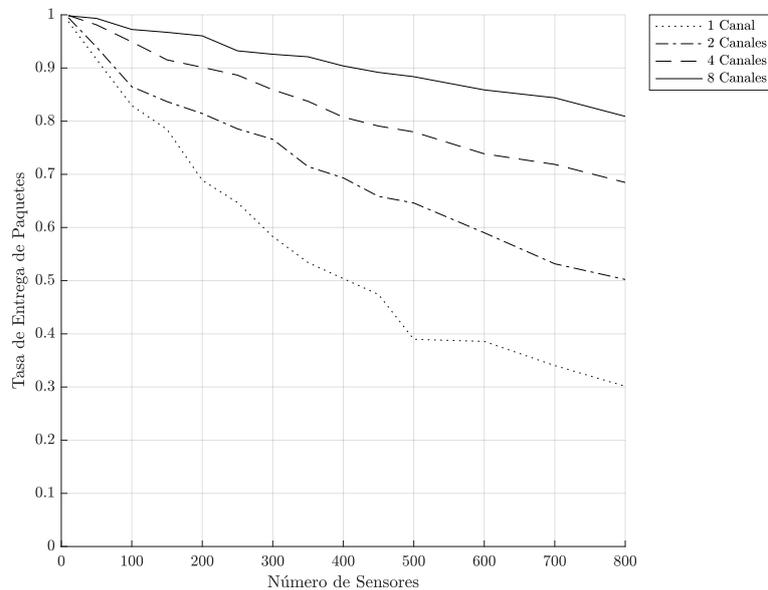


Figura 70. Simulación – Urbano - Tasa de entrega de paquetes vs. cantidad de sensores vs canales

7.6.2.2 TASA DE ENTREGA DE PAQUETES CONTRA CANTIDAD DE SENSORES Y TAMAÑO DE LA CARGA ÚTIL

Esta simulación muestra el impacto del crecimiento de la cantidad de sensores sobre la tasa de entrega exitosa de paquetes, en este escenario se deja fijo el número de canales en ocho y el intervalo de entrega de paquetes a 30 segundos por sensor.

En la Figura 71 se puede observar que para el caso de 14 bytes de carga útil física o sea 1 byte de carga útil de aplicación, la tasa de entrega baja a alrededor de 87 % en el nivel de 800 sensores. Por otro, se puede ver claramente cómo el aumento de la cantidad de tamaño de la carga útil la tasa de entrega exitosa baja y por lo tanto es posible tener menos sensores si se aumenta la misma.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

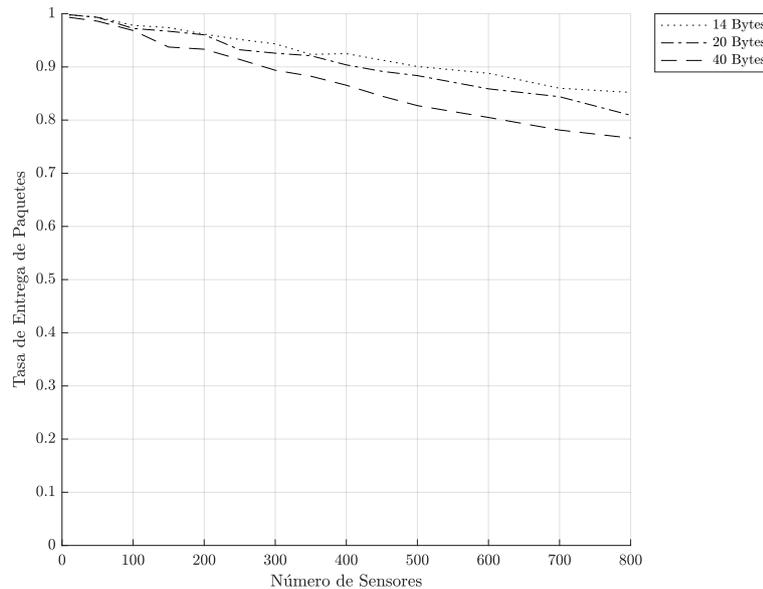


Figura 71. Simulación – Urbano - Tasa de entrega de paquetes vs. cantidad de sensores vs carga útil

7.6.2.3 TASA DE ENTREGA DE PAQUETES CONTRA CANTIDAD DE SENSORES E INTERVALO ENTRE PAQUETES

Esta simulación muestra el impacto del crecimiento de la cantidad de sensores sobre la tasa de entrega exitosa de paquetes, en este escenario se deja fijo el número de canales en uno y el tamaño de la carga útil en 20 bytes por sensor.

En la Figura 72 se puede observar que, para el caso de 30 segundos de intervalo, la tasa de entrega baja a alrededor de 30 % en el nivel de 800 sensores. Por otro se puede ver claramente como el aumento del intervalo hace que la tasa de entrega exitosa suba significativamente, más del 99 % para 1 hora, y por lo tanto es posible tener muchos más sensores si se aumenta dicho intervalo, esta es una consideración muy importante para no sobre-medir en los sensores dado que el impacto en la estabilidad de la red es significativo.

Simulaciones

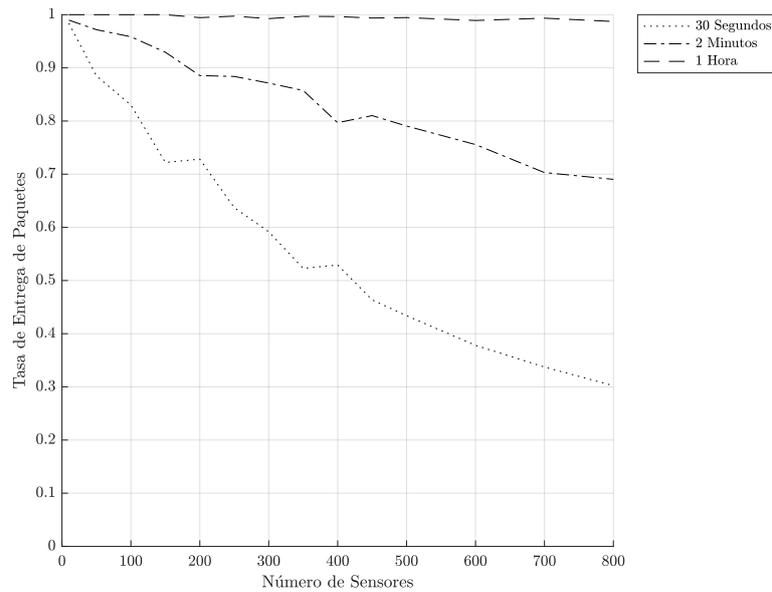


Figura 72. Simulación – Urbano - Tasa de entrega de paquetes vs. cantidad de sensores vs intervalo

7.6.2.4 COLISIONES CONTRA CANTIDAD DE SENSORES Y NÚMERO DE CANALES

Esta simulación muestra el impacto del crecimiento de la cantidad de sensores sobre la cantidad de colisiones en este escenario se deja fijo el tamaño de la carga útil en 20 bytes y el intervalo de entrega de paquetes a 30 segundos por sensor.

En la Figura 73 se puede observar que la cantidad de colisiones sube relativamente de manera proporcional a la cantidad de dispositivos para todos los casos, para el caso de menos canales la cantidad de colisiones es mayor debido a la mayor cantidad de transmisiones compartiendo el mismo medio pero que al agregar más canales la cantidad de baja.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

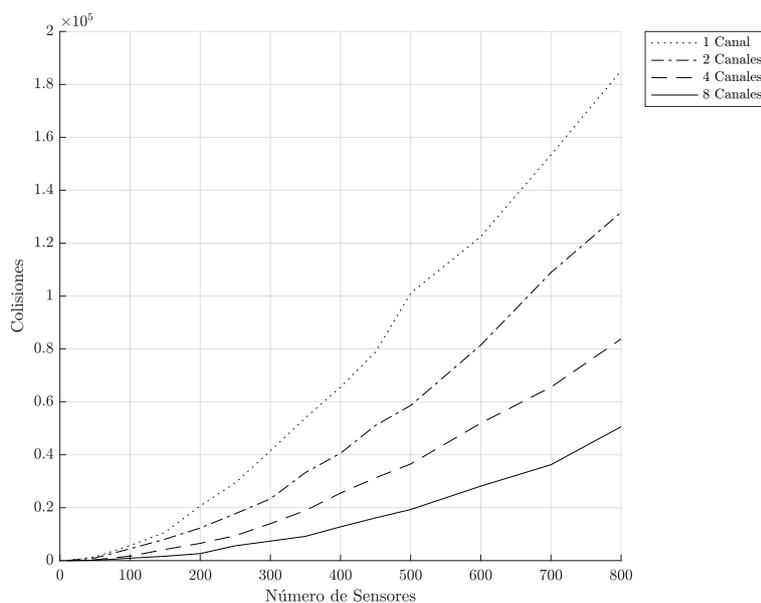


Figura 73. Simulación – Urbano - Colisiones vs cantidad de sensores vs canales

7.6.2.5 DISTRIBUCIÓN ESPACIAL DE SENSORES Y FACTORES DE DISPERSIÓN

La representación de la Figura 74 permite observar la distribución aleatoria de los sensores, para cada uno de los casos de prueba. Para este ejemplo se eligieron los casos de 50, 100, 200 y 400 sensores.

En los resultados se puede observar como la distribución alcanza distancias de 500 metros con factores de dispersión SF12 y valores de SF7 para distancias de 200 metros lo cual se corresponde con los resultados experimentales en el campo y con los cálculos teóricos de los modelos de Hata Urbano y del de Villardi

Simulaciones

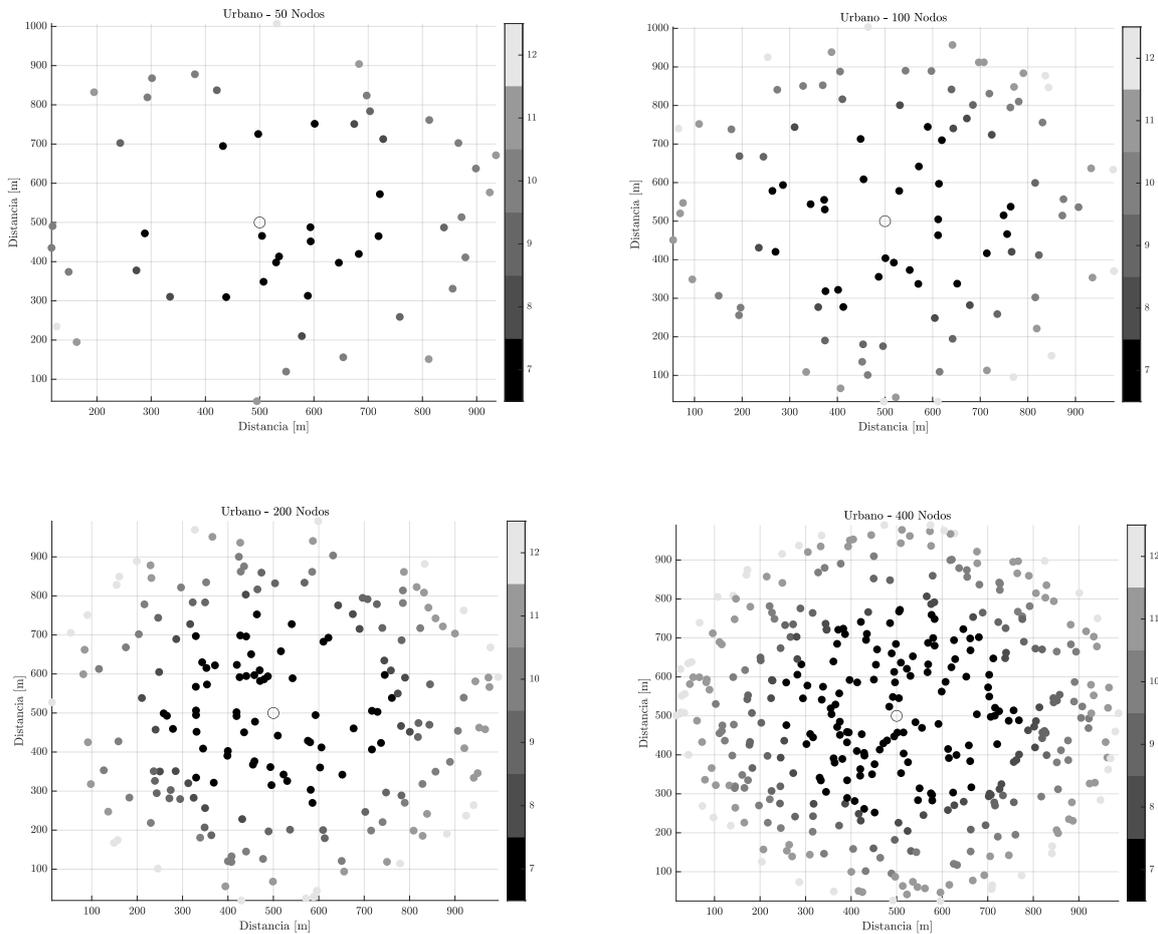


Figura 74. Simulación – Urbano – Distribución de sensores vs factor de dispersión

7.6.2.6 HISTOGRAMA DE DISTRIBUCIÓN DE FACTORES DE DISPERSIÓN

La representación de la Figura 75 permite visualizar la cantidad total de sensores por cada factor de dispersión. Como se revisó anteriormente es muy importante conocer esto dado que cuanto más alto es el factor de dispersión, más alto es el tiempo de aire para la misma cantidad de información transmitida (Figura 15) y por lo tanto menor será la capacidad total del canal (Figura 20).

Suponiendo una distribución equiprobable de sensores transmitiendo la misma carga útil se puede estimar la capacidad máxima real de este canal basado en los máximos teóricos de la Tabla 9 y la Ecuación 25

En este caso particular el canal tendrá una capacidad máxima estimada de 1871 bits por segundo según lo calculado en la Tabla 28

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

SF	SF7	SF8	SF9	SF10	SF11	SF12	Total
Capacidad Máxima [bps]	4845	2738	1359	584	261	146	2078
Dispositivos [n]	121	41	50	64	82	42	400
Porcentaje [%]	0.30	0.10	0.13	0.16	0.21	0.11	1

Tabla 29. Cálculo de capacidad máxima de canal

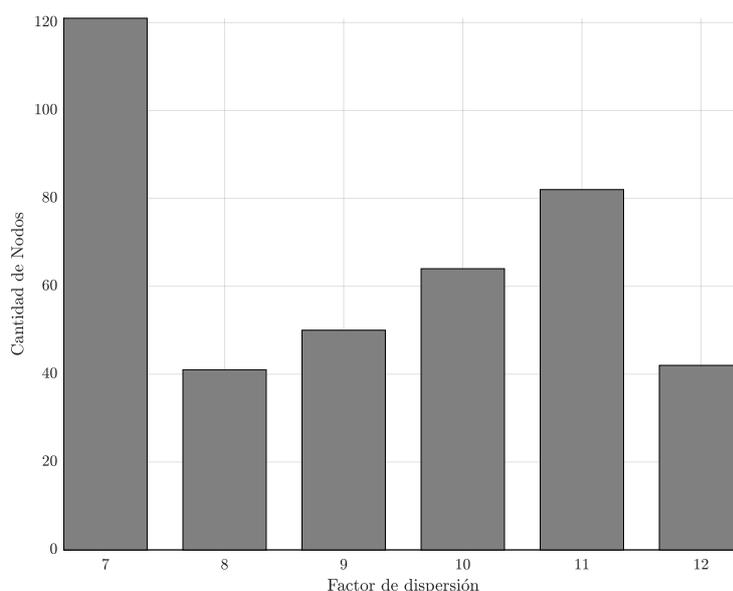


Figura 75. Simulación – Urbano – Distribución de sensores vs factor de dispersión

7.7. CONCLUSIONES

Este capítulo permito llegar a varias conclusiones muy importantes sobre todo principalmente a la verificación de la hipótesis principal de este trabajo: a medida que aumenta la cantidad de dispositivos se ve a afectada la probabilidad de entrega exitosa de los paquete con información debido a las colisiones por potenciales múltiples sensores transmitiendo al mismo tiempo, esto implica que cuantos más canales se encuentren disponibles mejor será la eficacia de la red en transmitir esos mensajes. Por otro lado, las curvas de tasa de transmisión exitosa permiten definir un tope máximo aceptable de pérdida de paquetes según la criticidad de los datos transmitidos para así poder definir la cantidad máxima de sensores a ser soportados. Por ejemplo, en un escenario rural, con un gateway de ocho canales se puede soportar 800 sensores por área de servicio si se

Simulaciones

tolera una pérdida de paquetes máxima del 20%. En segundo término, se validó la fuerte influencia del ciclo de actividad de los sensores en la tasa de entrega exitosa de paquetes, y como aplicaciones intensivas en datos afectan severamente la cantidad máxima de sensores. Por ejemplo, en un escenario urbano si toleramos un máximo de 20% de pérdida de paquetes, para aplicaciones con un ciclo de actividad de 30 segundos se podrían soportar 100 sensores, mientras que si el ciclo baja a dos minutos son 500 y si es de una hora está en el rango de los miles.

Por último, se comprobó como la dispersión espacial de dispositivos hace que a medida que sube la distancia se utilicen factores de dispersión más altos y como eso afecta la capacidad total de la red. Es importante mencionar que conocer esto permite tomar decisiones de limitar el alcance máximo para utilizar solo factores de dispersión más bajos y así mantener la capacidad total del canal en valores más altos.

8. ANÁLISIS DE APLICACIÓN EN REDES EXISTENTES

8.1. RESEÑA DEL CAPÍTULO

En este último capítulo se realiza un análisis de aplicación de una red LoRaWAN utilizando componentes de una red telecomunicaciones preexistente. En particular, se utiliza como ejemplo una red del tipo Híbrido Fibra Coaxial donde se utiliza la información obtenida en los capítulos 6 y 7 para analizar la factibilidad operativa de esa red con respecto al área geográfica de aplicación y con datos de densidad poblacional se validó el modelo.

Finalmente se analizan consideraciones de alimentación de energía y de conectividad a internet.

8.2. REDES HFC

Con el objetivo de aplicar en una red operativa lo investigado en los capítulos anteriores, se obtuvieron los diagramas de planta externa de diferentes nodos tipos de redes Híbridas Fibra Coaxial (HFC). De estos planos se relevó la cobertura geográfica de cada uno de los nodos y se llevó a cabo un análisis considerando ubicar un gateway en donde el nodo se encuentra actualmente

8.2.1. REDES NODO+0

Las redes HFC con 0 amplificadores en cascada suelen tener cables coaxiales cortos que se expanden en promedio no más de 200 metros desde la ubicación del nodo e incluso menos de 150 metros en ciudades densas como el ejemplo en la Figura 76

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

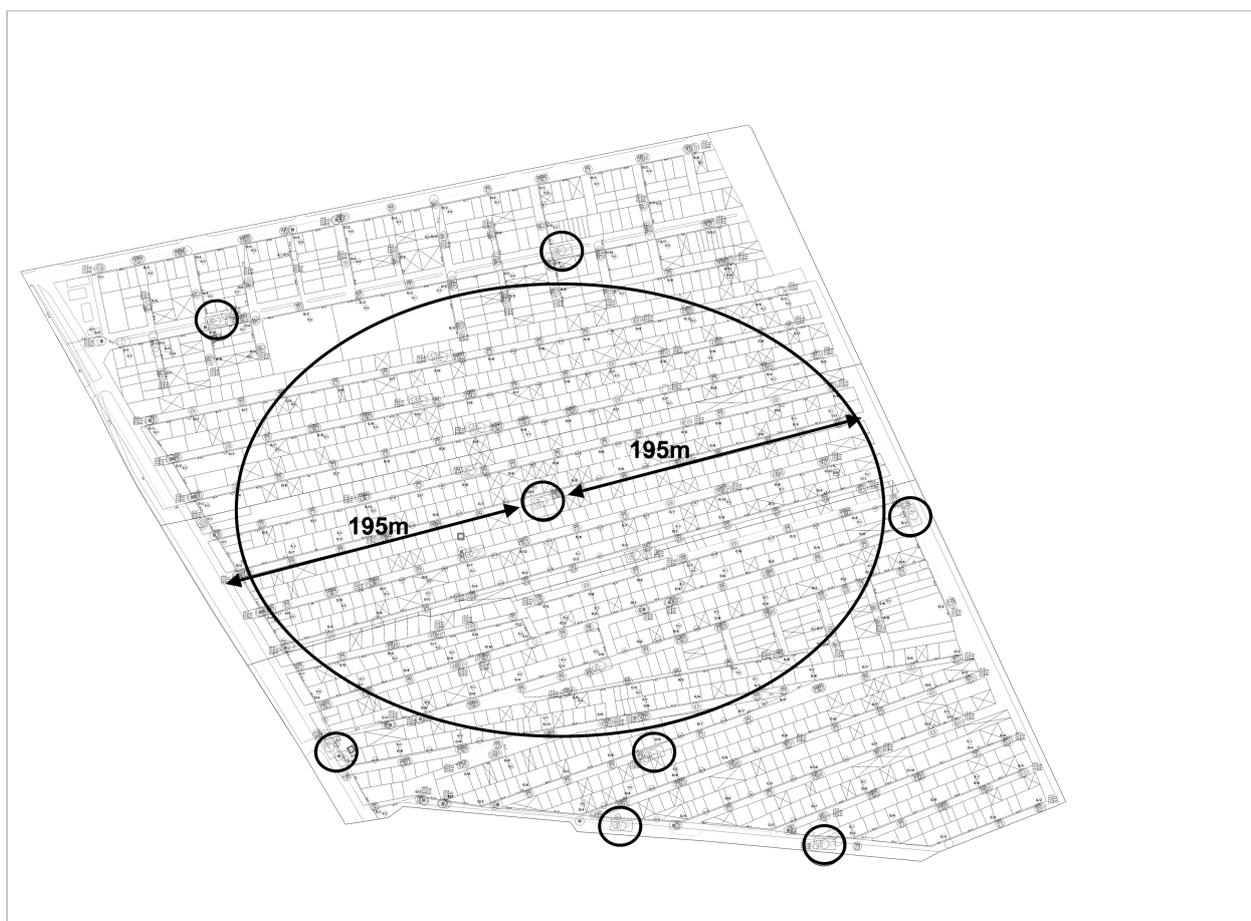


Figura 76. Red de distribución Nodo+0 de ejemplo

Ese escenario se encuentra cubierto con la configuración LoRaWAN presentada en la sección 4.5.1, si la puerta de enlace LoRaWAN se ubica en el nodo de fibra, en esta configuración se requiere una distancia máxima de 253 m para dar servicio a las estaciones que utilizan la capacidad máxima con un factor de dispersión $SF=7$.

En este escenario, no todos los nodos de fibra necesitan tener una puerta de enlace LoRa ya que la distancia no supera el alcance máximo, aunque si fuera beneficio dado que permitiría tener diversidad de *gateways* a los nodos.

8.2.2. REDES HFC NODO+1 O NODO+2

Análisis de aplicación en redes existentes

Las redes HFC con 1 amplificador en cascada suelen tener cables coaxiales medianos que se expanden en promedio 500 metros desde la ubicación del nodo en ciudades no tan densas con casas como en el ejemplo de la Figura 77.

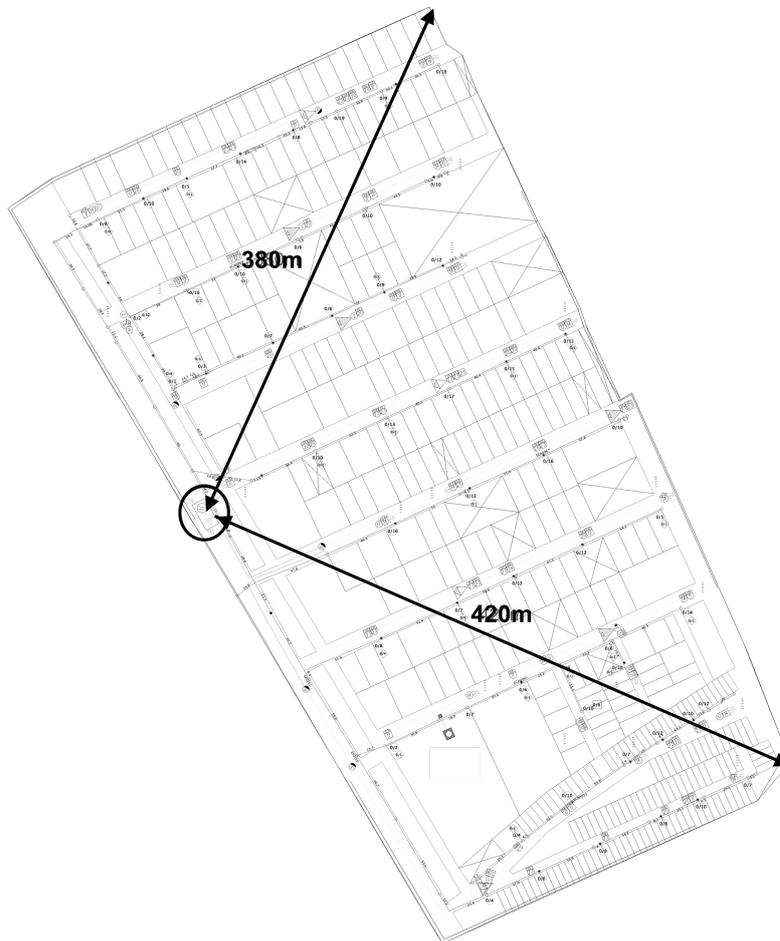


Figura 77. Red de distribución Nodo+1 de ejemplo

Ese escenario se encuentra cubierto con la configuración LoRaWAN presentada en la sección 4.5.2, si la puerta de enlace LoRaWAN se ubica en el nodo de fibra, en esta configuración se requiere una distancia máxima de 764 m para dar servicio a las estaciones que utilizan la capacidad máxima con un factor de dispersión SF=7.

En este escenario, todos los nodos de fibra deben tener una puerta de enlace LoRa, ya que la distancia está bien alineada con el alcance inalámbrico.

8.3. DENSIDAD Y RENDIMIENTO DE LA RED

Teniendo en cuenta las diferentes aplicaciones de IoT, es muy importante poder modelar los patrones de tráfico y la cantidad de dispositivos según la densidad del área. En este estudio se

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

considerará el uso público y algunos casos de uso residencial donde el tráfico de datos no es intensivo. La densidad de dispositivos por áreas suburbanas y urbanas se tomó de (Rui Huang et al., 2011) aunque también se verificó que la densidad es similar en la Ciudad Autónoma de Buenos Aires basado en los datos obtenidos en el portal de estadísticas del gobierno de la ciudad.

A efectos de poder calcular la cantidad de sensores soportados se presenta la Tabla 30, en la cual se usa como métrica el tiempo de aire de una transmisión, en los ejemplos de la tabla se supone un canal de 125 kHz y se considera el tiempo máximo para una carga máxima de 242 bytes y el mínimo para una carga 1 byte. Este tiempo proviene del cálculo realizado en la sección y presentado en la figura.

SF	Carga útil máxima PHY [bytes]	Carga útil máxima MAC [bytes]	Carga útil máxima FRM [bytes]	TdA máximo [s]	TdA mínimo [s]
7	255	250	242	0.400	0.046
8	255	250	242	0.707	0.082
9	127	122	114	0.677	0.165
10	64	59	51	0.698	0.289
11	64	59	51	1.561	0.659
12	64	59	51	2.793	1.155

Tabla 30. Tiempo de aire en función de la carga útil

Continuando con el análisis el siguiente factor a considerar es el intervalo de transmisión del sensor, naturalmente un intervalo más bajo de transmisión resultará en ocupación más frecuente del canal y por lo tanto dará lugar a una menor cantidad de dispositivos. A efectos de poder homogeneizar valores de diferentes intervalos de transmisión y de tamaño de carga útil se define una métrica llama ciclo de actividad, la cual relaciona el tiempo de aire del paquete con el intervalo en que esos paquetes son transmitidos según la

$$CA = \frac{TdA [s]}{Intervalo [s]}$$

Ecuación 26. Fórmula del ciclo de actividad

Análisis de aplicación en redes existentes

En la Tabla 31 se presenta un ejemplo para una carga útil de 20 bytes a intervalos de 30 segundos para los diferentes factores de dispersión, claramente se ve que a medida que el factor de dispersión es más alto sube el ciclo de actividad. Un cálculo relevante es poder conocer la cantidad máxima de sensores del mismo tipo que soporta un canal determinado en condiciones ideales de agendamiento de transmisiones, lo que se puede hacer llevando al 100% el ciclo de actividad según la Ecuación 27. Ahora bien, como se analizó en el capítulo 4, la cantidad máxima de ocupación de una canal ALOHA está limitada según la Ecuación 28

$$N_{Max}[Ideal] = \frac{1}{CA}$$

Ecuación 27. Fórmula de la cantidad máxima de sensores en condiciones ideales

$$N_{Max}[ALOHA] = \frac{1}{CA} \frac{1}{2e}$$

Ecuación 28. Fórmula de la cantidad máxima de sensores en un canal ALOHA

	Intervalo [s]	30	Tamaño [bytes]	20	Canales	1	
SF	Carga útil máxima PHY [bytes]	Carga útil máxima MAC [bytes]	Carga útil máxima FRM [bytes]	TdA	Ciclo de Actividad	Cantidad Ideal máxima de Sensores	Cantidad máxima de Sensores ALOHA
7	33	28	20	0.072	0.2398%	417	77
8	33	28	20	0.134	0.4454%	224	41
9	33	28	20	0.247	0.8226%	121	22
10	33	28	20	0.453	1.5087%	66	12
11	33	28	20	0.987	3.2905%	30	6
12	33	28	20	1.810	6.0348%	16	3

Tabla 31. Ciclo de actividad y cantidad de dispositivos en función de la carga útil

Diferentes tipos de sensores realizan diferentes tipos de mediciones que requieren variadas cargas útiles a variados intervalos, por otro lado, típicamente los gateway tienen 8 canales por lo que la cantidad máxima de sensores se multiplica. En la Tabla 32 se muestra una lista de sensores típicos junto con su carga útil e intervalos para así realizar los cálculos de Ciclo de Actividad de cada uno de ellos.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Sensor	Intervalo [s]	Tamaño [bytes]	ToA [s]	CA	Canales	Cantidad Ideal Máxima de Sensores	Cantidad de Sensores ALOHA
Señales de Transito	30	1	0.046	0.1546%	1	647	119
					8	5176	952
Semáforos	60	1	0.046	0.0773%	1	1294	238
					8	10352	1904
Sensores de Trafico	60	1	0.046	0.0773%	1	1294	238
					8	10352	1904
Posnet en un Negocio	1800	24	0.082	0.0046%	1	21904	4029
					8	175232	32232
Medidores inteligentes	9000	20	0.072	0.0008%	1	125111	23013
					8	1000888	184103
Sistema de seguridad para el hogar	600	20	0.072	0.0120%	1	8340	1534
					8	66720	12272
Electrodomésticos	60	8	0.057	0.0943%	1	1060	195
					8	8480	1560

Tabla 32. Ciclo de actividad y para diferentes tipos de dispositivos

Para facilitar los cálculos en el caso de los hogares se puede resumir todos los ciclos de actividad para obtener un ciclo único por hogar según se refleja en la tabla Tabla 33.

Sensor	Intervalo [s]	Tamaño [bytes]	ToA [s]	CA	Cantidad por hogar	Ciclo de actividad Total
Medidores inteligentes	9000	20	0.072	0.0008%	3	0.20%
Sistema de seguridad para el hogar	600	20	0.072	0.0120%	1	
electrodomésticos	60	8	0.057	0.0943%	2	

Tabla 33. Ciclo de actividad agregado para casas

En la tabla 34 se presentan valores típicos de densidad de hogares, comercios y dispositivos urbanos, estos fueron obtenidos de (Rui Huang et al., 2011), en un relevamiento para New York y Washington, se contrastaron los valores de densidad urbana con los valores para diferentes barrios

Análisis de aplicación en redes existentes

de la Ciudad autónoma de Buenos Aires obtenidos del portal de estadísticas, presentando valores equivalentes.

	Urbano Denso [cantidad/km2]	Semiurbano [cantidad/km2]
Casas	3844.0	1479.2
Supermercados	209.5	23.1
Negocios	2200.0	349.9
Señales de Tráfico	316.5	94.3
Semáforos	15.0	11.4
Sensores de Tráfico	15.0	11.4

Tabla 34. Densidad de sensores por escenario

Por último, se combinaron todos los cálculos para dar como resultado la Tabla 35, donde combinando la Ecuación 29 y la Ecuación 30 se puede comparar el valor de ciclo de actividad esperado versus el ofrecido por un gateway de ocho canales que se sitúa en $8/2e = 1,47$.

En el primero de los casos se ve con un solo gateway se puede cubrir la necesidad de ancho de banda de los sensores dado que estos requieren solo el 1,059 del ciclo de actividad que es menor que el ofrecido por el gateway de 1,47.

En el segundo caso el ciclo de actividad requerido es de 2.50 lo que excede el ofrecido por un gateway, esto no es problema dado que se pueden usar dos gateway en la misma área o usar un gateway de 16 canales.

$$CA_{Total} = \sum_1^n CA_{Sensor_n}$$

Ecuación 29. Ciclo de actividad total de un grupo de sensores

$$CA_{Max}[n_{canales}]_{ALOHA} = \frac{n_{canales}}{2e}$$

Ecuación 30. Ciclo de actividad máximo en función de número de canales

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

	CA	Urbano Denso		Semiurbano	
Distancia máxima [m]		200	CA	500	CA
Área [km ²]		0.1257		0.7854	
Casas	0.203%	483.05	98.092%	1161.78	235.919%
Supermercados	0.005%	26.32	0.120%	18.16	0.083%
Negocios	0.005%	276.46	1.262%	274.80	1.255%
Señales de Tráfico	0.155%	39.77	6.147%	74.09	11.451%
Semáforos	0.077%	1.89	0.146%	8.98	0.694%
Sensores de Tráfico	0.077%	1.89	0.146%	8.98	0.694%
		Total	105.91%	Total	250.09%
	CA Máximo de un gateway de ocho canales en ALOHA		147.15%		147.15%
	Numero de <i>gateways</i> requeridos por área de servicio		1		2

Tabla 35. Cálculo de cantidad de gateways de 8 canales según escenario

8.4. CONECTIVIDAD DE ALIMENTACIÓN Y BACKHAUL

8.4.1. ALIMENTACIÓN DE ENERGÍA

La mayoría de las puertas de enlace LoRaWAN independientes de ocho canales del mercado tienen un consumo de energía inferior a cinco watts, incluso con *backhaul* 3G/LTE y GPS integrado. El uso de un módem de cable incorporado o una conexión Ethernet al nodo de fibra en lugar de una red de retorno inalámbrica puede promediar un consumo de energía de 5W. Esto está muy por debajo del consumo de energía total de un nodo de fibra HFC tradicional, que varía de 70 a 120 W y de 140 a 160 W para un nodo habilitado para acceso distribuido.

La integración de la puerta de enlace LoRaWAN en el nodo de fibra proporciona beneficios significativos en el área de alimentación, ya que la puerta de enlace se puede alimentar con la fuente de alimentación regulada del nodo de fibra.

8.4.2. CONECTIVIDAD DE RED

En cualquier red LoRaWAN (Figura 11) se requiere una conexión de *backhaul* para conectar los dispositivos de puerta de enlace a los servidores de aplicaciones. En un escenario basado en una

red HFC, este *backhaul* puede ser proporcionado por un módem de cable incorporado o una conexión Ethernet directa en el nodo de fibra. Este enfoque proporciona un gran beneficio para la confiabilidad de la red de retorno, ya que requiere recursos insignificantes de la red DOCSIS. Como se analizó en los capítulos anteriores, una puerta de enlace de ocho canales puede requerir un máximo de 13,5 kbps de ancho de banda. Comparado con los cientos de megabits disponibles para los módems DOCSIS o GPON, esto es totalmente insignificante. Dado esto, se puede usar flujos de servicio garantizó de entrega de alta prioridad en tiempo real en ese módem para asegurar la entrega inmediata de la información.

8.5. CONCLUSIONES

Este capítulo permite concluir que se pueden utilizar redes ya desplegadas de banda ancha para dar soporte a nuevas redes LoRaWAN y que la ubicación de los puntos de acceso de energía y conectividad son adecuados para el despliegue de *gateways* en zonas urbanas y suburbanas y que la cobertura ofrecida permitiría satisfacer los requerimientos típicos de sensores para dichos escenarios.

9. CONCLUSIONES Y TRABAJOS FUTUROS

9.1. CONCLUSIONES

En el capítulo 2, se pudo concluir que es de particular interés el análisis de las redes del tipo no licenciado, dado que al operar en un espectro que no requiere previa asignación de este permite mucha más flexibilidad para el despliegue de nuevas aplicaciones.

En el capítulo 3, se pudo mencionar como principal conclusión que el protocolo LoRaWAN demuestra ser muy robusto antes las interferencias, principalmente a costa de sacrificar tasa de transferencia en pos de mayor radio de cobertura y tolerancia al ruido, y por esa razón es de particular interés un análisis detallado de la capacidad de transferencia en un área determinada dado que los márgenes son bastante ajustados.

En el capítulo 4, se pueden extraer varias conclusiones, primero que la naturaleza ALOHA del protocolo LoRaWAN impacta severamente en la capacidad en bits por segundo de cada canal, por otro lado, el comportamiento del protocolo está fuertemente influenciado por el medio donde esta aplicado, es decir se registra una variación significativa de escenarios rurales a urbanos en distancia de cobertura y capacidad. Esto hace necesario la utilización de gateways de canales múltiples para poder expandir la capacidad. Por último, se ve claramente como la restricción de espectro actual en la República Argentina permite solamente el uso de 8x4 canales bidireccionales, dado que la banda alta de la asignación se encuentra compartida con los canales de bajada, en comparación con la disponibilidad de 64 canales en el plan de frecuencias US915.

En el capítulo 5, se pudo concluir que la variedad de herramientas de software libre disponibles, junto con los desarrollos de hardware de diseño abierto permiten la implementación completa a muy bajo costo tanto de una suite de pruebas de una red LoRaWAN, como también de una red productiva. Por otro lado, la utilización de *software defined radio* como analizador de espectro y de protocolos, ofrece un gran avance en términos de agilidad comparado con herramientas dedicadas tradicionales.

En el capítulo 6, se permitió llegar a las siguientes conclusiones: Los modelos de propagación de Hakamura-Hata y Villardi aproximan de manera muy precisa el comportamiento de una red LoRaWAN en las condiciones de los casos de estudio tanto rural como urbano propuestos. En condiciones urbanas fue muy difícil alcanzar los 500 metros de cobertura sin una pérdida de paquetes considerable algo que se correlaciona con los análisis teóricos realizados en el capítulo 4. En el escenario rural la cobertura fue mucho más amplia y se lograron distancias mayores de 2000 metros sin una pérdida de datos significativa. Se comprobó que la orientación de las antenas

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

tiene influencia en la cobertura y se vuelve más relevantes cuando las condiciones de señal se vuelven marginales.

Por último, se verificó que la activación con el modelo OTAA provee un mecanismo mucho más sencillo y estable de operación en una red productiva con condiciones variables, mientras que el ABP es mucho más útil para pruebas de laboratorio, pero mucho menos escalable en grandes despliegues.

En el capítulo 7, se llegó a varias conclusiones muy importantes sobre todo principalmente a la verificación de la hipótesis principal de este trabajo: a medida que aumenta la cantidad de dispositivos se ve afectada la probabilidad de entrega exitosa de los paquetes con información debido a las colisiones por potenciales múltiples sensores transmitiendo al mismo tiempo, esto implica que cuantos más canales se encuentren disponibles mejor será la eficacia de la red en transmitir esos mensajes. Por otro lado, las curvas de tasa de transmisión exitosa permiten definir un tope máximo aceptable de pérdida de paquetes según la criticidad de los datos transmitidos para así poder definir la cantidad máxima de sensores a ser soportados. Por ejemplo, en un escenario rural, con un gateway de ocho canales se puede soportar 800 sensores por área de servicio si se tolera una pérdida de paquetes máxima del 20%. En segundo término, se validó la fuerte influencia del ciclo de actividad de los sensores en la tasa de entrega exitosa de paquetes, y como aplicaciones intensivas en datos afectan severamente la cantidad máxima de sensores. Por ejemplo, en un escenario urbano si toleramos un máximo de 20% de pérdida de paquetes, para aplicaciones con un ciclo de actividad de 30 segundos se podrían soportar 100 sensores, mientras que si el ciclo baja a dos minutos son 500 y si es de una hora está en el rango de los miles.

Por último, se comprobó como la dispersión espacial de dispositivos hacer que a medida que sube la distancia se utilicen factores de dispersión más altos y como eso afecta la capacidad total de la red. Es importante mencionar que conocer esto permite tomar decisiones de limitar el alcance máximo para utilizar solo factores de dispersión más bajos y así mantener la capacidad total del canal en valores más altos.

En el capítulo 8, se concluye que se pueden utilizar redes ya desplegadas de banda ancha para dar soporte a nuevas redes LoRaWAN y que la ubicación de los puntos de acceso de energía y conectividad son adecuados para el despliegue de *gateways* en zonas urbanas y suburbanas y que

Conclusiones y trabajos futuros

la cobertura ofrecida permitiría satisfacer los requerimientos típicos de sensores para dichos escenarios.

La suma de los trabajos de los diferentes capítulos permitió probar las dos hipótesis principales formuladas al principio de este trabajo, donde se comprobó el efecto no solo del crecimiento de la cantidad de dispositivos sino también de las diferentes aplicaciones en los mismos, por otro lado las mediciones de campo permitieron validar la influencia de la distribución demográfica en la operación de una red validando la mucho mayor cobertura posible en áreas rurales (5000 metros) con respecto a áreas suburbanas y urbanas (400 metros).

Como conclusión general del trabajo se puede decir que se cumplió con el objetivo principal que era evaluar los factores influyen en mayor medida en la operación viable de una red de internet de las cosas en el espectro de radiofrecuencia no licenciado de la República Argentina, siendo un resumen de estos el área geográfica/demográfica donde la red es desplegada, la cantidad de dispositivos en esa área y también la naturaleza de la aplicación para la que son utilizados esos dispositivos, principalmente en función de su ciclo de actividad y de cantidad de datos transmitidos por medición.

A su vez, los objetivos específicos planteados, fueron alcanzados. Esto permitió tener mayor entendimiento de los parámetros de operación que influyen, pero también en dar a conocer el potencial de estas y de las limitaciones que tiene la asignación actual de espectro con respecto a la disponibilidad de solo cuatro grupos de ocho canales bidireccionales y los beneficios que ofrecería una regulación que amplie la banda actual a la misma utilizada en muchas partes del mundo inclusive en Estados Unidos y muchos países de América Latina.

Por último, es muy importante destacar que algunos de los trabajos relacionados con la elaboración de esta tesis generaron una diversidad de resultados muy relevantes además de las publicaciones en revistas y congresos, como por ejemplo un método basado en *software defined radio* que dio lugar a la obtención de la patente “*Method and system for interference detection and diagnostic in cable networks*” en Estados Unidos, así como también el desarrollo de un proyecto de un producto hardware y software para la detección de interferencias de RF en bandas de 900 MHz que fue producto de exportación para una empresa nacional.

9.2. TRABAJOS FUTUROS

De los trabajos realizados en esta tesis surgen una gran cantidad de potenciales áreas de trabajo futuras, tales como, utilización de modelos de *machine learning* sobre los datos de telemetría

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

recibidos para realizar optimizaciones en los protocolos de *Adaptive Data Rate* (ADR) y para definir la colocación física de gateways.

Existe un área potencial de investigación en la seguridad de las redes LoRaWAN ante ataques de denegación de servicio y de falsificación de datos.

Por último, sería recomendable la realización de un proyecto para analizar los beneficios de la duplicación del ancho de banda disponible mediante la utilización del estándar de EE. UU. (US915) que utiliza la banda de 902 a 928 MHz., con el objetivo principal de proponer al ente regulador la expansión del espectro asignado con el objetivo de permitir comunicación bidireccional a todos los canales del espectro.

10. PUBLICACIONES Y TRABAJOS A QUE DIO LUGAR LA TESIS

10.1. PATENTES

Method and system for interference detection and diagnostic in cable networks - US10,284,258 – Latini, P.; Torchinsky, M.; Ambhul, D. – 2019

10.2. PUBLICACIONES EN REVISTAS INTERNACIONALES

Latini, P., Deploying DOCSIS 3.1: Using Software Defined Radio, Machine Learning and Big Data to Improve Network Trouble Detection, Society of Cable Telecommunication Engineers - JOURNAL OF NETWORK OPERATIONS, VOLUME 3, NUMBER 1 - October 2017, 2017, Exton, PA, USA

Latini, P., Distributed Access Architectures enable services in new markets, Broadband Library, SCTE-ISBE. Winter 2018.

10.3. PUBLICACIONES EN CONGRESOS INTERNACIONALES

Latini, P., An Analysis of How to Deploy Low Power WAN IoT Using HFC and Fiber Network Infrastructure, 2019 Fall Technical Forum – SCTE/ISBE, 2019, New Orleans

Latini, P., The Headend Evolution: Design Considerations For Deploying vCCAP And Other VNFs, 2020 Fall Technical Forum – SCTE/ISBE, 2020, Denver (Virtual)

10.4. PUBLICACIONES EN CONGRESOS

Latini P., DOCSIS 3.1, Remote - PHY y aplicaciones IoT, Convención Andina Link 2018, 2018, Cartagena, Colombia

Latini P., VCCAP Virtualización y su integración con Internet de las cosas, Convención Andina Link 2019, 2019, San José, Costa Rica

Latini P., Evolución del Acceso: Remote PHY-vCCAP y DOCSIS 4.0, SCTE Virtual

Latini P., How to evolve to Cloud Native Network Infrastructure in Telco Operators - SIT 2019, Buenos Aires, Argentina

Latini P., Making Distributed Access Technologies Real Operators - SIT 2019, Buenos Aires, Argentina

Latini P., Comunicaciones Avanzadas por Redes Híbridas Fibra Coaxial, 2018, IEEE COMSOC Session 2018, UTN FRBA, Buenos Aires, Argentina

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

10.5. LIBROS / NORMAS TECNICAS

Primer for PNM Best Practices in HFC Networks, Cablelabs, 2021, Electronic Publication

10.6. DESARROLLOS TECNOLOGICOS

Intraway Mobile Leakage Detection System - Sistema de detección de fugas e interferencias de radiofrecuencia en bandas de telefonía móvil - 2019

11. REFERENCIAS BIBLIOGRÁFICAS

- Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, 34–40.
- Ashton, K. (2009). *That ‘Internet of Things’ thing: In the real world, things matter more than ideas*. <http://www.rfidjournal.com/articles/view?4986>
- Bankov, D., Khorov, E., & Lyakhov, A. (2016). *On the Limits of LoRaWAN Channel Access*. 10–14.
- Bor, M., Roedig, U., Voigt, T., & Alonso, J. (2016). Do LoRa low-power wide-area networks scale? *MSWiM '16 Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 59–67. <https://doi.org/10.1145/2988287.2989163>
- Charles Bell. (2017). *MicroPython for the Internet of Things*.
- ChirpStack architecture—ChirpStack open-source LoRaWAN® Network Server*. (2022). <https://www.chirpstack.io/project/architecture/>
- D. F. Macedo, D. Guedes, L. F. M. Vieira, M. A. M. Vieira, and M. Nogueira,. (2018). Programmable Networks—From Software-Defined Radio to Software-Defined Networking. *IEEE Communications Surveys & Tutorials*, 17(2), 1102–1125.
- Resolucion 4653/2019*, (2019) (testimony of Enacom).
- Gershenfeld, N. (1999). *When Things Start to Think*. Henry Holt and Co.
- GNURadio. (2012). *What is GNU Radio? - GNU Radio*. https://wiki.gnuradio.org/index.php/What_is_GNU_Radio%3F

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Haas, Z. J., & Deng, J. (2003). On optimizing the backoff interval for random access schemes. *IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 51*, 2081–2090.

Hassan, Q. (2018). *Internet of Things A to Z: Technologies and Applications*. Wiley-IEEE Press.

Hata, M. (1980). Empirical Formula for Propagation Loss in Land Mobile Radio Services. *IEEE Transactions on Vehicular Communications, 29*(3), 317–325. <https://doi.org/10.1109/TVT.1980.23859>

Huda Ali Hashim, Salim Latif Mohammed, & Sadik Kamel Gharghan. (2018). Path Loss Model-Based PSO for Accurate Distance Estimation in Indoor Environments. *Journal of Communications, 13*(12).

Idris, S., Karunathilake, T., & Förster, A. (2022). Survey and Comparative Study of LoRa-Enabled Simulators for Internet of Things and Wireless Sensor Networks. *Sensors, 22*(15). <https://doi.org/10.3390/s22155546>

IEEE Std 145-2013 (Revision of IEEE Std 145-1993). (2014). IEEE. <https://ieeexplore.ieee.org/servlet/opac?punumber=6799984>

InfluxData. (2022). What is Time Series Data? | Definition, Examples, Types & Uses. *InfluxData*. <https://www.influxdata.com/what-is-time-series-data/>

IoT Analytics. (2021, September 22). State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion. *IoT Analytics*. <https://iot-analytics.com/number-connected-iot-devices/>

ITU, T. S. S. of. (2012). *Overview of the Internet of things*. International Telecommunication Union.

Referencias Bibliográficas

- Kleinrock, L., & Lam, S. (1975). Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation. *IEEE Transactions on Communications*, 23(4), 410–423. <https://doi.org/10.1109/TCOM.1975.1092814>
- Kranz, M. (2018). *6 ways the Internet of Things is improving our lives*. <https://www.weforum.org/agenda/2018/01/6-ways-the-internet-of-things-is-improving-our-lives/>
- Lee, J. S., & Miller, L. E. (1998). *CDMA Systems Engineering Handbook*. Artech House.
- Libelium. (2018). *50 Sensor Applications for a Smarter World*. http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/
- LimeSDR. (2022). Lime Microsystems. <https://limemicro.com/products/boards/limesdr/>
- LoRa Alliance. (2017). *LoRaWAN Specification 1.1*. LoRa Alliance. https://loralliance.org/sites/default/files/2018-05/lorawan1_0_2-20161012_1398_1.pdf
- LoRa Alliance. (2022, May 15). About LoRa Alliance®. *LoRa Alliance®*. <https://loralliance.org/about-lora-alliance/>
- LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL*. (n.d.). Retrieved July 17, 2022, from <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>
- LoRaWAN Specification 1.1*. (2017). LoRa Alliance. https://loralliance.org/sites/default/files/2018-05/lorawan1_0_2-20161012_1398_1.pdf
- Machina Research. (2015). *LPWA Technologies*. LoRa Alliance.
- Noreen, U., Bounceur, A., & Clavier, L. (2017). *A study of LoRa low power and wide area network technology*.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

- Petajajarvi, J., Mikhaylov, K., Roivainen, A., Hanninen, T., & Pettissalo, M. (2015). *On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology*.
- Polonelli, T., Brunelli, D., Marzocchi, A., & Benini, L. (2019). Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment. *Sensors*, *19*, 838. <https://doi.org/10.3390/s19040838>
- Resolucion 581/2018*, Presidencia de la Nacion (2018) (testimony of Presidencia de la Nacion).
- Proakis, J. (2000). *Digital Communications, 4 Ed.* McGraw Hill.
- Reynders, B., Meert, W., & Pollin, S. (2016). *Range and Coexistence Analysis of Long Range Unlicensed Communication*.
- Reynders, B., Wang, Q., & Pollin, S. (2018). A LoRaWAN Module for Ns-3: Implementation and Evaluation. *Proceedings of the 2018 Workshop on Ns-3*, 61–68. <https://doi.org/10.1145/3199902.3199913>
- Riegsecker, A. (2017). *Measuring Environmental Effects on LoRa Radios in Cold Weather Using 915 MHz* [PhD Thesis].
- rpp0. (2022). *Gr-lora* [C++]. <https://github.com/rpp0/gr-lora> (Original work published 2016)
- Rui Huang, Honggang Li, Belal Hamzeh, Yang-Yang, Shantidev Mohanty, & Yen Hsu. (2011). *Proposal for Evaluation Methodology for 802.16p*.
- Sakshama Ghoslya. (2018). *LoRa: Symbol Generation*. <http://www.sghoslya.com/p/lora-is-chirp-spread-spectrum.html>
- Seller, Olivier Bernard André & Sornin, Nicolas. (n.d.). *Low power long range transmitter*.

Referencias Bibliográficas

- Semtech. (n.d.). *LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL*. Retrieved September 2, 2022, from <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>
- Semtech. (2013). *SX1272/3/6/7/8: LoRa Modem Designer's Guide*. https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf
- Semtech. (2015). *LoRaTM Modulation Basics*. <https://www.semtech.com/uploads/documents/an1200.22.pdf>
- Slats, L. (2020, January 8). *A Brief History of LoRa: Three Inventors Share Their Story*. <https://blog.semtech.com/a-brief-history-of-lora-three-inventors-share-their-personal-story-at-the-things-conference>
- Smith, I. G. (2012). *The Internet of Things 2012—New Horizons*. Platinum.
- The Things Industries. (2022). *ABP vs OTAA*. <https://www.thethingsindustries.com/docs/devices/abp-vs-otaa/>
- The Things Network. (2017, September 8). *Ground breaking world record! LoRaWAN® packet received at 702 km (436 miles) distance*. The Things Network. <https://www.thethingsnetwork.org/article/ground-breaking-world-record-lorawan-packet-received-at-702-km-436-miles-distance>
- The Things Network. (2022). *LoRaWAN Architecture*. The Things Network. <https://www.thethingsnetwork.org/docs/lorawan/architecture/>
- Theodore Rappaport. (2002). *Wireless Communications—Principles and Practice*.
- Torrieri, D. (2018). *Principles of Spread-Spectrum Communication Systems, 4th ed.*
- Travis F. Collins, Robin Getz, Di Pu, & Alexander M. Wyglinski. (2018). *Software-Defined Radio for Engineers*.

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

- Valenta, V., Masalek, R., Baudoin, G., Villegas, M., Suarez, M., & Robert, F. (2010). Survey on Spectrum Utilization in Europe: Measurements, Analyses and Observations. *5th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications*.
- Vázquez-Rodas, A., Astudillo, D., Sanchez, C., Arpi, B., & Minchala, I. (2020). Experimental Evaluation of RSSI-based Positioning System with Low-cost LoRa Devices. *Ad Hoc Networks*, *105*, 102168. <https://doi.org/10.1016/j.adhoc.2020.102168>
- Villard, G., Sum, C.-S., Sun, C., Alemseged, Y., Lan, Z., & Harada, H. (2012). Efficiency of dynamic frequency selection based coexistence mechanisms for tv white space enabled cognitive wireless access points. *Wireless Communications, IEEE*, *19*, 69–75. <https://doi.org/10.1109/MWC.2012.6393520>
- Wixted, A. J., Kinnaird, P., Larijani, H., Tait, A., Ahmadinia, A., & Strachan, N. (2016). *Evaluation of LoRa and LoRaWAN for wireless sensor networks*.
- X. Cai, M. Zhou, and X. Huang. (2017). Model-Based Design for Software Defined Radio on an FPGA. *IEEE Access*, *5*, 8276–8283.

ANEXO 1 – CODIGO DE SENSORES Y APLICACIONES

Gateway

main.py

```

from network import WLAN
import time
import machine
from machine import RTC
import pycom

# Disable Hearbeat
pycom.heartbeat(False)

# Define callback function for Pygate events
def machine_cb (arg):
    evt = machine.events()
    if (evt & machine.PYGATE_START_EVT):
        # Green
        pycom.rgbled(0x103300)
    elif (evt & machine.PYGATE_ERROR_EVT):
        # Red
        pycom.rgbled(0x331000)
    elif (evt & machine.PYGATE_STOP_EVT):
        # RGB off
        pycom.rgbled(0x000000)

# register callback function
machine.callback(trigger = (machine.PYGATE_START_EVT |
machine.PYGATE_STOP_EVT | machine.PYGATE_ERROR_EVT), handler=machine_cb)

# Connect to a Wifi Network
wlan = WLAN(mode=WLAN.STA)
wlan.connect(ssid='XXXXXX', auth=(WLAN.WPA2, "XXXXXX"))

while not wlan.isconnected():
    time.sleep(1)

print("Wifi Connection established")

# Sync time via NTP server for GW timestamps on Events
rtc = RTC()
rtc.ntp_sync(server="0.pool.ntp.org")

# Read the GW config file from Filesystem
fp = open('/flash/config.json','r')
buf = fp.read()

machine.pygate_init(buf)

```

config.json

```

{
  "SX1301_conf": {
    "lorawan_public": true,
    "clksrc": 1,
    "antenna_gain": 0,
    "radio 0": {

```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
"enable": true,
"type": "SX1257",
"freq": 917200000,
"rssi_offset": -166,
"tx_enable": true,
"tx_freq_min": 915000000,
"tx_freq_max": 928000000
},
"radio_1": {
  "enable": true,
  "type": "SX1257",
  "freq": 917900000,
  "rssi_offset": -166,
  "tx_enable": false
},
"chan_multiSF_0": {
  "enable": true,
  "radio": 0,
  "if": -400000
},
"chan_multiSF_1": {
  "enable": true,
  "radio": 0,
  "if": -200000
},
"chan_multiSF_2": {
  "enable": true,
  "radio": 0,
  "if": 0
},
"chan_multiSF_3": {
  "enable": true,
  "radio": 0,
  "if": 200000
},
"chan_multiSF_4": {
  "enable": true,
  "radio": 1,
  "if": -300000
},
"chan_multiSF_5": {
  "enable": true,
  "radio": 1,
  "if": -100000
},
"chan_multiSF_6": {
  "enable": true,
  "radio": 1,
  "if": 100000
},
"chan_multiSF_7": {
  "enable": true,
  "radio": 1,
  "if": 300000
},
"chan_Lora_std": {
  "enable": true,
```

Anexo 1 – Código de sensores y aplicaciones

```
"radio": 0,  
"if": 300000,  
"bandwidth": 500000,  
"spread_factor": 8  
},  
"chan_FSK": {  
  "enable": false  
},  
"tx_lut_0": {  
  "pa_gain": 0,  
  "mix_gain": 8,  
  "rf_power": -6,  
  "dig_gain": 0  
},  
"tx_lut_1": {  
  "pa_gain": 0,  
  "mix_gain": 10,  
  "rf_power": -3,  
  "dig_gain": 0  
},  
"tx_lut_2": {  
  "pa_gain": 0,  
  "mix_gain": 12,  
  "rf_power": 0,  
  "dig_gain": 0  
},  
"tx_lut_3": {  
  "pa_gain": 1,  
  "mix_gain": 8,  
  "rf_power": 3,  
  "dig_gain": 0  
},  
"tx_lut_4": {  
  "pa_gain": 1,  
  "mix_gain": 10,  
  "rf_power": 6,  
  "dig_gain": 0  
},  
"tx_lut_5": {  
  "pa_gain": 1,  
  "mix_gain": 12,  
  "rf_power": 10,  
  "dig_gain": 0  
},  
"tx_lut_6": {  
  "pa_gain": 1,  
  "mix_gain": 13,  
  "rf_power": 11,  
  "dig_gain": 0  
},  
"tx_lut_7": {  
  "pa_gain": 2,  
  "mix_gain": 9,  
  "rf_power": 12,  
  "dig_gain": 0  
},  
"tx_lut_8": {  
  "pa_gain": 1,  
  "mix_gain": 15,  
  "rf_power": 13,  
  "dig_gain": 0  
},  
},
```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
"tx_lut_9": {
  "pa_gain": 2,
  "mix_gain": 10,
  "rf_power": 14,
  "dig_gain": 0
},
"tx_lut_10": {
  "pa_gain": 2,
  "mix_gain": 11,
  "rf_power": 16,
  "dig_gain": 0
},
"tx_lut_11": {
  "pa_gain": 3,
  "mix_gain": 9,
  "rf_power": 20,
  "dig_gain": 0
},
"tx_lut_12": {
  "pa_gain": 3,
  "mix_gain": 10,
  "rf_power": 23,
  "dig_gain": 0
},
"tx_lut_13": {
  "pa_gain": 3,
  "mix_gain": 11,
  "rf_power": 25,
  "dig_gain": 0
},
"tx_lut_14": {
  "pa_gain": 3,
  "mix_gain": 12,
  "rf_power": 26,
  "dig_gain": 0
},
"tx_lut_15": {
  "pa_gain": 3,
  "mix_gain": 14,
  "rf_power": 27,
  "dig_gain": 0
}
},
"gateway_conf": {
  "gateway_ID": "00010203FFFFFFFF0",
  "server_address": "192.168.2.20",
  "serv_port_up": 1700,
  "serv_port_down": 1700,
  "servers": [
    {
      "gateway_ID": "00010203FFFFFFFF0",
      "server_address": "192.168.2.20",
      "serv_port_up": 1700,
      "serv_port_down": 1700,
      "serv_enabled": true
    }
  ]
}
```

Anexo 1 – Código de sensores y aplicaciones

```
}  
}
```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

PySense 1

main.py

```
import socket
import time
import binascii
import pycom
from network import LoRa
from network import WLAN
from CayenneLPP import CayenneLPP

from pycoproc_1 import Pycoproc
from LIS2HH12 import LIS2HH12
from SI7006A20 import SI7006A20
from LTR329ALS01 import LTR329ALS01
from MPL3115A2 import MPL3115A2, ALTITUDE, PRESSURE

py = Pycoproc(Pycoproc.PYSENSE)
si = SI7006A20(py)
lt = LTR329ALS01(py)
li = LIS2HH12(py)

# Disable heartbeat LED
pycom.heartbeat(False)

# Connect to a Wifi Network
wlan = WLAN(mode=WLAN.STA)
wlan.connect(ssid='XXXXX', auth=(WLAN.WPA2, "XXXXX"))

##while not wlan.isdisconnected():
##    time.sleep(1)

##print("Wifi Connection established")

LORA_FREQUENCY = 915200000
LORA_NODE_DR = 5

'''
    call back for handling RX packets
'''
def lora_cb(lora):
    events = lora.events()
    if events & LoRa.RX_PACKET_EVENT:
        print("rx packet")
        if s is not None:
            frame, port = s.recvfrom(512) # longest frame is +-220
            print(port, frame)
    if events & LoRa.TX_PACKET_EVENT:
        print("tx_time_on_air: {} ms @dr {}", lora.stats().tx_time_on_air,
lora.stats().sftx)

# Initialize LoRa in LORAWAN mode.
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.AU915, adr=True)

for index in range(0, 71):
```

Anexo 1 – Código de sensores y aplicaciones

```
lora.remove_channel(index)
for i in range(8, 16):
    fq = LORA_FREQUENCY + (i * 200000)
    lora.add_channel(i, frequency=fq, dr_min=0, dr_max=LORA_NODE_DR)
    print("AU915 Adding channel up %s %s" % (i, fq))
for i in range(65, 66):
    fq = LORA_FREQUENCY + ((i-64) * 1600000) + 700000
    lora.add_channel(i, frequency=fq, dr_min=6, dr_max=6)
    print("AU915 Adding channel up %s %s" % (i, fq))

# create an OTAA authentication parameters
app_eui = binascii.unhexlify('0000000000000000')
app_key = binascii.unhexlify('D48982641B53A98874E3816EEFF218AB')

print("DevEUI: %s" % (binascii.hexlify(lora.mac())))
print("AppEUI: %s" % (binascii.hexlify(app_eui)))
print("AppKey: %s" % (binascii.hexlify(app_key)))

lora.join(activation=LoRa.OTAA, auth=(app_eui, app_key), timeout=0)

# wait until the module has joined the network
while not lora.has_joined():
    pycom.rgbled(0x140000)
    time.sleep(2.5)
    pycom.rgbled(0x000000)
    time.sleep(1.0)
    print('Not yet joined...')

print('OTAA joined')
pycom.rgbled(0x001400)

# create a LoRa socket
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)

# set the LoRaWAN data rate
s.setsockopt(socket.SOL_LORA, socket.SO_DR, LORA_NODE_DR)

s.setblocking(False)

lora.callback(trigger=( LoRa.RX_PACKET_EVENT |
                        LoRa.TX_PACKET_EVENT |
                        LoRa.TX_FAILED_EVENT ), handler=lora_cb)

time.sleep(4) # this timer is important and caused me some trouble ...

while True:
    pycom.rgbled(0x000014)
    lpp = CayenneLPP()

    print('\n\n** 3-Axis Accelerometer (LIS2HH12)')
    print('Acceleration', li.acceleration())
    print('Roll', li.roll())
    print('Pitch', li.pitch())
    lpp.add_accelerometer(1, li.acceleration()[0], li.acceleration()[1],
li.acceleration()[2])
    lpp.add_gyrometer(1, li.roll(), li.pitch(), 0)

    print('\n\n** Digital Ambient Light Sensor (LTR-329ALS-01)')
    print('Light', lt.light())
    lpp.add_luminosity(1, lt.light()[0])
    lpp.add_luminosity(2, lt.light()[1])
```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
print('\n\n** Humidity and Temperature Sensor (SI7006A20)')
print('Humidity', si.humidity())
print('Temperature', si.temperature())
lpp.add_relative_humidity(1, si.humidity())
lpp.add_temperature(1, si.temperature())

mpPress = MPL3115A2(py,mode=PRESSURE)
print('\n\n** Barometric Pressure Sensor with Altimeter (MPL3115A2)')
print('Pressure (hPa)', mpPress.pressure()/100)
lpp.add_barometric_pressure(1, mpPress.pressure()/100)

mpAlt = MPL3115A2(py,mode=ALTITUDE)
print('Altitude', mpAlt.altitude())
print('Temperature', mpAlt.temperature())
lpp.add_gps(1, 0, 0, mpAlt.altitude())
lpp.add_temperature(2, mpAlt.temperature())

print('Sending data (uplink)...')
s.send(bytes(lpp.get_buffer()))
pycom.rgbled(0x001400)
time.sleep(30)
```

PyTrack

main.py

```
import socket
import time
import binascii
import pycom
from network import LoRa
from network import WLAN

from CayenneLPP import CayenneLPP
from pycoproc_1 import Pycoproc
from LIS2HH12 import LIS2HH12
from L76GNSS import L76GNSS

py = Pycoproc(Pycoproc.PYTRACK)
li = LIS2HH12(py)

# after 60 seconds of waiting without a GPS fix it will
# return None, None
gnss = L76GNSS(py, timeout=120)

# Disable heartbeat LED
pycom.heartbeat(False)

# Connect to a Wifi Network
wlan = WLAN(mode=WLAN.STA)
wlan.connect(ssid='XXXXX', auth=(WLAN.WPA2, "XXXXX"))

##while not wlan.isdisconnected():
##    time.sleep(1)
```

Anexo 1 – Código de sensores y aplicaciones

```
##print("Wifi Connection established")

LORA_FREQUENCY = 915200000
LORA_NODE_DR = 5

'''
    call back for handling RX packets
'''
def lora_cb(lora):
    events = lora.events()
    if events & LoRa.RX_PACKET_EVENT:
        if s is not None:
            frame, port = s.recvfrom(512) # longest frame is +-220
            print(port, frame)
    if events & LoRa.TX_PACKET_EVENT:
        print("tx_time_on_air: %s ms @dr %s", lora.stats().tx_time_on_air,
lora.stats().sftx)

# Initialize LoRa in LORAWAN mode.
#lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.AU915, adr=True)
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.AU915)

for index in range(0, 71):
    lora.remove_channel(index)
for i in range(8, 16):
    fq = LORA_FREQUENCY + (i * 200000)
    lora.add_channel(i, frequency=fq, dr_min=0, dr_max=LORA_NODE_DR)
    print("AU915 Adding channel up %s %s" % (i, fq))

for i in range(65, 66):
    fq = LORA_FREQUENCY + ((i-64) * 1600000) + 700000
    lora.add_channel(i, frequency=fq, dr_min=6, dr_max=6)
    print("AU915 Adding channel up %s %s" % (i, fq))

# create an OTAA authentication parameters
app_eui = binascii.unhexlify('0000000000000000')
app_key = binascii.unhexlify('E73954605A2B98801E712B308DF3F56F')

print("DevEUI: %s" % (binascii.hexlify(lora.mac())))
print("AppEUI: %s" % (binascii.hexlify(app_eui)))
print("AppKey: %s" % (binascii.hexlify(app_key)))

lora.join(activation=LoRa.OTAA, auth=(app_eui, app_key), timeout=0)

# wait until the module has joined the network
while not lora.has_joined():
    pycom.rgbled(0x140000)
    time.sleep(2.5)
    pycom.rgbled(0x000000)
    time.sleep(1.0)
    print('Not yet joined...')

print('OTAA joined')
pycom.rgbled(0x001400)

# create a LoRa socket
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)

# set the LoRaWAN data rate
s.setsockopt(socket.SOL_LORA, socket.SO_DR, 2)
```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
s.setsockopt(socket.SOL_LORA, socket.SO_CONFIRMED, False)

s.setblocking(False)

lora.callback(trigger=( LoRa.RX_PACKET_EVENT |
                        LoRa.TX_PACKET_EVENT |
                        LoRa.TX_FAILED_EVENT ), handler=lora_cb)

time.sleep(4) # this timer is important and caused me some trouble ...

while True:
    pycom.rgbled(0x000014)
    lpp = CayenneLPP()

    print('\n\n** 3-Axis Accelerometer (LIS2HH12)')
    print('Acceleration', li.acceleration())
    print('Roll', li.roll())
    print('Pitch', li.pitch())
    lpp.add_accelerometer(1, li.acceleration()[0], li.acceleration()[1],
li.acceleration()[2])
    lpp.add_gyrometer(1, li.roll(), li.pitch(), 0)

    print('\n\n** GPS (L76GNSS)')
    loc = gnss.coordinates()
    if loc[0] == None or loc[1] == None:
        print('No GPS fix within configured timeout :-(')
    else:
        print('Latitude', loc[0])
        print('Longitude', loc[1])
        lpp.add_gps(1, loc[0], loc[1], 0)

    print('Sending data (uplink)...')
    s.send(bytes(lpp.get_buffer()))
    pycom.rgbled(0x001400)
    time.sleep(30)
```

PySense 2

main.py

```
import socket
import struct
import time
import binascii
import pycom
from network import LoRa
from network import WLAN
from CayenneLPP import CayenneLPP

from pycoproc_1 import Pycoproc
from LIS2HH12 import LIS2HH12
from SI7006A20 import SI7006A20
from LTR329ALS01 import LTR329ALS01
from MPL3115A2 import MPL3115A2, ALTITUDE, PRESSURE

py = Pycoproc(Pycoproc.PYSENSE)
si = SI7006A20(py)
```

Anexo 1 – Código de sensores y aplicaciones

```
lt = LTR329ALS01(py)
li = LIS2HH12(py)

# Disable heartbeat LED
pycom.heartbeat(False)

# Connect to a Wifi Network
wlan = WLAN(mode=WLAN.STA)
wlan.connect(ssid='XXXXX', auth=(WLAN.WPA2, "XXXX"))

##while not wlan.isdisconnected():
##    time.sleep(1)

##print("Wifi Connection established")

LORA_FREQUENCY = 915200000
LORA_NODE_DR = 2

'''
    call back for handling RX packets
'''
def lora_cb(lora):
    events = lora.events()
    if events & LoRa.RX_PACKET_EVENT:
        print ("rx packet")
        if s is not None:
            frame, port = s.recvfrom(512) # longest frame is +-220
            print(port, frame)
    if events & LoRa.TX_PACKET_EVENT:
        print("tx_time_on_air: {} ms @dr {}", lora.stats().tx_time_on_air,
lora.stats().sftx)

# Initialize LoRa in LORAWAN mode.
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.AU915, adr=True)

for index in range(0, 71):
    lora.remove_channel(index)
for i in range(9, 10):
    fq = LORA_FREQUENCY + (i * 200000)
    lora.add_channel(i, frequency=fq, dr_min=0, dr_max=LORA_NODE_DR)
    print("AU915 Adding channel up %s %s" % (i, fq))
for i in range(65, 65):
    fq = LORA_FREQUENCY + ((i-64) * 1600000) + 700000
    lora.add_channel(i, frequency=fq, dr_min=6, dr_max=6)
    print("AU915 Adding channel up %s %s" % (i, fq))

# create an ABP authentication parameters
dev_addr = struct.unpack(">l", binascii.unhexlify('0054d85d'))[0]
nwk_swkey = binascii.unhexlify('f6e0b4cc24482cfc19d49e12e14c42ca')
app_swkey = binascii.unhexlify('00b44872b0d662a5038f581f5161af8a')

print("DevEUI: %s" % (binascii.hexlify(lora.mac())))
print("NetSKey: %s" % (binascii.hexlify(nwk_swkey)))
print("AppSKey: %s" % (binascii.hexlify(app_swkey)))

lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))

# wait until the module has joined the network
while not lora.has_joined():
    pycom.rgbled(0x140000)
    time.sleep(2.5)
```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
pycom.rgbled(0x000000)
time.sleep(1.0)
print('Not yet joined...')

print('OTAA joined')
pycom.rgbled(0x001400)

# create a LoRa socket
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)

# set the LoRaWAN data rate
s.setsockopt(socket.SOL_LORA, socket.SO_DR, LORA_NODE_DR)

s.setblocking(False)

lora.callback(trigger=( LoRa.RX_PACKET_EVENT |
                        LoRa.TX_PACKET_EVENT |
                        LoRa.TX_FAILED_EVENT ), handler=lora_cb)

time.sleep(4) # this timer is important and caused me some trouble ...

while True:
    pycom.rgbled(0x000014)
    lpp = CayenneLPP()

    # print('\n\n** 3-Axis Accelerometer (LIS2HH12)')
    # print('Acceleration', li.acceleration())
    # print('Roll', li.roll())
    # print('Pitch', li.pitch())
    # lpp.add_accelerometer(1, li.acceleration()[0], li.acceleration()[1],
    li.acceleration()[2])
    # lpp.add_gyrometer(1, li.roll(), li.pitch(), 0)

    # print('\n\n** Digital Ambient Light Sensor (LTR-329ALS-01)')
    # print('Light', lt.light())
    # lpp.add_luminosity(1, lt.light()[0])
    # lpp.add_luminosity(2, lt.light()[1])

    print('\n\n** Humidity and Temperature Sensor (SI7006A20)')
    # print('Humidity', si.humidity())
    print('Temperature', si.temperature())
    # lpp.add_relative_humidity(1, si.humidity())
    lpp.add_temperature(1, si.temperature())

    # mpPress = MPL3115A2(py,mode=PRESSURE)
    # print('\n\n** Barometric Pressure Sensor with Altimeter (MPL3115A2)')
    # print('Pressure (hPa)', mpPress.pressure()/100)
    # lpp.add_barometric_pressure(1, mpPress.pressure()/100)

    # mpAlt = MPL3115A2(py,mode=ALTITUDE)
    # print('Altitude', mpAlt.altitude())
    # print('Temperature', mpAlt.temperature())
    # lpp.add_gps(1, 0, 0, mpAlt.altitude())
    # lpp.add_temperature(2, mpAlt.temperature())

    print('Sending data (uplink)...')
    s.send(bytes(lpp.get_buffer()))
```

Anexo 1 – Código de sensores y aplicaciones

```
pycom.rgbled(0x001400)
time.sleep(30)
```

LoRaSIM – Script de instalacion

```
wget https://www.lancaster.ac.uk/scc/sites/lora/lorasim-20170710.tgz

source ~/.local/bin/virtualenvwrapper.sh
tar xvf lorasim-20170710.tgz
cd lorasim/
VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source ~/.local/bin/virtualenvwrapper.sh
mkvirtualenv -p python2 lorasim
pip install numpy
pip install simpy
pip install matplotlib
```

FLoRa – Script de instalacin

```
wget -c https://github.com/omnetpp/omnetpp/releases/download/omnetpp-6.0/omnetpp-6.0-linux-x86_64.tgz
tar zxvf omnetpp-6.0-linux-x86_64.tgz
cd omnetpp-6.0/
source setenv
./configure
make
bin/omnetpp

Help -> Install Simulation Models.
INET
Flora

Project Explorer
Chose flora project - copy
Paste
```

FLoRa – omnetpp.ini

```
[General]
network = florasim.simulations.LoRaNetworkTest
**.maxTransmissionDuration = 4s
**.energyDetection = -110dBm
**.numberOfChannels = ${numberOfChannels = 1,2,4,8}
#**.numberOfChannels = 1

#cmdenv-autoflush = true
#cmdenv-status-frequency = 1000000s
#cmdenv-express-mode = false
#cmdenv-output-file = cmd_env_log.txt
**.vector-recording = false

rng-class = "cMersenneTwister"
**.loRaGW[*].numUdpApps = 1
**.loRaGW[0].packetForwarder.localPort = 2000
**.loRaGW[0].packetForwarder.destPort = 1000
**.loRaGW[0].packetForwarder.destAddresses = "networkServer"
**.loRaGW[0].packetForwarder.indexNumber = 0

**.networkServer.numApps = 1
**.networkServer.**.evaluateADRinServer = true
**.networkServer.app[0].typename = "NetworkServerApp"
**.networkServer.app[0].destAddresses = "loRaGW[0]"
```

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

```
**networkServer.app[0].destPort = 2000
**networkServer.app[0].localPort = 1000
**networkServer.app[0].adrMethod = "max"

**numberOfPacketsToSend = 0 #${numberOfPackets = 200..5000 step 200} #100
#obviously 0 means infinite number of packets
warmup-period = 0s
sim-time-limit = 1000s
simtime-resolution = -11
#repeat = 30

**timeToFirstPacket = exponential(30s)
**timeToNextPacket = 30s
**alohaChannelModel = false

#nodes features
**numberOfNodes = ${numberOfNodes=10,50,100,150,200,250,300,350,400}

**loRaNodes[*]**.initialX = uniform(0m,1000m)
**loRaNodes[*]**.initialY = uniform(0m,1000m)
**loRaNodes[*]**.initialLoRaSF = 7
**loRaNodes[*]**.initialLoRaBW = 125 kHz
**loRaNodes[*]**.initialLoRaCR = 1
**loRaNodes[*]**.initialLoRaTP = 20dBm
**loRaNodes[*]**.dataSize = 20bytes
**loRaNodes[*]**.initFromDisplayString = false
**loRaNodes[*]**.evaluateADRinNode = true

**loRaNodes[*].numApps = 1
**loRaNodes[*].app[0].typename = "SimpleLoRaApp"

#gateway features
**numberOfGateways = 1
**loRaGW[0]**.initialX = 500.00m
**loRaGW[0]**.initialY = 500.00m
**LoRaGWNic.radio.iAmGateway = true
**loRaGW[*]**.initFromDisplayString = false

#power consumption features
**loRaNodes[*].LoRaNic.radio.energyConsumer.typename = "LoRaEnergyConsumer"
**loRaNodes[*]**.energySourceModule = "^IdealEpEnergyStorage"
**loRaNodes[*].LoRaNic.radio.energyConsumer.configFile =
xmlDoc("energyConsumptionParameters.xml")

#general features
**sigma = 0
**constraintAreaMinX = 0m
**constraintAreaMinY = 0m
**constraintAreaMinZ = 0m
**constraintAreaMaxX = 1000m
**constraintAreaMaxY = 1000m
**constraintAreaMaxZ = 0m

LoRaNetworkTest.**.radio.separateTransmissionParts = false
LoRaNetworkTest.**.radio.separateReceptionParts = false

**ipV4Delayer.config = xmlDoc("cloudDelays.xml")
```

Anexo 1 – Código de sensores y aplicaciones

```
**radio.radioMediumModule = "LoRaMedium"  
#**.LoRaMedium.pathLossType = ${pathLossType = "LoRaLogNormalShadowing",  
"LoRaHataOkumura"}  
**.LoRaMedium.pathLossType = " LoRaHataOkumura"  
**.minInterferenceTime = 0s  
**.displayAddresses = false  
  
# cache features  
**.LoRaMedium.mediumLimitCacheType = "LoRaMediumCache"  
**.LoRaMedium.rangeFilter = "communicationRange"  
**.LoRaMedium.neighborCacheType = "LoRaNeighborCache"  
**.LoRaMedium.neighborCache.range = 546m  
**.LoRaMedium.neighborCache.refillPeriod = 3000s
```


ANEXO 2 – TABLAS DE MEDICIONES

Laboratorio

RSSI

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
1	-24.41	-26.56	-22.12	0.51
2	-30.58	-32.28	-26.09	0.69
4	-34.13	-36.55	-30.37	0.84

Rural

RSSI

Antena Alineada

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
100	-62.97	-82.56	-56.22	4.43
500	-82.99	-88.77	-66.77	3.18
1000	-93.16	-101.83	-90.10	2.99
2000	-103.12	-109.10	-100.59	2.03

Antena No Alineada

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
100	-65.06	-78.14	-56.58	6.1
500	-89.42	-95.29	-82.49	3.94
1000	-198.77	-107.82	-90.95	2.15
2000	-110.54	-109.17	-105.63	4.12

SNR

Antena Alineada

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
100	10.18	6.8	11.00	1.03
500	10.34	7.14	10.99	3.18
1000	5.42	0.12	8.19	2.99
2000	3.13	-4.32	5.34	2.03

Estudio de los efectos producidos por el crecimiento de internet de las cosas en el espectro de radiofrecuencia

Antena No Alineada

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
100	10.67	8.56	11.05	0.98
500	8.33	6.34	10.01	2.34
1000	4.18	-2.37	5.80	1.22
2000	0.10	-6.4	4.10	4.33

Urbano

RSSI

Antena Alineada

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
150	-99.66	-106.55	-96.61	2.99
300	-108.86	-110.12	-97.05	3.18
500	-116.82	-114.76	-103.35	4.12

Antena No Alineada

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
150	-104.86	-85.26	-84.32	6.1
300	-110.40	-101.07	-89.20	3.94
500	-121.34	-114.28	-97.18	2.15

SNR

Antena Alineada

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
150	6.23	4.51	8.45	2.12
300	-3.20	-7.02	0.09	2.85
500	-10.89	-11.58	-3.44	3.47

Antena No Alineada

Distancia [m]	Promedio [dBm]	Mínimo [dBm]	Máximo [dBm]	Desvio [dBm]
150	5.82	4.76	7.77	0.78
300	-6.10	-10.54	-7.15	3.09
500	-12.51	-13.23	-7.61	1.88

Anexo 2 – Tablas de mediciones