



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

MODELACIÓN DE INFRAESTRUCTURA TICAR HABILITANTE
PARA LAS SMART CITIES, CON FOCO A LAS REDES DE TELECOMUNICACIONES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

IGNACIO GABRIEL BUGUEÑO CÓRDOVA

PROFESOR GUÍA:
ALFONSO EHIJO BENBOW

MIEMBROS DE LA COMISIÓN:
JORGE SANDOVAL ARENAS
PABLO MEDINA COFRE

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: IGNACIO GABRIEL BUGUEÑO CÓRDOVA
FECHA: 2019
PROF. GUÍA: ALFONSO EHIJO BENBOW

MODELACIÓN DE INFRAESTRUCTURA TICAR HABILITANTE PARA LAS SMART CITIES, CON FOCO A LAS REDES DE TELECOMUNICACIONES

Una *Smart City* es un concepto asociado a un tipo de desarrollo urbano basado en la sostenibilidad, desarrollo que es capaz de responder a las necesidades básicas de instituciones, empresas y de los propios habitantes, tanto en el plano económico como en los aspectos operativos, sociales y ambientales. El rol de las TIC es pasivo, remitiéndose a la recopilación y análisis de datos, y la optimización en la utilización de infraestructuras, así como facilitar la comunicación entre los diferentes servicios de la ciudad. Asimismo, la introducción de nuevas tecnologías emergentes se convierte en una oportunidad por la cual diversas comunidades políticamente organizadas optan por adoptar, gracias a los beneficios energéticos que estas entregan, en conjunto con la inteligencia que estas proporcionan. De esta manera, múltiples ciudades han decidido integrar infraestructuras TICAR (Tecnología de Información, Telecomunicaciones, Automatización y Robótica) a fin de responder adecuadamente a diversas necesidades.

En el caso de Chile, si bien existen diversas iniciativas para apoyar la investigación, desarrollo e integración de nuevas tecnologías, estas no logran ser suficientes para el potencial mismo del país. En el contexto de Smart Cities, el país presenta un gran potencial para integrar tecnologías TICAR, gracias al sólido conocimiento existente de las mismas, la factibilidad económica para su adquisición, y la existencia de múltiples infraestructuras.

Referido a este último punto, una infraestructura de gran interés resulta ser el alumbrado público. Este sistema se caracteriza por iluminar zonas urbanas y sectores residenciales, con el objetivo de proporcionar la visibilidad adecuada para el normal desarrollo de actividades. Si existe una infraestructura física, que dispone de recursos energéticos, ¿por qué no diseñar un sistema de luminarias inteligentes, que responda como infraestructura habilitante para Smart Cities, con versatilidad para la integración de diversas tecnologías de comunicaciones e IoT?

En este contexto, la realización del presente trabajo de título tiene por objetivo modelar una infraestructura TICAR habilitante para Smart Cities, con foco a las redes de comunicaciones. Como infraestructura habilitante se propone el desarrollo de una red pública de luminarias inteligentes. Para esto, se realiza una revisión profunda de antecedentes que permitan contextualizar la problemática presente, una comparación preliminar entre tecnologías IoT que permita observar el desempeño y comportamiento de las mismas en entornos urbanos, y el diseño de la infraestructura habilitante desde la perspectiva de las tecnologías de la comunicación. A fin de validar esta propuesta, se considera dentro de los parámetros de diseño estandarizaciones y regulaciones de ambientes urbanos y tecnologías emergentes, junto a la utilización de plataformas de simulación que permitan caracterizar de manera robusta cada uno de los escenarios elaborados.

*"Y dijo Dios: Sea la luz; y fue la luz.
Y vio Dios que la luz era buena;
y separó Dios la luz de las tinieblas."
Génesis 1:3-4*

*"Porque de Él, y por Él, y para Él, son todas las cosas.
A Él sea la gloria por siempre. Amén."
Romanos 11:36*

Agradecimientos

Números 23:19 *"Dios no es hombre, para que mienta, ni hijo de hombre para que se arrepienta. Él dijo, ¿y no hará? Habló, ¿y no lo ejecutará?"*

El cumplimiento de esta etapa académica es el cumplimiento de una promesa de Dios, promesa a un joven que con un corazón lleno de fe estaba dispuesto a ser testimonio de Cristo en todo lugar. Una etapa que requirió de esfuerzo, una etapa que requirió de perseverancia, una etapa donde la integridad, la fe, la convicción y la santidad fueron puestas a prueba, y por medio de Él hubo victoria.

Primeramente agradezco a Dios por ser su hijo, por soplar cada día aliento de vida en mi corazón, por cubrir con un manto de amor, de bondad, de gracia, extendiendo su misericordia y su favor, proveyendo de sabiduría y conocimiento. Todo es en Él, es por Él y es para Él.

Agradezco a mi familia: a mi mami, a mi mamá y a mi hermano, quienes han sido y son mi hogar, a quienes amo, quienes han estado para formar, para sostener, para cubrir, para orar, en los momentos de alegría y de tristeza, caminando juntos hacia la meta del supremo llamamiento. Agradezco a mi tío, un hombre que ha dispuesto siempre su corazón en amor y en valentía, quien pese a la distancia siempre ha estado presente con un corazón fraternal.

Agradezco a mi casa espiritual, agradezco a mis padres espirituales, quienes me han formado e instruido desde mi adolescencia con una palabra de Dios, enseñando y desafiando a ser ejemplos en integridad, en honrar al Señor en toda circunstancia, en todo tiempo y en todo lugar, a amar su presencia y ser portadores de su gloria.

Agradezco a quienes han estado a lo largo de este camino: ex compañeros y profesores del colegio, ex compañeros de Escuela de Verano, compañeros y amigos de la Universidad, compañeros y profesores de Laboratorios de Investigación (Laboratorio de Robótica y Laboratorio de Comunicaciones Avanzadas), profesores de ayudantías y auxiliares, hermanos y amigos de la iglesia, líderes y equipos de trabajo, aún quienes ya no están mas fueron en algún momento parte de mi vida, gracias totales.

Agradezco a mi maestro Alfonso, por ser más que un Profesor Guía: un excelente profesional, pero ante todo una increíble persona. Por cada llamada, por cada locura, por cada oportunidad: gracias por tanta confianza, gracias por tanta estima.

Y por último agradezco al Señor, porque lo que ayer fue promesa hoy es realidad. Todo sea para la gloria y honra de su nombre.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Objetivo general	2
1.3. Objetivos específicos	2
1.4. Metodología	3
1.5. Descripción del documento	3
2. Antecedentes	4
2.1. Smart Cities	4
2.2. Modelamiento en telecomunicaciones	5
2.2.1. Modelo de referencia para los protocolos de la red de arquitectura en capas	5
2.2.2. Modelamiento en capa física	7
2.2.3. Modelamiento en capa de enlace y capa de red	8
2.2.4. Modelamiento en capa de transporte	11
2.2.5. Diseño de redes de ordenadores	12
2.3. Internet of Things	12
2.3.1. Capacidades relacionadas con dispositivos IoT	12
2.3.2. Aplicaciones de tecnologías IoT para las Smart Cities	14
2.3.3. Arquitectura IoT	14
2.3.4. Tecnologías	15
2.4. Wireless Sensor Network	18
2.4.1. Restricciones de consumo de energía	18
2.4.2. Seguridad en los nodos sensores	19
2.4.3. Interoperabilidad	19
2.4.4. Computación distribuida	19
2.4.5. Redes auto-organizadas	20
2.4.6. Modelo de capas	21
2.4.7. Protocolos MAC en redes WSN	22
2.5. Marco regulatorio: Internet of Things para Smart Cities	23
2.6. Infraestructura	24
2.6.1. Iluminación Pública: eficiencia energética	24
2.6.2. Iluminación Pública: estudios económicos	25
2.6.3. Iluminación Pública: impactos en la salud	27
2.7. Antecedentes metodológicos	28
2.7.1. Software de simulación en redes de telecomunicaciones	28

2.7.2.	Softwares de simulación en entornos IoT	29
3.	Metodología	32
3.1.	Descripción de trabajo de memoria	32
3.2.	Metodología para el desarrollo de modelos	32
3.3.	Metodología para el desarrollo de simulaciones	33
3.4.	Planificación de trabajo	33
3.5.	Modelamiento	34
3.5.1.	Comparativa entre tecnologías IoT	34
3.5.2.	Diseño de infraestructura TICAR habilitante	36
4.	Resultados	45
4.1.	Comparativa de tecnologías IoT	45
4.1.1.	Métricas en recepción	45
4.1.2.	Desglose de los paquetes recibidos	46
4.1.3.	Latencia del nivel de aplicación	47
4.2.	Diseño de infraestructura TICAR habilitante	49
4.2.1.	Frame Delivery Ratio	49
4.2.2.	End to End Delay	50
4.2.3.	RSSI Recibido	51
4.2.4.	Internet Cloud Queue	52
4.2.5.	Energía total consumida por nodo	53
5.	Discusión	54
5.1.	Comparativa de tecnologías IoT	54
5.1.1.	Métricas en recepción	54
5.1.2.	Desglose en la recepción de los paquetes	56
5.1.3.	Latencia del nivel de aplicación	57
5.2.	Diseño de infraestructura TICAR habilitante	58
5.2.1.	Frame Delivery Ratio	58
5.2.2.	End to End Delay	58
5.2.3.	RSSI Recibido	59
5.2.4.	Internet Cloud Queue	60
5.2.5.	Energía total consumida por nodo	61
6.	Conclusiones	62
	Bibliografía	65
A.	Antecedentes	68
A.1.	Modelo TCP/IP	68
A.2.	Protocolo MAC: CSMA/CA	69
A.2.1.	Distributed Coordination Function	69
A.2.2.	RTS/CTS	69
A.3.	Protocolo TCP	71
A.4.	Protocolo UDP	72
A.5.	Diseño de redes de ordenadores	72
A.5.1.	Paradigma Cliente-Servidor	72

A.5.2. Paradigma P2P	73
A.6. Protocolos de enlace de datos en IoT	73
A.6.1. IEEE 802.15.4	73
A.6.2. IEEE 802.11ah	75
A.7. Infraestructura	76
A.7.1. Iluminación Pública: eficiencia energética	76
B. Desarrollos	78

Índice de Tablas

2.1. Comparativa de modelos OSI, TCP/IP, WLAN y WSN.	22
3.1. Parámetros de diseño en modelo físico del canal.	35
3.2. Parámetros de diseño en tamaño de paquetes y frames.	35
3.3. Parámetros de diseño en configuración de frecuencia de operación de módulos LoRa.	43
3.4. Parámetros de diseño en configuración de antenas para Transceiver LoRa. . .	43
3.5. Parámetros de diseño en configuración de capa MAC.	43
3.6. Parámetros de diseño en configuración de Gateways LoRa.	43
4.1. Número de paquetes recibidos, en función de la distancia de separación d y protocolo MAC.	45
4.2. Packet reception rate , en función de la distancia de separación d y protocolo MAC.	46
4.3. Packet loss rate , en función de la distancia de separación d y protocolo MAC.	46
4.4. Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 10$ [m].	46
4.5. Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 15$ [m].	46
4.6. Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 20$ [m].	47
4.7. Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 25$ [m].	47
4.8. Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 30$ [m].	47
4.9. Desglose en latencia de número de paquetes recibidos, para distancia $d=10$ [m].	47
4.10. Desglose en latencia de número de paquetes recibidos, para distancia $d=15$ [m].	48
4.11. Desglose en latencia de número de paquetes recibidos, para distancia $d=20$ [m].	48
4.12. Desglose en latencia de número de paquetes recibidos, para distancia $d=25$ [m].	48
4.13. Desglose en latencia de número de paquetes recibidos, para distancia $d=30$ [m].	48
4.14. Frame Delivery Ratio - Condiciones normales de la red.	49
4.15. Frame Delivery Ratio - Sobrecarga de tráfico en la red.	49
4.16. End to End Delay - Condiciones normales de la red.	50
4.17. End to End Delay - Sobrecarga de tráfico en la red.	50
4.18. RSSI Recibido - Condiciones normales de la red.	51
4.19. RSSI Recibido - Sobrecarga de tráfico en la red.	51

4.20. Internet Cloud Queue - Condiciones normales de la red.	52
4.21. Internet Cloud Queue - Sobrecarga de tráfico en la red.	52
4.22. Energía total consumida por nodo - Condiciones normales de la red.	53
4.23. Energía total consumida por nodo - Sobrecarga de tráfico en la red.	53
A.1. Estructura RTS Frame.	70
A.2. Estructura CTS & ACK Frame.	71

Índice de Ilustraciones

2.1. Ilustración de una Smart City.	4
2.2. Modelo de capas OSI.	5
2.3. Modelo de capas TCP/IP.	7
2.4. Protocolo ALOHA.	10
2.5. Protocolo Slotted ALOHA.	11
2.6. Diagrama de arquitectura general IoT.	15
2.7. Wireless Sensor Network: Modelo de capas.	21
2.8. Logos institucionales: (a) Subsecretaría de Telecomunicaciones, (b) Superintendencia de Electricidad y Combustibles, (c) Instituto Nacional de Normalización.	23
2.9. Ambiente de Simulador OMNeT++.	29
2.10. Seguimiento de secuencias en análisis de protocolo, en simulador OMNeT++.	29
2.11. Visualización 3D de desempeño de redes en simulador OMNeT++.	30
3.1. Escenario de simulación para comparación de tecnologías IoT.	35
3.2. Recreación de ambiente urbano y sistema de iluminación pública.	37
3.3. Case de protección para plataforma de desarrollo Raspberry Pi.	38
3.4. Luminarias Inteligentes: Arquitectura de red de sensores WSN, basado en IoT.	40
3.5. Luminarias Inteligentes: Diseño de red WSN - Topología A.	41
3.6. Luminarias Inteligentes: Diseño de red WSN - Topología B.	41
3.7. Luminarias Inteligentes: Diseño de red WSN - Topología C.	41
3.8. LoRa Dataframe.	42
5.1. Número de paquetes recibidos, en función de la distancia.	54
5.2. Tasa de paquetes recibidos, en función de la distancia.	55
5.3. Tasa de paquetes perdidos, en función de la distancia.	55
5.4. Desglose paquetes recibidos, con distancia de separación $d = 15[m]$	56
5.5. Desglose paquetes recibidos, con distancia de separación $d = 20[m]$	56
5.6. Desglose paquetes recibidos, con distancia de separación $d = 25[m]$	56
5.7. Desglose paquetes recibidos, con distancia de separación $d = 30[m]$	56
5.8. Latencia del nivel de aplicación, con distancia de separación $d = 15[m]$	57
5.9. Latencia del nivel de aplicación, con distancia de separación $d = 20[m]$	57
5.10. Latencia del nivel de aplicación, con distancia de separación $d = 25[m]$	57
5.11. Latencia del nivel de aplicación, con distancia de separación $d = 30[m]$	57
5.12. Frame Delivery Ratio, en red con condiciones normales de tráfico.	58
5.13. Frame Delivery Ratio, en red con sobrecarga de tráfico.	58

5.14. End-to-End Delay promedio, en red con condiciones normales de tráfico.	59
5.15. End-to-End Delay promedio, en red con sobrecarga de tráfico.	59
5.16. RSSI recibido promedio, en red con condiciones normales de tráfico.	59
5.17. RSSI recibido promedio, en red con sobrecarga de tráfico.	59
5.18. Internet Cloud Queue promedio, en red con condiciones normales de tráfico.	60
5.19. Internet Cloud Queue promedio, en red con sobrecarga de tráfico.	60
A.1. Arquitectura IEEE 802.15.4	74
A.2. Estructura MCPS Data Request.	74
A.3. Estructura MCPS Data Confirm.	75
A.4. Estructura MCPS Data Indication.	75
A.5. Protocolo de transferencia de datos de la capa MAC, en IEEE 802.15.4.	76

Acrónimos

5G:	5th Generation of Cellular Mobile Communications.
6LoWPAN:	IPv6 over Low power Wireless Personal Area Networks.
AChEE:	Agencia Chilena de Eficiencia Energética.
ACK:	<i>Acknowledgement</i> . Señal que se transmite entre procesos de comunicación para indicar la recepción de un mensaje.
ADR:	Adaptive Data Rate.
ALOHA:	Sistema de redes de computadoras desarrollado en la Universidad de Hawái.
ANSI:	American National Standards Institute.
API:	Application Programming Interface.
ARCEP:	Autorité de Régulation des Communications Électroniques et des Postes.
BDA:	Big Data Analytics.
BER:	Bit Error Rate.
Castalia:	Simulador para redes de sensores inalámbricos, redes de área corporal y, en general, redes de dispositivos integrados de bajo consumo.
Cisco:	Empresa global con sede en San José, California, Estados Unidos, principalmente dedicada a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones.
CoAP:	Constrained Application Protocol.
CPU:	Central Processing Unit.

CSMA:	Carrier Sense Multiple Access.
CSMA/CA:	Carrier Sense Multiple Access with Collision Avoidance.
CSMA/CD:	Carrier Sense Multiple Access with Collision Detection.
CSMA/CR:	Carrier Sense Multiple Access with Collision Resolution.
DDoS:	<i>Denial of Service.</i> Ataque informático a una red, que causa que un servicio o recurso sea inaccesible a los usuarios legítimos.
EE:	Eficiencia Energética.
FCC:	Federal Communications Commission.
FLoRa:	<i>Framework for LoRa.</i> Simulador de eventos discretos para redes LoRa.
FSK:	Frequency Shift Keying.
GHG:	Gases de Efecto Invernadero.
GTS:	<i>Guaranteed Time Slots</i> . Mecanismo del estándar IEEE 802.15.4 utilizado para asignar una duración específica de tiempo dentro de una estructura de datos para aplicaciones que requieren un ancho de banda de datos específico, a fin de minimizar el desperdicio de ancho de banda del canal.
HP:	Hewlett-Packard.
HTTP:	Hypertext Transfer Protocol.
IDE:	<i>Integrated Development Environment.</i> Aplicación informática que proporciona servicios integrales para facilitar el desarrollo de software.
IEC:	International Electrotechnical Commission.
IEEE:	Institute of Electrical and Electronics Engineers.
INN:	Instituto Nacional de Normalización de Chile.
IoT:	Internet of Things.
IP:	<i>Internet Protocol.</i> Protocolo de comunicación de datos digitales.
IPv6:	<i>Internet Protocol version 6.</i> Versión más reciente del Protocolo de Internet (IP), proporciona un sistema de identificación y ubicación para dispositivos en redes y rutas de tráfico a través de Internet.

ISO:	International Organization for Standardization.
IT:	Information Technology.
LAN:	Local Area Network.
LBT:	<i>Listen Before Talk</i> . Principio de funcionamiento de CSMA/CA.
LED:	Light-Emitting Diode.
LoRa:	<i>Long Range</i> . Tecnología de comunicación inalámbrica de datos patentada y desarrollada por <i>Cycleo</i> de Grenoble, Francia.
LoRaWAN:	<i>Long-Range Wide Area Network</i> . Tecnología inalámbrica emergente diseñada para redes WAN de baja potencia con bajo costo, movilidad, seguridad y comunicación bidireccional para aplicaciones de IoT.
LPWAN:	Low-Power Wide-Area Networking.
LTE:	<i>Long Term Evolution</i> . Estándar para comunicaciones inalámbricas de transmisión de datos de alta velocidad.
M2M:	Machine-to-Machine.
MAC:	Medium Access Control.
MIC:	Ministry of Internal Affairs and Communications.
MiWi:	Protocolo inalámbrico, para redes de área personal, basado en el estándar IEEE 802.15.4.
Modelo OSI:	Open System Interconnection.
MQTT:	<i>Message Queuing Telemetry Transport</i> . Protocolo de mensajería basado en el estándar ISO/IEC 20922.
NB-IoT:	Narrowband-IoT.
NED:	<i>Network Description</i> . Lenguaje de programación modular utilizado para el diseño estructural de modelos en entornos de simulación.
OFCOM:	The Office of Communications .
OMNeT++:	Simulador de eventos discretos de redes orientado a objetos.

P2P:	<i>Peer-to-Peer</i> . Modelo de red que permite el intercambio de archivos, aplicaciones, fotos.
PHY:	Abreviatura de la capa física del modelo OSI, y se refiere a los circuitos necesarios para implementar las funciones de la capa física.
PSK:	Phase Shift Keying.
REST:	<i>Representational State Transfer</i> . Modelo de arquitectura de software para sistemas hipermedia distribuidos.
RF4CE:	Zigbee specification.
RTS/CTS:	<i>Request to Send / Clear to Send</i> . Es el mecanismo opcional utilizado por el protocolo de red inalámbrica IEEE 802.11 para reducir colisiones.
RX:	Abreviatura de <i>recepción</i> .
SDK:	Software Development Kit.
SEC:	Superintendencia de Electricidad y Combustibles de Chile.
Semtech:	Proveedor de circuitos integrados analógicos y mixtos..
SINR:	Signal-to-Interference-plus-Noise Ratio.
S-MAC:	Sensor MAC.
SNR:	Signal-to-Noise Ratio.
SUBTEL:	Subsecretaría de Telecomunicaciones de Chile.
TCP:	<i>Transmission Control Protocol</i> . Protocolo de control de transmisión.
Thread:	Tecnología de red de baja potencia basada en IPv6 para productos IoT.
TIC:	Tecnologías de la Información y la Comunicación.
TICAR:	Tecnologías de la Información, Comunicación, Automatización y Robótica.
T-MAC:	Timeout MAC.
TX:	Abreviatura de <i>transmisión</i> .

- UDP:** *User Datagram Protocol.* Protocolo de datagramas de usuario.
- UNE-EN:** *Una Norma Española.* Conjunto de normas, normas experimentales e informes creados en los Comités Técnicos de Normalización de la Asociación Española de Normalización y Certificación.
- V2V:** Vehicle-to-Vehicle communication systems.
- WAN:** *Wide Area Network.* Red de dispositivos que une varias redes locales.
- WiFi:** *Wireless Fidelity.* Tecnología para redes inalámbricas de área local de dispositivos basados en el estándar IEEE 802.11.
- WLAN:** Wireless Local Area Network.
- WPAN:** Wireless Personal Area Network.
- WSN:** *Wireless Sensor Networks.* Redes de sensores inalámbricos.
- ZigBee:** Nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica y de bajo consumo, basada en el estándar IEEE 802.15.4 de redes WPAN.

Capítulo 1

Introducción

1.1. Motivación

Una *Smart City* es un concepto asociado a un tipo de desarrollo urbano basado en la sostenibilidad, desarrollo que es capaz de responder a las necesidades básicas de instituciones, empresas y de los propios habitantes, tanto en el plano económico como en los aspectos operativos, sociales y ambientales. En paralelo, la introducción de nuevas tecnologías emergentes se convierte en una oportunidad por la cual diversas comunidades políticamente organizadas optan por adoptar, gracias a los beneficios energéticos que estas entregan.

Grandes ciudades a nivel mundial han decidido integrar tecnologías TICAR en sus urbes, a fin de responder a necesidades básicas de instituciones, empresas y los propios habitantes, basándose en la sostenibilidad y en la eficiencia. Ahora bien, estas ciudades (Tokio, Londres, Nueva York, entre otros) presentan un denominador común: pertenecen a países desarrollados, donde la investigación en ciencias y tecnología es posible gracias a la inversión público-privado existente, lo cual permite liderar avances en tecnologías y su misma integración.

En el caso de Chile, si bien existen diversas iniciativas para apoyar la investigación, desarrollo e integración de nuevas tecnologías, estas no logran ser suficientes para el potencial mismo del país. En el contexto de Smart Cities, el país presenta un gran potencial para integrar tecnologías TICAR, gracias al sólido conocimiento existente de las mismas, la factibilidad económica para su adquisición, y la existencia de múltiples infraestructuras.

Referido a este último punto, una infraestructura de gran interés resulta ser el alumbrado público. Este sistema se caracteriza por iluminar zonas urbanas y sectores residenciales, con el objetivo de proporcionar la visibilidad adecuada para el normal desarrollo de actividades. En cuanto al funcionamiento de este sistema, en iluminación urbana pueden ser empleadas lámparas de filamento incandescente, lámparas de luz mixta, lámparas de sodio de alta presión, fuentes LED, entre otros. Las lámparas de sodio de alta presión vienen a ser la tecnología de alumbrado tradicional, en potencias de 50[W], 70[W], 100[W], 150[W], 250[W], 400[W] y 600[W], lo cual deja en manifiesto la existencia de fuentes de energía en cada uno de los postes de iluminación. Si existe una infraestructura física, que dispone de recursos energéticos,

¿por qué no diseñar un sistema de luminarias inteligentes, que responda como infraestructura habilitante para Smart Cities, con versatilidad para la integración de diversas tecnologías de comunicaciones e IoT?

1.2. Objetivo general

El objetivo general del presente trabajo de memoria consiste en diseñar una infraestructura TICAR habilitante para las Smart Cities, con foco a las redes de telecomunicaciones. Para esto, se propone como infraestructura habilitante una red de sensores en luminarias inteligentes, caracterizadas por presentar tanto sistemas de autogestión como sistemas de inteligencia, interconexión digital que se enmarca en el paradigma IoT. De esta manera, existe un enfoque de optimización en los procesos operativos asociados al funcionamiento de las luminarias públicas, en conjunto con la adición de inteligencia a la red.

1.3. Objetivos específicos

Los objetivos específicos corresponden a una serie de pasos que permiten el desarrollo de los objetivos fijados. Estos objetivos son:

- Modelamiento de infraestructura TICAR habilitante, en redes de luminarias inteligentes, para calles inteligentes, en Smart Cities.
 - Revisión del estado del arte en redes de luminarias públicas, y desarrollos e implementaciones actuales de infraestructuras TICAR habilitantes.
 - Modelamiento de red de sensores en luminarias inteligentes, con enfoque en telecomunicaciones.
 - Modelamiento de red de sensores en luminarias inteligentes, con enfoque en inteligencia y paradigma Internet of Things.
 - Modelamiento de red de sensores en luminarias inteligentes, bajo una mirada económica.
- Realización de simulaciones para validación de modelos propuestos.
 - Utilización de softwares de simulación de redes de telecomunicaciones tradicionales.
 - Utilización de softwares de simulación de redes en entorno Internet of Things.
 - Utilización de softwares para el cálculo de proyecciones e indicadores económicos de interés.
- Validación de modelo de infraestructura TICAR habilitante propuesta, bajo tres enfoques: redes de telecomunicaciones tradicionales, entorno IoT y mirada económica.

1.4. Metodología

La metodología propuesta para el desarrollo del trabajo se divide en las siguientes actividades:

1. Metodología para el desarrollo de modelos.
 - Modelamiento con enfoque en redes de telecomunicaciones.
 - Modelamiento con enfoque en costos.
2. Metodología para el desarrollo de simulaciones.
 - Análisis de escenarios.
 - Utilización de software para simulación en redes de telecomunicaciones: OM-NeT++.
 - Utilización de softwares para simulación en entornos IoT: Castalia y FLoRa.
 - Utilización de software para simulación de coste de modelo: Excel.
3. Análisis, discusión y conclusiones.

1.5. Descripción del documento

En esta sección se describe la estructura del trabajo de título. Este trabajo se divide en tres capítulos que se describen a continuación.

El *Capítulo Introducción* presenta las motivaciones para el modelamiento de una infraestructura TICAR habilitante para las Smart Cities, con foco a las redes de telecomunicaciones. Se enuncian los objetivos generales y específicos del trabajo de memoria, junto con la metodología utilizada.

El *Capítulo Antecedentes* presenta los contenidos que enmarcan los principales conceptos referentes al modelamiento de la infraestructura TICAR, tanto desde una perspectiva tecnológica como normativa.

El *Capítulo Metodología* muestra en detalle la metodología propuesta para obtener los resultados que permitirán determinar la validez del modelo propuesto, en conjunto con el cumplimiento de los objetivos del trabajo.

El *Capítulo Resultados* presenta cada uno de los resultados obtenidos tras la aplicación del marco metodológico previamente descrito.

El *Capítulo Discusión* presenta un análisis y discusión de cada uno de los resultados obtenidos tras la aplicación del marco metodológico.

Finalmente, el *Capítulo Conclusiones* presenta cada una de las resoluciones adoptadas una vez finalizada la realización del presente documento.

Capítulo 2

Antecedentes

2.1. Smart Cities

Si bien no existe una definición única aceptada, un consenso contemporáneo define una *Smart City* como una estrategia de desarrollo urbano coherente administrado por los gobiernos de las ciudades, a fin de planificar y alinear a largo plazo la gestión de los diversos bienes de infraestructura y servicios municipales de la ciudad, con el único objetivo de mejorar la calidad de vida de los ciudadanos. El rol de las TIC (Tecnologías de la Información y Comunicación) en la visión actual de Smart City es pasivo, principalmente relacionado con la recopilación y análisis de datos, predicción, y la optimización en la utilización de las infraestructuras, así como facilitar la comunicación entre los diferentes servicios de la ciudad.

Aunque es extremadamente complejo, las ciudades inteligentes de hoy en día se pueden describir mejor como una metodología de planificación urbana, que depende en gran medida de las TIC para recopilar los datos necesarios y tomar decisiones óptimas de ingeniería y planificación. Esto significa que las estrategias de la ciudades se deben planificar con mucha anticipación, dedicando grandes presupuestos de inversión a través de grandes presupuestos de infraestructura.



Figura 2.1: Ilustración de una Smart City.¹

Considerando la importancia que adquieren las TIC en este tipo de desarrollo urbano, es que resulta fundamental conocer a fondo el background teórico que las describe, de manera de tener una panorámica global al momento de diseñar soluciones para las mismas.

¹Fuente: <https://pngimage.net/smart-city-png-5/>

2.2. Modelamiento en telecomunicaciones

2.2.1. Modelo de referencia para los protocolos de la red de arquitectura en capas

Durante las últimas dos décadas se ha presentado un enorme crecimiento en la cantidad y tamaño de las redes, en donde muchas de ellas se desarrollaron bajo la implementación de hardware y software diferentes. Como resultado, muchas de las redes eran incompatibles, volviéndose muy difícil poder comunicarse entre sí. Para solucionar este problema, la ISO (Organización Internacional para la Normalización) reconoció que era necesario crear un modelo de red ayudara a los diseñadores de red a implementar redes que pudieran comunicarse y trabajar en conjunto. Esta identificación de la problemática conllevó al diseño e implementación de un modelo de referencia de capas para las comunicaciones, estandarizando y regularizando la implementación de las redes.

Un modelo de referencia proporciona una referencia común para mantener consistencia en todos los tipos de protocolos y servicios de red, permitiendo tener una visión más clara de las funciones y los procesos involucrados. Así, en este contexto, la utilización de un modelo de capas permite asistir en el diseño del protocolo, fomentar la competencia, y proporcionar un lenguaje común en la descripción de las funciones y capacidades de la red.

2.2.1.1. Modelo OSI

El modelo de interconexión de sistemas abiertos, o mayormente conocido como modelo OSI, establece un modelo de referencia para los protocolos de red por medio de la implementación de siete capas que definen las fases por la que deben pasar los datos, a fin de ser transmitidos desde una máquina a otra sobre la red de telecomunicaciones. Estas siete capas se presentan a continuación, visualizándose en la *Figura 2.2*.

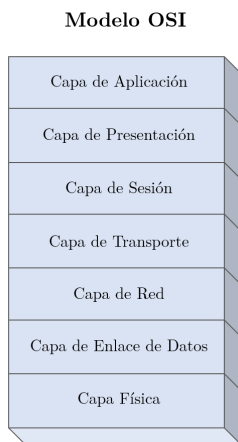


Figura 2.2: Modelo de capas OSI.

- **Capa física:** corresponde a la capa más baja del modelo OSI, y se encuentra encargada de transmitir y recibir una secuencia de bits sin procesar, esto por medio del medio físico. Esta capa describe las interfaces eléctrica, óptica, mecánica y funcional al medio físico, llevando las señales a capas superiores. Así, proporciona la codificación de datos, modificando el modelo de señal digital, interpreta el estado de la señal frente a secuencias de bits, determina cómo se transmitirán los bits codificados, y permite transmitir esta información como señales eléctricas u ópticas, determinando los voltios que deben ser utilizados para representar el estado de la señal.
- **Capa de enlace de datos:** esta capa inferior del modelo OSI permite transferir sin errores la trama de datos desde un nodo a otros, por medio de la capa física, posibilitando que las capas superiores logren asumir la transmisión sin error. Esta capa permite establecer y finalizar el vínculo lógico entre dos nodos de una red, tener un control del tráfico de las tramas (cabeceras más datos de la información), transmitiéndolas y recibéndolas de manera secuencial, detectar los errores presentes ocurridos en la capa física, los cuales alteran la información transmitida, y administrar el acceso al medio por parte de los datos.
- **Capa de red:** esta capa intermedia del modelo OSI controla el funcionamiento general de la subred de comunicación, estableciendo qué ruta de acceso físico deberían tomar los datos en función de las condiciones actuales de la red, entre otros parámetros de relevancia. Esta capa permite enrutar (direccionar) los paquetes entre las redes, controlar el tráfico de las subredes, fragmentar las tramas en los casos que resulten necesarios, asignar direcciones lógicas físicas de los equipos, y disponer de funciones de contabilidad que permitan seguir las tramas reenviadas por sistemas intermedios de subred. Además, ingresando ya al software de esta capa, este debe ser capaz de generar los headers que permitan reconocer la información transmitida en cada una de las capas.
- **Capa de transporte:** esta capa intermedia del modelo OSI garantiza que los mensajes se entregan sin errores, secuencialmente, sin presentar pérdidas o duplicaciones de la información. Esta capa segmenta los mensajes de la capa de sesión (capa superior) en unidades más pequeñas, realiza confirmaciones en la entrega de mensajes, controla el tráfico de los mensajes y multiplexa las sesiones de transmisión respectivas.
- **Capa de sesión:** esta capa intermedia del modelo OSI permite establecer sesiones entre procesos que se ejecutan en diferentes estaciones, controlando el estado de las conexiones de la red (sesiones), y entregando soporte para esta, por medio de la habilitación de procesos de comunicación a través de la red y de patrones de seguridad.
- **Capa de presentación:** esta capa superior del modelo OSI da formato a los datos que serán presentados en la capa superior de aplicación, actuando como intérprete de la información presente en la red. Esta capa permite realizar una conversión desde los códigos a los caracteres respectivos, una conversión de los datos, la comprensión de estos para su transmisión y su posterior cifrado en términos de seguridad.
- **Capa de aplicación:** esta capa superior del modelo OSI opera como ventana entre los usuarios y los procesos de aplicaciones, a fin de tener acceso a los servicios de red. Esta capa permite acceder a archivos remotos, establecer comunicación entre múltiples procesos, administración de la red y de terminales virtuales, entre otros.

2.2.1.2. Modelo TCP/IP

El modelo TCP/IP, establece un modelo de referencia para los protocolos de red por medio de la implementación de cuatro capas que definen las fases por la que deben pasar los datos, a fin de ser transmitidos desde una máquina a otra sobre la red de telecomunicaciones. TCP significa *Protocolo de Control de Transmisión*, el cual corresponde a uno de los protocolos fundamentales en Internet, mientras que IP significa *Protocolo de Internet*, el cual corresponde a un número que identifica, de manera lógica y jerárquica, a la interfaz de red de un dispositivo electrónico. Estas cuatro capas se presentan a continuación, visualizándose en la *Figura 2.3*.

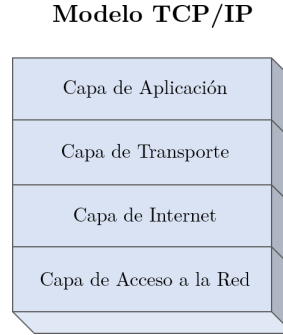


Figura 2.3: Modelo de capas TCP/IP.

Una descripción exhaustiva del presente modelo se puede encontrar en el capítulo *Apéndice A, A.1 Modelo TCP/IP*.

2.2.2. Modelamiento en capa física

Un aspecto importante en el modelamiento de canales inalámbricos es estimar el *average path loss* entre dos nodos, o en general entre dos puntos en el espacio. Para redes WSN, donde la separación entre los nodos va desde un par de metros hasta cientos de metros, se ha demostrado que el modelo *lognormal shadowing* proporciona estimaciones precisas para la pérdida de trayectoria promedio [29]. De esta manera, la siguiente fórmula permite obtener el *path loss* en dB, como función de la distancia entre dos nodos y de otros parámetros.

$$P_L(d) = P_L(d_0) + 10 \cdot \eta \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (2.1)$$

$P_L(d)$ corresponde al *path loss* para una distancia d , $P_L(d_0)$ es el *path loss* conocido para una distancia d_0 , η corresponde al exponente asociado al *path loss*, y X_σ es variable aleatoria con distribución gaussiana con media cero y una desviación estándar σ , vale decir, corresponde a AWGN ($X_\sigma \sim \mathcal{N}(0, \sigma^2)$).

Ahora bien, si se busca capturar la correlación entre las dos direcciones de un enlace, el modelo lognormal shadowing presenta una baja precisión. Por otro lado, si se tratan las

dos direcciones de transmisión como enlaces independientes, la varianza que se obtiene es significativamente mayor en comparación con la experimentada en condiciones reales. Por esta razón, se debe utilizar un modelo que sea capaz de devolver un **average path loss** para ambas direcciones de un enlace, y luego sumar y restar una variable aleatoria gaussiana de media cero y con desviación estándar bidireccional. De esta manera, se integra una segunda desviación estándar, obteniendo una manera más limpia e independiente para controlar la correlación entre las dos direcciones del enlace.

2.2.3. Modelamiento en capa de enlace y capa de red

En una red de área local (LAN), todos los elementos de la red comparten un medio de transmisión: un cable. Las redes inalámbricas, naturalmente no usan cables, pero incluso con WLAN todos los dispositivos involucrados envían y reciben datos utilizando un solo medio de transmisión, en un determinado rango de radio. En este sentido, WiFi (IEEE 802.11) es comparable a las primeras redes Ethernet semidúplex. Por lo tanto, también debe haber un protocolo para una red inalámbrica que regule el uso del medio.

La regla más importante en situaciones de comunicación conformada por múltiples participantes, es que solo una persona puede enviar su información a la vez. Si todos hablan al mismo tiempo, es confuso y nadie puede entender la información transmitida. En telecomunicaciones, los paquetes de datos también pueden superponerse: esto se denomina colisión, cuando los paquetes de datos se encuentran y alteran sus contenidos.

En este contexto, se presentan los protocolos asociados a la capa MAC (Medium Access Control, subcapa perteneciente a la capa de enlace y capa de red), los cuales intentan reducir la frecuencia de estas colisiones, junto con proporcionar un plan sobre cómo proceder si se produce una colisión. Estos protocolos adquieren importancia al considerar que las transmisiones en las redes inalámbricas no pueden ejecutarse en el mismo orden como lo harían con un cable. Así, en una red descentralizada, es necesario que todos los participantes sigan un conjunto de reglas y organicen la comunicación entre ellos.

Las diferencias técnicas entre las redes cableadas e inalámbricas también conducen al llamado problema de la estación oculta. Las estaciones en una red inalámbrica tienen un rango limitado, por lo que podría ser que los nodos de una red no sean capaces de reconocerse entre sí. De esta manera, es posible, pero no improbable, que dos estaciones no se reconozcan, y procuren alcanzar simultáneamente una estación que se encuentre entre ellas. Las transmisiones pueden superponerse en el nodo receptor, lo que significa que los datos se pierden.

2.2.3.1. Protocolo MAC: CSMA/CA

El protocolo CSMA (Carrier Sense Multiple Access) es un método básico que controla la comunicación de múltiples participantes en un medio de transmisión compartido y descentralizado. En este protocolo, el transmisor intenta determinar si existe otra transmisión en curso antes de iniciar una transmisión utilizando un mecanismo de detección de portadora,

vale decir, intenta detectar la presencia de una señal portadora de otro nodo antes de intentar transmitir. Si se detecta una portadora, el nodo espera a que finalice la transmisión en curso antes de iniciar su propia transmisión. De esta manera, usando CSMA, múltiples nodos pueden, a su vez, enviar y recibir en el mismo medio.

El buen desempeño de CSMA ha permitido el desarrollo de tres variantes: CSMA/CA (en redes inalámbricas), CSMA/CD (Ethernet), y CSMA/CR (en redes CAN). Enfocándose en redes inalámbricas, CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) intenta garantizar que dos o más participantes no inicien una transmisión al mismo tiempo para evitar colisiones. Si se produce una superposición, se detectará y la transmisión se intentará de nuevo.

La idea básica detrás de CSMA/CA es el principio "Listen Before Talk" (LBT), esto significa que se debe revisar la línea para ver si está libre antes de que la estación pueda iniciar una transmisión. Otras funciones dentro del procedimiento aseguran que las colisiones pueden evitarse en gran medida. Una descripción exhaustiva del presente protocolo se puede encontrar en el capítulo *Apéndice A, A.2 Protocolo MAC: CSMA/CA*.

2.2.3.2. Protocolo MAC: ALOHA

En los años setenta, el sistema ALOHA fue propuesto como una solución eficaz para proporcionar acceso inalámbrico a los sistemas informáticos. La red ALOHA de la Universidad de Hawái empleó transmisores fijos en islas ubicadas a distancias de varias decenas de kilómetros. La principal ventaja del esquema de acceso aleatorio ALOHA fue la simplicidad. Los terminales pueden transmitir sus datos independientemente de la actividad de otros terminales. Si un mensaje es exitoso, la estación base envía un ACK a través de un canal de retroalimentación. Si el terminal no recibió un ACK, el terminal retransmite el mensaje después de esperar un tiempo aleatorio. El retraso se determina principalmente por la probabilidad de que no se reciba un paquete (debido a la interferencia de otra transmisión, llamada "colisión") y el valor promedio del tiempo de espera aleatorio antes de que se realice una retransmisión.

2.2.3.2.1 Resolución de colisiones

La simplicidad de ALOHA tiene su contra: para una población infinita de usuarios y bajo ciertas condiciones de canal, el sistema es inestable. Los paquetes perdidos en una colisión se retransmiten, pero la retransmisión nuevamente experimenta una colisión. Esto puede desencadenar una avalancha de intentos de retransmisión. Casi con seguridad, el backlog (número de paquetes que anteriormente no tuvieron éxito y que deben retransmitirse) crece más allá de cualquier límite finito. Un método para mitigar la inestabilidad es adaptar dinámicamente los tiempos de espera aleatorios de todos los terminales si la estación base nota que se producen muchas colisiones.

2.2.3.2.2 ALOHA en redes móviles

ALOHA es muy utilizado en sistemas de comunicación inalámbrica, pero el rendimiento difiere de lo que uno esperaría en una red alámbrica.

En un canal de radio, los paquetes pueden perderse debido al desvanecimiento de la señal, incluso si no hay otra señal contendiente presente. Por otro lado, los paquetes pueden recibirse con éxito a pesar de la interferencia de los terminales de la competencia; esto se llama *receiver capture*. Este efecto tiene una influencia significativa en el rendimiento de la red.

La reutilización de frecuencia óptima para las redes de acceso aleatorio ALOHA difiere de la reutilización de frecuencia para telefonía, debido a que los criterios de rendimiento difieren. El mejor patrón de reutilización para un sistema ALOHA es usar la misma frecuencia en todas las celdas.

2.2.3.2.3 Pure ALOHA (Unslotted ALOHA)

El conjunto de reglas bajo el cual se rige el protocolo Pure ALOHA en la capa MAC se describe a continuación, visualizándose en la Figura 2.4.

- Un usuario transmite cuando tiene paquetes para transmitir.
- Cuando dos o más transmisiones de paquetes se superponen en el tiempo, se produce una colisión y se destruyen todos los paquetes involucrados en la colisión (*non-capture*).
- Si el ACK no se recibió dentro del tiempo de espera, entonces el nodo elige un tiempo aleatorio de inactividad (para evitar colisiones repetidas).
- El nodo retransmite el paquete después del tiempo de inactividad.

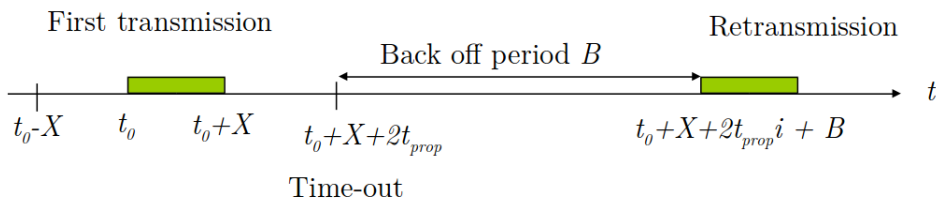


Figura 2.4: Protocolo ALOHA [16].

2.2.3.2.4 Slotted ALOHA

A fin de mejorar el desempeño de Unslotted ALOHA se definió una nueva variante el protocolo: Slotted ALOHA. En Slotted ALOHA se obliga al nodo a transmitir en forma sincronizada. Todos los nodos de la red realizan un seguimiento de los intervalos de transmisión y pueden iniciar la transmisión solo al comienzo de un intervalo de tiempo (el eje de tiempo se divide en intervalos de tiempo con una duración igual al tiempo para transmitir un paquete), disminuyendo el número de colisiones en comparación a Unslotted ALOHA. Una ilustración de este protocolo se presenta en la Figura 2.5

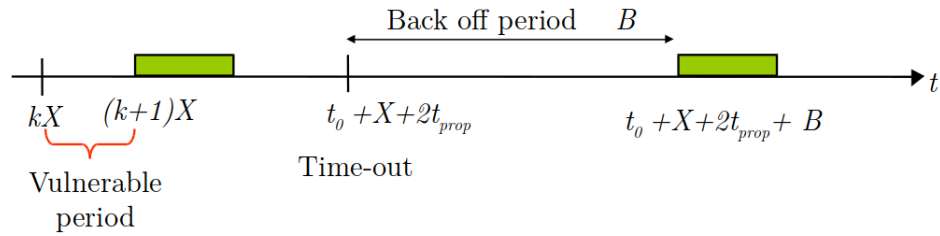


Figura 2.5: Protocolo Slotted ALOHA [16].

2.2.4. Modelamiento en capa de transporte

La capa de transporte, tanto en el modelo OSI como en el modelo TCP/IP, garantiza que los mensajes transmitidos se entregan sin errores, de manera secuencial, sin presentar pérdidas o duplicaciones de la información, realizando un control de tráfico de los mensajes y multiplexando las sesiones de transmisión respectivas.

Asociado a esta capa se disponen de determinados protocolos para la transmisión de mensajes. La definición de protocolo establece que este corresponde a “*un conjunto de reglas que se establecen en el proceso de comunicación entre dos sistemas*”. Establecer un conjunto de reglas permite regularizar la forma en el proceso de comunicación será establecido en distintos lugares del mundo, y adquiere una mayor relevancia al ver la interconectividad presente en la actualidad. Bajo este contexto, a continuación, se presentan dos de los protocolos en la capa de transporte más importantes en los sistemas comunicacionales.

2.2.4.1. Protocolo TCP

TCP corresponde a un protocolo de comunicación orientado a la conexión entre máquinas, el cual permite que estas se comuniquen entre sí y controlen el estado de la transmisión de la información, garantizando el envío sin errores de los datos presentes. Este protocolo es uno de los más conocidos, estandarizados y utilizados en el ámbito de las telecomunicaciones, puesto que monitorea el flujo de datos, los multiplexa cuando provienen desde distintas fuentes (para que así puedan ser transmitidos simultáneamente), y trabaja con ellos en el orden en el cual fueron enviados. Una descripción exhaustiva del presente protocolo se puede encontrar en el capítulo *Apéndice A, A.3 Protocolo TCP*.

2.2.4.2. Protocolo UDP

UDP, o Protocolo de Datagrama de Usuario, corresponde a un protocolo de comunicación de nivel de transporte basado en el intercambio de datagramas (paquetes de datos caracterizados por tener una cabecera y la información a transmitir). Este protocolo permite el envío de estos paquetes a través de la red sin requerir el establecimiento previo de una conexión entre las máquinas, proporcionando una interfaz sencilla entre la capa de red y la de aplicación. Ahora bien, este protocolo no detecta los errores asociados a la transmisión ni tampoco

controla los flujos de datos, estableciendo así que este protocolo no está orientado a la conexión, al no garantizar la entrega de los mensajes transmitidos. Una descripción exhaustiva del presente protocolo se puede encontrar en el capítulo *Apéndice A, A.4 Protocolo UDP*.

2.2.5. Diseño de redes de ordenadores

Considerando una aplicación que distribuye un archivo de gran tamaño desde un único origen a un gran número de hosts, es importante disponer de un modelo de red que sea capaz de operar bajo estas condiciones de red. Una descripción de dos modelos de interés (*Cliente-Servidor* y *P2P*) se puede encontrar en el capítulo *Apéndice A, A.5 Diseño de redes de ordenadores*.

2.3. Internet of Things

Internet of Things (IoT) es una extensión de la computación hacia objetos de uso común. Este paradigma permite que los objetos actúen como Access Points a servicios en la red mediante su control remoto. La base de IoT, así como de muchas redes privadas industriales, son las redes Machine-To-Machine (M2M), las que están conformadas por elementos sensores, actuadores y gateways, así como las redes de comunicaciones que los interconectan. En su arquitectura se distinguen tres áreas funcionales bien distinguidas: el área o dominio de red, el área o dominio de dispositivos sensores y actuadores, y el dominio del gateway. Estas redes son un foco importante del proceso de estandarización para las futuras redes móviles de quinta generación o 5G, existiendo una fuerte perspectiva en la industria de que dichas redes sean la base fundamental para comunicaciones IoT en un futuro cercano. Desafíos importantes a explorar son el despliegue de comunicaciones IoT en zonas de difícil acceso (a través de satélites, por ejemplo), el consumo energético, y la sustentabilidad del ciclo de vida de los dispositivos.

En los procesos propios de la IoT pueden distinguirse cuatro etapas, las cuales no necesariamente están presentes de manera clara en todas las situaciones.

2.3.1. Capacidades relacionadas con dispositivos IoT

IoT tiene sus bases en los procesos de automatización industrial comenzados décadas atrás. El aumento de capacidades, de la ubicuidad y de las prestaciones de las redes de comunicaciones han hecho posible en los últimos años el desarrollo de soluciones de control a distancia empleando las redes públicas de acceso a Internet, con características de seguridad y baja latencia, que las hacen atractivas.

El abaratamiento de la electrónica asociada a sensores, actuadores y controladores, así como el surgimiento de plataformas con facilidades para el desarrollo (SDK) y APIs estandarizadas que posibilitan la interoperabilidad entre elementos diversos ha dado impulso al

concepto de redes de elementos conectados entre sí, para posibilitar una amplia variedad de aplicaciones a nivel doméstico, corporativo, industrial, medioambiental, regional, nacional y global.

Un listado no taxativo de capacidades relacionadas con los dispositivos IoT se aborda a continuación.

- **Comunicación y cooperación entre dispositivos:** los dispositivos IoT deben ser capaces de comunicarse entre sí, con las plataformas de control con los gestionan, y cooperar para lograr los objetivos de diseño de las soluciones de las que forman parte.
- **Direccionabilidad e identificación (IPv6):** los dispositivos IoT deben ser capaces de ser encuestados por las plataformas de control o por procesos remotos, para lo cual necesitan una identificación dentro de la red (dirección de Internet o dirección IP) y la red debe tener la capacidad de acceder a ellos a través de procesos de enrutamiento y sistemas de seguridad adecuados.
- **Sensores:** estos dispositivos tienen como finalidad detectar cambios en su entorno, y producir una salida correspondiente. La cantidad de variables que pueden ser medidas es enorme, por lo que la variedad de dispositivos sensores también es grande. Así, adquieren un rol relevante en el diseño de soluciones basadas en tecnologías de IT.
- **Actuadores:** son todos aquellos dispositivos que, ante la recepción de un comando, realizan una acción determinada. De manera similar a los sensores, los actuadores realizan una multiplicidad de funciones, dependiendo de su aplicación, tecnología, entre otros factores.
- **Procesamiento de información embebido:** con el objetivo de limitar el uso de las redes públicas para la transmisión de información no relevante, los dispositivos en la IoT deben ser capaces de discriminar y procesar la información localmente, a un nivel básico, dada la necesidad de mantener los costos y el consumo energético acotados.
- **Localización geográfica:** IoT necesita tener sentido de la situación geográfica de sus dispositivos (sistemas de alerta temprana de incendios forestales, la monitorización de la calidad del aire, etc), por lo que deberá contar con mecanismos eficientes de localización geográfica de sus dispositivos, con un nivel de precisión que variará acorde a las necesidades de la aplicación en cuestión.
- **Interfaces hacia los usuarios humanos:** en muchos casos los dispositivos necesitarán interfaces físicas o virtuales para su interacción con los humanos, de los que pueden recibir información (sensores, como pantallas touch, interruptores, botones, aplicaciones para smartphones) o a los que pueden presentarla (actuadores tales como pantallas, parlantes, alarmas, etc).

2.3.2. Aplicaciones de tecnologías IoT para las Smart Cities

Entre las aplicaciones de tecnologías IoT para las Smart Cities se encuentran:

- **Soluciones IoT en base a wearables:** servicios de e-health, localización de mascotas, chalecos y cascos con videocámaras para los operarios de túneles y obras críticas en las ciudades, basados en wearables y provistos directamente por operadores LTE Advanced.
- **Servicios de proximidad:** servicios de anuncios de seguridad, de tráfico o de marketing basados en la proximidad del cliente objetivo, servicios de comunicaciones directas entre dispositivos cercanos, servicios relacionados con la realidad aumentada.
- **Comunicaciones V2V:** comunicaciones públicas entre vehículos para propósitos de seguridad, información de gestión de tráfico y servicios de valor agregado hacia los vehículos.
- **Control crítico en latencia:** podría aplicarse para sistemas de seguridad vehicular, control de vuelo de drones en las ciudades, transmisión de alarmas, entre otros. Las aplicaciones de Realidad Aumentada para procesos industriales y para turismo se ven potenciadas por la baja latencia y gran ancho de LTE Advanced.
- **Backhaul para transmisión de imágenes:** capturadas por drones y vehículos en movimiento. El procesamiento de las imágenes se ve potenciado por las técnicas de BDA y el Cloud Computing. Los requerimientos de latencia y de ancho de banda potencian a LTE Advanced en este tipo de aplicaciones.

2.3.3. Arquitectura IoT

Las redes de IoT se basan en el concepto de objetos inteligentes que realizan funciones y brindan nuevos servicios conectados. Estos objetos son "inteligentes" porque utilizan una combinación de información contextual para realizar acciones. Estas acciones pueden ser auto-contenidas (es decir, el objeto inteligente no depende de sistemas externos para sus acciones); sin embargo, en la mayoría de los casos, estos deben interactuar con un sistema externo para transmitir la información que ha sido recopilada o interactuar con una plataforma de administración. En este caso, la plataforma de administración se puede utilizar para procesar los datos recopilados desde el dispositivo inteligente. De un punto de vista arquitectónico, varios componentes tienen que trabajar juntos para que una red IoT sea operativa:

- **Capa de objetos:** en esta capa, los dispositivos físicos deben adaptarse a las limitaciones del entorno en el cual han sido implementados, debiendo asimismo ser capaces de proporcionar la información necesaria.
- **Capa de red de comunicaciones:** cuando los objetos inteligentes no son independientes, deben comunicarse con un sistema externo. En general, esta comunicación utiliza una tecnología inalámbrica. Esta capa tiene cuatro subcapas:
 - **Subcapa de acceso al medio:** esta capa se compone típicamente de tecnologías inalámbricas, tales como IEEE 802.11ah, IEEE 802.15.4g, y LoRa.
 - **Subcapa de gateways & backhaul:** es común que un sistema de comunicación IoT organice múltiples objetos inteligentes alrededor de un gateway común. El

gateway se comunica directamente con los objetos inteligentes, reenviando la información recolectada a través de un medio de mayor alcance (llamado **backhaul**) a una estación central donde la información es procesada. Este intercambio de información es una función de la capa de aplicación, que es el motivo por el cual este objeto se denomina gateway. En redes IP, este gateway también reenvía paquetes de una red IP a otra, y por lo tanto actúa como un enrutador.

- **Subcapa de transporte:** para que la comunicación tenga éxito, los protocolos de red y transporte (como IP y UDP) deben ser implementados de manera de soportar la conexión de múltiples dispositivos.
- **Subcapa de gestión IoT:** se debe autorizar la utilización de protocolos adicionales que permitan el intercambio de datos desde la capa de aplicación con los sensores. Ejemplos de estos protocolos son CoAP y MQTT.
- **Capa de aplicación y análisis:** en la capa superior, una aplicación debe procesar los datos recopilados para tomar una decisión inteligente basada en la información recopilada y, a su vez, instruir a los dispositivos a adaptarse a las condiciones analizadas.

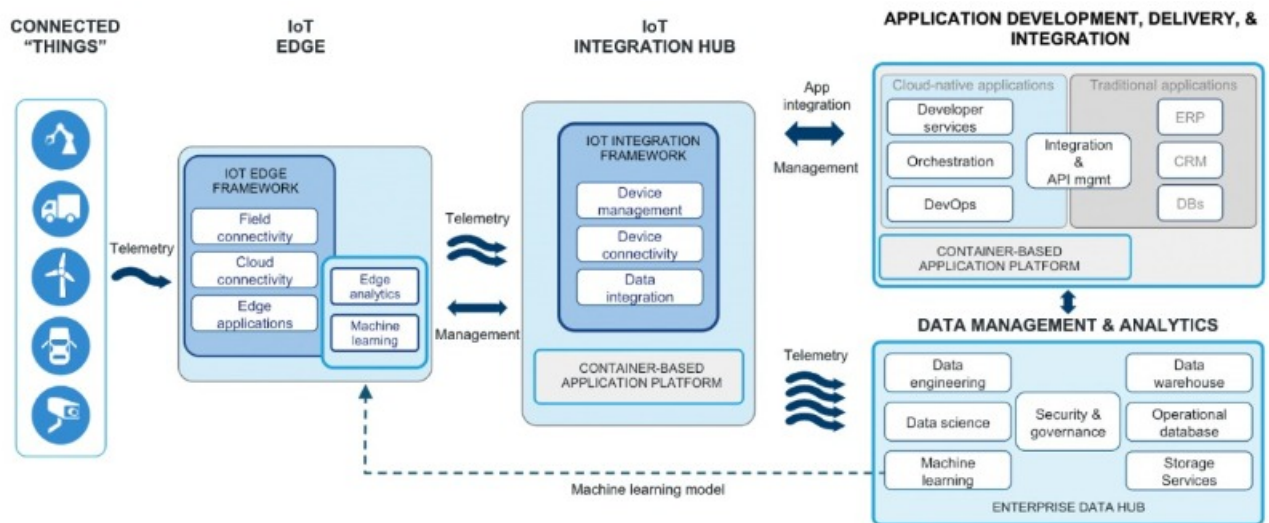


Figura 2.6: Diagrama de arquitectura general IoT [9].

2.3.4. Tecnologías

2.3.4.1. NB-IoT

Es la primera tecnología centrada en conectar a Internet objetos cotidianos que requieren pequeñas cantidades de datos en períodos de tiempo largos. Es una de las distintas tecnologías Low-Power Wide-Area Networking (LPWAN). Esta tecnología ha sido desarrollada para permitir comunicaciones eficientes y una alta durabilidad de la batería, para dispositivos distribuidos masivamente. Utilizado frecuentemente en ambientes rurales.

2.3.4.2. LoRa

LoRa es una tecnología de comunicación inalámbrica digital de datos en sistemas IoT, desarrollada y patentada por Semtech, y que entre sus principales ventajas ofrece una alta tolerancia a las interferencias, alta sensibilidad para recibir datos, bajo consumo energético (hasta 10 años con una batería), baja transferencia de datos (hasta 255 bytes), conexión punto a punto, y múltiples frecuencias de trabajo: 915[MHz] en América, 868[MHz] en Europa, y 433[MHz] en Asia. Al tratarse de una tecnología LPWAN, en ambientes rurales presenta un radio de cobertura de 10[km] a 20[km], y en ambientes urbanos de 1[km] a 5[km] [17].

2.3.4.3. ZigBee

Es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica y de bajo consumo, basada en el estándar IEEE 802.15.4 de redes WPAN. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías. Utilizado frecuentemente en ambientes urbanos, con radio de cobertura de hasta 100[m].

2.3.4.4. Protocolos de enlace de datos en IoT

2.3.4.4.1 IEEE 802.15.4

IEEE 802.15.4 es un estándar que se desarrolló para proporcionar un framework y las capas más bajas en el modelo OSI para redes de conectividad inalámbrica de bajo costo y bajo consumo.

IEEE 802.15.4 proporciona las capas MAC y PHY, dejando que las capas superiores se desarrollen para estándares más específicos, como *Thread*, *ZigBee*, *6LoWPAN* y muchos otros.

Como resultado, IEEE 802.15.4 no toma el protagonismo de la manera en que otros estándares pueden hacerlo, pero a pesar de ello, constituye la base de un gran número de estándares y, por lo tanto, se implementa de manera más amplia de lo que puede ser aparente a primera vista.

La baja potencia es uno de los elementos clave de IEEE 802.15.4, ya que se utiliza en muchas áreas donde los sensores remotos necesitan funcionar con energía de la batería, posiblemente durante años sin atención.

Aspectos básicos El estándar IEEE 802.15.4 tiene como objetivo proporcionar las capas de red inferiores esenciales para una red de área personal inalámbrica, WPAN. Los requisitos principales son la comunicación ubicua de bajo costo y baja velocidad entre dispositivos.

El concepto de IEEE 802.15.4 es proporcionar comunicaciones a distancias de hasta 10 metros y con velocidades de transferencia de datos máximas de 250 [Kbps]. Anticipando que la reducción de costos requerirá soluciones de dispositivos altamente integrados, el concepto general de IEEE 802.15.4 se ha ideado para adaptarse a esto.

IEEE 802.15.4 MAC layer El propósito de la capa MAC IEEE 802.15.4 es proporcionar una interfaz entre la capa física y la capa de aplicación. El IEEE 802.15.4 no especifica una capa de aplicación, siendo este generalmente un sistema de aplicación como *ZigBee*, *RF4CE*, *MiWi*, entre otros.

La capa MAC del estándar IEEE 802.15.4 realiza las siguientes funciones:

- Transfiere datos a la capa física y viceversa; transfiere datos a la capa de red y viceversa.
- En el coordinador, ofrece una ranura de tiempo garantizada (GTS) opcional para cada dispositivo que accede a la red.
- Genera un beacon frame en el coordinador.
- Apoya la seguridad del dispositivo.
- Proporciona CSMA/CA como método de acceso para la red.
- Proporciona una conexión confiable entre dos capas MAC mediante el uso de un ACK.

Una descripción exhaustiva del presente protocolo se puede encontrar en el capítulo *Apéndice A, A.6.1 IEEE 802.15.4*.

2.3.4.4.2 LoRaWAN

Long-Range Wide Area Network (LoRaWAN) es una tecnología inalámbrica emergente diseñada para redes WAN de baja potencia con bajo costo, movilidad, seguridad y comunicación bidireccional para aplicaciones de IoT. Es un protocolo optimizado de bajo consumo diseñado para redes inalámbricas escalables con millones de dispositivos. Es compatible con operaciones redundantes, sin ubicación, de bajo costo, de bajo consumo y de energía tecnologías de recolección para satisfacer las necesidades futuras de IoT al tiempo que permite la movilidad y la facilidad de uso de las funciones [4].

2.3.4.4.3 IEEE 802.11ah

Una descripción exhaustiva del presente protocolo se puede encontrar en el capítulo *Apéndice A, A.6.2 IEEE 802.11ah*.

2.4. Wireless Sensor Network

Las redes de sensores inalámbricos (WSN) desempeñan un papel fundamental en Internet of Things (IoT). Son la interfaz del sistema IoT con el mundo físico, recogiendo información de los terminales para su posterior transmisión al sistema. Cada nodo de la red es responsable de detectar y recoger ciertas propiedades físicas, enviar y recibir información hacia y desde otros nodos en la red, y agregar información de interés. La información recogida se envía a un gateway, y posteriormente a Internet.

En la última década, el costo de los sensores, actuadores, microprocesadores y las propias infraestructuras de las redes se han reducido rápidamente, haciendo de WSN una realidad práctica. Muchas de las aplicaciones que antiguamente presentaban costos elevados se encuentran actualmente al alcance, a un costo razonable. Referido a la red en sí, en muchos casos los nodos de los sensores se despliegan en áreas remotas o de difícil acceso, alimentándose energéticamente a partir del uso de baterías, y comunicándose de forma inalámbrica. Debido a esto, los nodos asociados a los sensores necesitan consumir muy poca energía, y ser capaces de autoorganizarse en el caso de fallo de algún nodo de la red.

De esta manera, en la presente sección se describe las características de la WSN, analizando las restricciones energéticas, el funcionamiento de la red, el sistema operativo asociado, y el paradigma informático resultante de estas características.

2.4.1. Restricciones de consumo de energía

Los sensores se pueden alimentar a través de la conexión de los mismos a baterías [15][25]. Las fuentes de alimentación conectadas son constantes, pero pueden ser poco prácticas o costosas en aplicaciones remotas o lugares de difícil acceso. Las baterías representan una alternativa viable, pero la vida útil de la batería, la carga y el reemplazo representan problemas importantes que no se pueden ignorar [8]. Hay dos aspectos importantes:

- **Eficiencia:** los sensores modernos son muy eficientes en términos de consumo de energía, debido al avance de las tecnologías de silicio. En algunos casos, los sensores pueden permanecer operativos con baterías durante más de 10 años [8].
- **Fuente:** la mayoría de las veces, los nodos de los sensores se despliegan en un lugar remoto o en lugares de difícil acceso. La disponibilidad de energía es limitada y la sustitución de la batería es difícil. Bajo esta condición, el consumo de energía de los nodos sensores debe estar restringido. Si bien los sensores a menudo dependen de las baterías, la recolección de energía de varias fuentes de energía puede proporcionar algunas alternativas [14]. Sin embargo, las tecnologías de captación de energía actualmente disponibles son caras, y las empresas dudan en realizar la inversión dada la poca confiabilidad asociada con el suministro de energía alternativa [14].

Debido a esta restricción energética, el sistema operativo de los nodos sensores, la organización de la red y el protocolo utilizado en la comunicación de red deben ser diseñados o construidos de manera de restringir el consumo energético.

2.4.2. Seguridad en los nodos sensores

La seguridad es definitivamente una característica clave cuando se considera la implementación de IoT [15][26]. Dentro del sistema de IoT, la integridad y la autenticidad de los datos son preocupaciones de seguridad importantes en el nivel del nodo del sensor. Sin embargo, debido a limitaciones como la baja potencia de procesamiento, la capacidad de memoria y la disponibilidad de potencia a nivel del sensor, son preferibles los protocolos de comunicación ligeros [12]. Como resultado, esto podría impedir el uso de algoritmos complejos criptográficos para la integridad de los datos.

Uno de estos protocolos ligeros es el Protocolo de Aplicación Restringida (Constrained Application Protocol - CoAP) [28], el cual busca aplicar el mismo paradigma de transferencia de aplicaciones y las características básicas de HTTP a las redes restringidas, manteniendo un diseño simple y una sobrecarga baja. Si bien CoAP es adecuado para sistemas de sensores con limitaciones de energía, no incluye funciones de seguridad integradas. CoAP es vulnerable a los ataques habituales de Internet, incluidos los ataques de redes externas, como la denegación de servicio distribuido (DDoS). Estos ataques no solo traen nuevos problemas de seguridad, sino que también podrían agotar la batería de nodos restringidos. El problema no es la falta de soluciones de seguridad disponibles, sino que las implementaciones a menudo no las admiten, ya que se prefiere la ligereza a la seguridad [15][2][18]. De esta manera, se necesita más investigación sobre las medidas de seguridad preventivas en los nodos sensores.

2.4.3. Interoperabilidad

Muchos sistemas de sensores están diseñados para aplicaciones específicas y, cuando están interconectados, generan problemas de interoperabilidad en áreas de comunicación, intercambio, almacenamiento y seguridad de datos, y escalabilidad [15]. Para resolver este problema, se necesitan protocolos de comunicación para habilitar las comunicaciones entre sistemas de sensores heterogéneos.

Sin embargo, los protocolos de comunicación convencionales, como HTTP, no son adecuados para redes de sensores debido a las limitaciones de los sistemas de sensores, como la disponibilidad de energía limitada, la pequeña cantidad de memoria disponible y la capacidad de procesamiento limitada. Como tal, son preferibles los protocolos de comunicación ligeros, como CoAP, el cual utiliza el mismo modelo REST [15] que HTTP. Desde el punto de vista de un desarrollador, CoAP es muy parecido a HTTP, y eso facilita el proceso de integración. Dado que tanto HTTP como CoAP comparten el mismo modelo REST, se pueden conectar utilizando proxies de protocolo cruzado independientes de la aplicación.

2.4.4. Computación distribuida

En aplicaciones de IoT se pueden identificar cuatro componentes principales: sensores, comunicaciones, computación y servicio. Se recopila y analiza gran cantidad de datos para revelar información y crear conocimiento. Para admitir tareas de cálculo y tamaño de datos

a gran escala, no es factible enviar cada bit de datos a servidores en la nube centralizados. Una solución viable es distribuir el cálculo a cada nodo en IoT [27].

Una solución es lo que HP denominó *Distributed Mesh Computing* y lo que Cisco denominó *Fog Computing*. La computación en malla distribuida consiste en pequeños servidores "en el borde", servidores de tamaño mediano y almacenamiento en el medio que almacenan datos, y servidores grandes en una ubicación central donde se llevan a cabo los análisis pesados y el almacenamiento a largo plazo.

La computación local puede determinar que algunos datos deben transmitirse a la nube para un análisis más detallado, pero en este escenario no es necesario que todos los datos vayan y vengan a la nube para su procesamiento.

Todos estos servidores y nodos trabajarán juntos en un sistema de computación de malla distribuida para atender el crecimiento masivo de datos y dispositivos conectados que se espera en un sistema IoT.

Los aspectos clave de la computación en malla distribuida son:

- **Distribuir, calcular y almacenar en todas partes:** cada nodo tiene la potencia de cálculo y el almacenamiento local. De esta manera, los datos y el procesamiento están cerca de donde se encuentra el evento, lo que promueve la eficiencia.
- **Mover los datos y/o análisis según sea necesario:** los datos recopilados localmente se pueden agregar de forma diferente y enviar a otros nodos según sea necesario. Cada dispositivo es de bajo costo, con cierta potencia informática. Si se requiere de un análisis adicional, este se puede realizar en un nivel "superior".
- **Asumir condiciones intermitentes de la red:** la conectividad de la red fallará. Sin embargo, con la computación en malla distribuida, la red se puede reorganizar y los datos se pueden enrutar de manera diferente al nivel.
- **Requerimiento de bajo consumo energético:** dado que cada nodo sensor requiere poca energía y menor espacio, el costo del sistema se reduce.

2.4.5. Redes auto-organizadas

En IoT, los nodos y los sensores dependen de las redes para comunicarse y transmitir la información de interés, sin embargo, esta infraestructura está sujeta a deterioro, desastres y otras condiciones adversas. Con el fin de mitigar la situación que hace que la infraestructura de red original sea menos efectiva o inútil, se necesita la capacidad de autoorganizarse entre estos dispositivos para mantener la capacidad de comunicación [6].

Cuando IoT es observado desde una perspectiva de sistema, todos los dispositivos contemplados en el paradigma IoT forman parte de una red heterogénea. Esta heterogeneidad incluye la capacidad de computación del dispositivo, las tecnologías de red y comunicación, y los servicios ofrecidos por varios dispositivos en IoT. Por lo tanto, las operaciones de comunicación de estos dispositivos deben ser tratadas como parte de un solo sistema.

Las propiedades clave que son críticas para la autoorganización eficiente en IoT incluyen un modelo de comunicación cooperativo, conocimiento de la situación y balanceo de carga automatizado. Con la perspectiva de estas propiedades, se pueden identificar cinco componentes clave de la autoorganización en IoT:

- Descubrimiento de nodos vecinos.
- Control de acceso medio (MAC).
- Conectividad local y establecimiento de rutas.
- Gestión de la recuperación del servicio.
- Gestión de la energía.

En cuanto a los desafíos propios de las redes auto-organizadas en IoT, estos son:

- Diseño cross-layer para auto-organización.
- Heterogeneidad para la auto-organización.
- Comunicaciones multicanal y multiradio.
- Computación de baja potencia y balanceo de carga.
- Redes tolerantes al retraso en redes auto-organizadas

2.4.6. Modelo de capas

El modelo de capas más común para WSN sigue el modelo OSI. Básicamente, en las redes de sensores, se necesitan de cinco capas: capa de aplicación, capa de transporte, capa de red, capa de enlace de datos y capa física. A las cinco capas se agregan tres planos cruzados, tal como se muestra en la Figura 2.7.

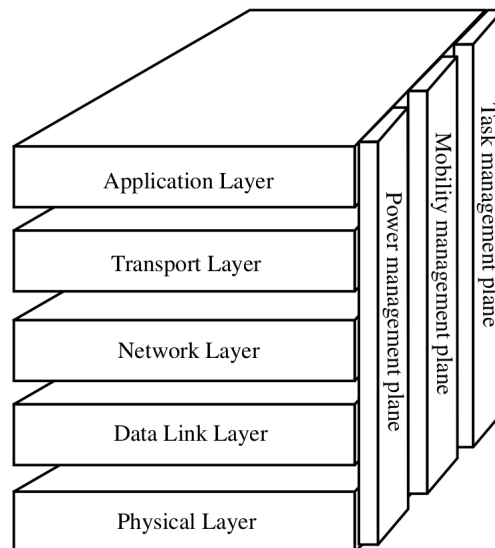


Figura 2.7: Wireless Sensor Network: Modelo de capas [3].

Los tres planos transversales corresponden al plano de administración de energía, el plano

de gestión de movilidad y el plano de administración de tareas. Estas capas se utilizan para gestionar la red y hacer que los sensores trabajen en conjunto para aumentar la eficiencia general de la red.

En cuanto a la diferencia de arquitecturas entre los modelos OSI, TCP/IP WLAN y WSN se muestra en la Tabla 2.1.

Modelo OSI	Modelo TCP/IP	Modelo WLAN	Modelo WSN
Aplicación	Aplicación	Programa de aplicación	Aplicación WSN
Presentación		Middleware	Middleware WSN
Sesión		Socket API	
Transporte	Transporte	TCP/UDP	Protocolos de transporte WSN
Red	Internet	IP	Protocolos de ruteo WSN
Enlace de datos	Acceso a la red	Adaptador WLAN	Control de error
		Protocolos MAC WLAN	Protocolos MAC WSN
Física		Transceiver	Transceiver

Tabla 2.1: Comparativa de modelos OSI, TCP/IP, WLAN y WSN [3].

2.4.7. Protocolos MAC en redes WSN

El protocolo MAC desempeña un papel importante en las redes WSN, ya que el consumo de batería se puede controlar fácilmente si se diseña un buen protocolo MAC. Sin embargo, hay algunos atributos más aparte de la eficiencia energética que se necesitan para ser considerados: se necesita una utilización eficiente de la energía para aumentar la vida útil de la red, saber cómo reacciona la red de sensores a los cambios en el tamaño de la red y cómo se adapta a las dinámicas cambiantes del entorno.

2.4.7.1. Sensor-MAC (S-MAC)

S-MAC es un protocolo MAC que regula los períodos de hibernación en una red de sensores para ahorrar energía y mejorar vida útil de la red. Este protocolo se basa en upon **sleep** y **active time** para lograr la eficiencia energética en las redes WSN. S-MAC usa la programación de hibernación estática para reducir el consumo de energía y dividir el tiempo en **frames**. Cada **frame** se divide en un período activo y un período de suspensión. En el período activo, el transmisor-receptor se enciende y se apaga durante el período de espera. Un ciclo completo de escucha y período de hibernación se llama un **frame**.

Este protocolo reduce el tiempo inactivo de escucha al permitir que los nodos entren en modo de suspensión periódica. Cada nodo se duerme durante un tiempo y luego se despierta y escucha para ver si algún otro nodo quiere hablar con él. Todos los nodos son libres de elegir sus propios horarios de escucha/suspensión. Ahora bien, para reducir la sobrecarga de control, los nodos vecinos se sincronizan juntos. Los nodos intercambian sus horarios mediante la transmisión periódica de un paquete SYNC a sus vecinos inmediatos. Esto garantiza que

todos los nodos vecinos puedan comunicarse entre sí, incluso si tienen diferentes horarios. El período para que cada nodo envíe un paquete SYNC se denomina período de sincronización.

El protocolo S-MAC sigue procedimientos similares al protocolo IEEE 802.11 para evitar colisiones, empleando RTS/CTS, sin embargo, el problema con S-MAC es que incluso si el intercambio de datos finaliza dentro del período activo, el nodo permanecerá activo hasta que el tiempo de inactividad desperdicie energía. Por lo tanto, el `static duty cycle` del protocolo S-MAC provoca un desperdicio de energía.

2.4.7.2. Timeout-MAC (T-MAC)

El problema del `static duty cycle` del protocolo S-MAC se resuelve mediante el protocolo T-MAC, en el cual se utiliza un `dynamic duty cycle`. El protocolo T-MAC introduce un mecanismo de tiempo de espera activo que disminuye la sobrecarga de escucha inactiva, ajustando dinámicamente el período activo de acuerdo con las cargas de tráfico de la red.

2.5. Marco regulatorio: Internet of Things para Smart Cities

En Chile, la Subsecretaría de Telecomunicaciones (Subtel), la Superintendencia de Electricidad y Combustibles (SEC) y el Instituto Nacional de Normalización (INN) deberán ser los principales organismos que regulen la implementación y uso de servicios y dispositivos relacionados con la IoT. No obstante, la regulación en esta materia aún está en sus primeras fases en el país, lo que debe ser solucionado para fomentar el despliegue y masificación de soluciones IoT. La Subtel deberá regular fundamentalmente el uso de espectro para la transmisión inalámbrica de comunicaciones entre dispositivos, así como los límites de emisión permitidos para los mismos, mientras que la SEC deberá establecer los mecanismos para la certificación de los productos empleados para la IoT.



Figura 2.8: Logos institucionales: (a) Subsecretaría de Telecomunicaciones, (b) Superintendencia de Electricidad y Combustibles, (c) Instituto Nacional de Normalización.

En marzo de 2017 fue firmado por la Presidenta de la República un proyecto de ley para la protección de datos personales, que modifica la actual normativa vigente (Ley 19.628), que data de 1998. La nueva ley creará una nueva institucionalidad para la protección de, entre otros, datos biométricos y de geolocalización, claves para muchas soluciones en la IoT.

El INN deberá normalizar las interfaces, protocolos de comunicación y requerimientos de interoperabilidad de dispositivos y plataformas para su uso en la IoT, entre otras cuestiones relevantes. En 2017 culminó la adaptación de una serie de herramientas normativas para la normalización de la telegestión del alumbrado público, en el marco del Proyecto de Bienes Públicos Estratégicos para la Competitividad “Marco normativo necesario para la implementación de la telegestión de alumbrado público”, en el que el INN es el beneficiario y la AChEE, el Ministerio de Energía y la SEC, mandantes.

Este proyecto considera la adopción de un conjunto base de Normas Internacionales ISO/IEC, UNE-EN y ANSI sobre Tecnologías de la Información y Protocolos de Red de Control, requisitos generales y particulares de luminarias e iluminación en carreteras, con el fin de especificar un procedimiento de comunicación para redes locales de control de área, requisitos generales para la luminarias que incorporen fuentes de luz eléctrica con tensiones de alimentación hasta 1000[V] y requisitos fotométricos para iluminación en carreteras, dirigido a las necesidades visuales de los usuarios. También considera la elaboración un cuerpo normativo compuesto por ocho normas chilenas, que recogerán los resultados de la experiencia y del desarrollo tecnológico, establecido en normativas internacionales y/o extranjeras, determinadas en base a lo establecido por ISO (International Organization for Standardization) en su comité *ISO/IEC JTC 1/SC 6 Telecommunications and Information Exchange Between Systems* y lo dispuesto en IEC (international Electrotechnical Commission) en su comité *TC34/SC34D Luminaires*.

La ISO ha generado un elevado número de estándares que guardan relación con el uso de TICAR en el ámbito de las Smart Cities. Varios comités y subcomités técnicos están involucrados en la estandarización de diversos aspectos relativos a dicho uso. En el caso de las normas sobre el uso de TICAR en las Smart Cities y sus diferentes ámbitos, existen 5 comités técnicos fundamentalmente que generan estándares relevantes: *ISO/TC 22 - Road vehicles*, *ISO/TC 268 - Sustainable Cities and Communities* y su subcomité *SC 1 - Smart Community Infrastructures*, *ISO/TC 241 - Road Traffic Safety Management Systems*, *ISO/TC 215 - Health Informatics*, e *ISO/TC 204 - Intelligent Transport Systems*.

2.6. Infraestructura

2.6.1. Iluminación Pública: eficiencia energética

Las transformaciones urbanas que están viviendo las distintas ciudades del mundo se presentan como oportunidades significativas para abordar problemáticas tales como la administración y uso de la energía, elaboración de visión frente a las emisiones de gases, entre otros. De esta manera, se debe asegurar que las ciudades desarrollen una metodología en la cual exista un consumo eficiente de la energía, contribuyendo a la reducción de las restricciones de suministro de energía de los sistemas actuales y a la seguridad energética nacional.

Por otro lado, el crecimiento y la expansión de las ciudades conlleva a un incremento acelerado de la utilización de recursos energéticos, esto debido a la implementación de sistemas

de alumbrado público, sistemas de bombeos de edificios, entre otros. Dado que estos recursos deben estar constantemente disponibles para los sistemas anteriormente mencionados, se provoca un bloqueo del consumo de energía por un largo período de tiempo. Es en este contexto en que tecnologías de iluminación con eficiencia energética (tales como los LEDs) se presentan como una de las oportunidades más rentables para ahorrar energía, reducir los costos asociados a los diversos municipios, y reducir las emisiones GHG asociadas al sistema de luminarias.

Bajo la línea anterior, los sistemas de iluminación pública son los servicios que presentan mayores oportunidades en el ahorro de energía, debido a su escala e impacto público. Diversos estudios demuestran que el uso de infraestructuras con iluminación eficiente puede reducir entre un 25 % y un 60 % los niveles de energía asociados. Es debido a este potencial de ahorro energético que existen múltiples implementaciones de estos modelos, generando asociaciones públicas-privadas para la inversión en estos servicios, siendo usados por las municipalidades y ciudades en sus programas de alumbrado público. Esta estructura de negocio ha demostrado ser exitosa, por sobre todo en su relación con las municipalidades correspondientes, presentando un alto rendimiento en mercados desarrollados tales como Francia, Alemania, Reino Unido y Estados Unidos, y de manera similar en economías emergentes tales como Chile, Brasil Tailandia, Corea del Sur y México.

En todo el mundo, los enfoques de gestión de la demanda municipal proporcionan a la ciudad una oportunidad de mejorar sus servicios y operaciones, conservando electricidad y reduciendo las emisiones de gases GHG. Por lo tanto, resulta importante aumentar la penetración de tecnologías con eficiencia energética que permitan llevar a una rápida urbanización de las ciudades, y por ende de las economías emergentes. Ahora bien, el éxito de la integración y escalamiento de las tecnologías de alumbrado público EE (eficiencia energética) depende de la superación de múltiples barreras, entre ellos barreras financieras, barreras de información y conciencia, barreras institucionales y regulaciones, y barreras técnicas, las cuales son detalladas en el capítulo *Apéndice A, A.7 Infraestructura*.

2.6.2. Iluminación Pública: estudios económicos

Tal como fue mencionado en la subsección anterior, las transformaciones urbanas que están viviendo las distintas ciudades se presentan como oportunidades significativas para abordar problemáticas tales como la administración y uso de la energía, entre los más destacables. En el contexto del presente trabajo de título, aún cuando pueda existir un modelo apropiado y una planificación acorde para la integración y escalamiento de las tecnologías de alumbrado público que aborden dichas problemáticas, resulta fundamental que ambos se ciñan a los diversos estudios económicos que determinan la factibilidad de implementación de las tecnologías presentes.

Al realizar una intensa revisión bibliográfica, no es menor la cantidad de estudios económicos que se pueden encontrar y que estén ligados a la integración de nuevas tecnologías bajo el enfoque de reducción de uso energético y sus respectivos impactos en la sociedad. Bajo esta gamma de estudios, un informe que sobresale es [24], el cual se presenta como un caso de negocios propuesto al Consejo Municipal de Stockport, ubicado en Manchester,

Inglaterra, para la implementación de luminarias LED en el sistema de alumbrado público. En este documento, se propone emprender una inversión de capital al municipio, teniendo bajo consideración que las diversas tecnologías asociadas a iluminación han percibido un crecimiento significativo desde el año 2010 hasta el año 2015 (año de publicación del documento), comprobándose a través de la realización de múltiples ensayos en Reino Unido que la integración de luminarias LED ofrece la oportunidad de reducir las emisiones de carbono de los sistemas de iluminación tradicionales y asimismo lograr eficiencias operacionales, por medio de la reducción del consumo eléctrico y la disminución de los costos de mantenimiento. Por otro lado, sujeto a las adquisiciones de los equipos correspondientes y los precios actuales del mercado eléctrico, en este mismo estudio se determina que una inversión de capital de esta envergadura logrará ahorros económicos considerables, los cuales pueden ser destinados a múltiples proyectos de diversa índole ligados al municipio. De esta manera, invertir en el capital se presenta como mejor opción a largo plazo por sobre "*no hacer nada*", ya que en este último escenario se deben enfrentar los costos crecientes de energía, los costos crecientes de mantenimientos de infraestructura, y el aumento progresivo de las tecnologías de iluminación tradicionales.

Ahora bien, ¿por qué un municipio como Stockport debiese realizar un cambio en sistemas de alumbrado público? Son múltiples las razones que llevan a plantearse esta interrogante, y que puede ser extrapolable a diversas ciudades a nivel mundial.

En primer lugar, existe una preocupación a nivel mundial en cuanto a la contribución de las emisiones de material particulado al cambio climático, lo cual se ve plasmado en múltiples jurisdicciones que se elaboran a fin de abordar estas problemáticas. Particularmente, en el año 2008 se legisló en Reino Unido una ley enfocada a la reducción del 80% de los gases de efecto invernadero para el año 2050 [20], lo cual conllevó a la elaboración de presupuestos gubernamentales basados en la reducción de estas emisiones, obligando de esta manera a que las diversas ciudades pertenecientes a este bloque adoptaran políticas con dicho enfoque.

En segundo lugar, es conocido que los sistemas de alumbrado público tradicionales conllevan un gasto económico significativo, como es el caso de Stockport, en el cual a lo menos el 54% de los gastos está asociado a consumo eléctrico por parte de las luminarias públicas [24]. Teniendo bajo consideración que la infraestructura de los sistemas de alumbrado público tradicional se vuelve obsoleta e ineficiente, esto implica un mayor consumo y un mayor costo de energía, en conjunto a los costos de mantenimiento. De esta manera, la decisión de mantener estas infraestructuras implica realizar un mantenimiento continuo y exhausto al sistema de alumbrado público, elevando los ya altos costos de mantenimiento y operación.

En tercer y último lugar, la introducción de tecnologías modernas tales como los sistemas de iluminación LED, que presentan un bajo consumo de energía, están destinadas a reemplazar a las tecnologías de iluminación tradicionales presentes en los múltiples sistemas públicos, gracias a la reducción de los costos energéticos como la disminución de emisión de huellas de carbono, en conjunto a la disminución de los costos asociados a la mantención y operación del sistema de alumbrado público.

Teniendo bajo consideración los puntos anteriormente tratados, ¿cuáles serían los objetivos principales y los beneficios obtenidos en un proyecto de esta envergadura, bajo una mirada económica?

2.6.2.1. Objetivos principales

Los objetivos principales para la integración de nuevas tecnologías en el sistema de alumbrado público (teniendo como referencia el caso de negocio del municipio de Stockport [24]) serían:

- Reducción del consumo de energía del sistema de alumbrado público, reducción de los costos de energía asociados, y la minimización de la exposición de este sistema a aumentos futuros del precio de la energía.
- Reducción de las emisiones de carbono en bases a las jurisdicciones internacionales.
- Disminuir la realización continua de mantenimiento al sistema de alumbrado público.
- Disponer de un sistema sostenible a largo plazo, tanto bajo una mirada económica como una mirada medioambiental.

2.6.2.2. Beneficios

La integración de nuevas tecnologías al sistema de alumbrado público puede proporcionar diversos beneficios comunitarios, tales como:

- Mejorar la prestación del servicio del sistema de alumbrado público, por medio de la mejora de la calidad de la iluminación y la reducción de las interrupciones lumínicas.
- Disponer de un sistema de seguridad con televigilancia más efectivo durante las noches.
- Reducción al crimen callejero, gracias a las mejoras presentes en los sistemas de iluminación.
- Promover el deporte en período nocturno.

2.6.3. Iluminación Pública: impactos en la salud

Si bien son múltiples los beneficios que puede traer la integración de nuevas tecnologías al sistema de alumbrado público, es importante tener en consideración los efectos de los mismos en la salud de las personas. Considerando nuevamente el caso de negocio propuesto para el municipio de Stockport en Inglaterra [24], el sistema de salud público elaboró un informe sobre los impactos potenciales en la sociedad en cuanto a la introducción de luminarias LED que operan a una temperatura de 4000 K [23]. ¿Por qué realizar un informe de estas características? Puesto que una evaluación de impacto en la salud contribuye a garantizar a la sociedad que el bienestar de las personas se encuentra debidamente considerados por todos los entes presentes. Este estudio es capaz de determinar tanto los impactos positivos como negativos de la propuesta tecnológica realizada. De esta manera, el objetivo principal es identificar los principales impactos de la propuesta en estudio, en conjunto con la discusión rápida sobre las mejores formas de tratar con ellos, a fin de maximizar los beneficios y evitar cualquier impacto adverso.

En este contexto, realizando múltiples casos de estudio bajo la utilización de una luminaria

LED que opera a 4000 K [23], los test no muestran que exista evidencia alguna en cuanto a un impacto directo negativo en la integración de luces LED al sistema de alumbrado público. Si bien existen resultados que dejan en manifiesto que frente a una exposición prolongada a luz artificial durante el período nocturno puede interferir en la producción de melatonina, aumentando la probabilidad de adquirir cáncer u obesidad, la muestra estadística no es suficiente como para establecer que este impacto sea generalizado. Ahora bien, si es un factor a tener en consideración, puesto que sugiere la existencia de un riesgo plausible para la salud mediante el uso de sistemas de iluminación con tecnología LED que operan a 4000 K. Una solución podría ser la utilización de LED que operen a menor temperatura, sin embargo bajo la mirada económica sería inviable, dado los altos costos implicados. De esta manera, para minimizar cualquier riesgo potencial que afecte la salud, diversos estudios proponen de manera preventiva que la iluminación se encuentre dirigida de manera vertical, a fin de dispersarse lo menos posible a otras direcciones, en sectores residenciales las unidades de iluminación estén instaladas con blindaje a su alrededor para minimizar el ingreso de las mismas a los hogares, y finalmente diseñar las luminarias de tal manera que estas no afecten el campo de visión de un conductor.

2.7. Antecedentes metodológicos

2.7.1. Software de simulación en redes de telecomunicaciones

2.7.1.1. OMNeT++

OMNeT ++ es un framework de simulación de eventos discretos escrito en el lenguaje de programación *C++*, enfocado principalmente para la elaboración de simuladores de redes de comunicación cableadas e inalámbricas, redes en chip, redes de colas, entre otros. La funcionalidad que entrega para la simulación de redes de sensores, redes ad-hoc inalámbricas, protocolos de Internet, modelado de rendimiento, redes fotónicas, entre otros, es proporcionada por modelos desarrollados como proyectos independientes. OMNeT ++ ofrece un IDE basado en Eclipse, un entorno de ejecución gráfico y una gran cantidad de otras herramientas. A su vez, incluye extensiones para simulación en tiempo real, emulación de red, integración de bases de datos, junto a múltiples funcionalidades adicionales.

Entre los componentes principales que conforman OMNet++ se encuentran las librerías de simulación inteligente, un compilador para topología de descripción de lenguaje NED (el cual define los módulos simples y complejos, los parámetros presentes y los puertos de entrada/salida), definición de campos de los mensajes y los tipos de datos asociado (archivo *.msg*), una interfaz gráfica para la ejecución del simulador, una interfaz al usuario de las líneas de comando ejecutadas durante la simulación, herramientas de visualización escalar y vectorial de gráficas de los resultados, y una herramienta de documentación de los modelos simulados, Además, dispone de una interfaz gráfica llamada *Tkenv*, la cual presenta animaciones de las ejecuciones de las simulaciones (como seguimiento de ejecución de protocolos, ver *Figura 2.10*), ventanas de inspección para examinar o cambiar variables del modelo, despliegue

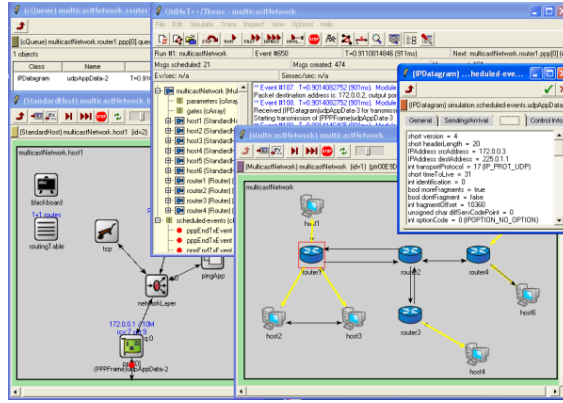


Figura 2.9: Ambiente de Simulador OMNeT++.

gráfico de los resultados de la simulación para cada una de las capas de red, entre los más destacable.

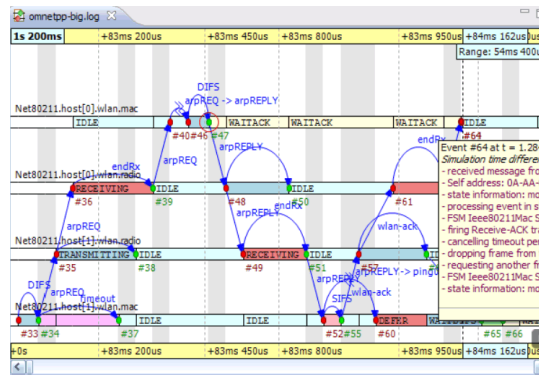


Figura 2.10: Seguimiento de secuencias en análisis de protocolo, en simulador OMNeT++.

Uno de los grandes desarrollos presentes en este framework de simulación es la integración de otros frameworks *open source*, tales como *Veins* que está orientado a la simulación de comunicaciones inter-vehiculares, *INETMANET* framework de INET que incluye una serie de características y protocolos experimentales, principalmente para redes ad-hoc móviles, *ANSAINET* proyecto dedicado al desarrollo de modelos de simulación compatibles con las especificaciones de la RFC, y *Castalia* que está dedicado a la simulación de redes de sensores inalámbricos (WSN), redes de área corporal (BAN) y redes de dispositivos integrados de baja potencia. Finalmente, el desempeño en la simulación de estos sistemas puede ser visualizado en 3D, lo cual aporta mayor información en cuanto a patrones de radiación, enlaces de comunicación, interferencias, entre otros, tal como se puede visualizar en la *Figura 2.11*.

2.7.2. Softwares de simulación en entornos IoT

2.7.2.1. Castalia

Castalia es un simulador para redes de sensores inalámbricos (WSN), redes de área del cuerpo (BAN) y, en general, redes de dispositivos integrados de bajo consumo [19]. Se basa en

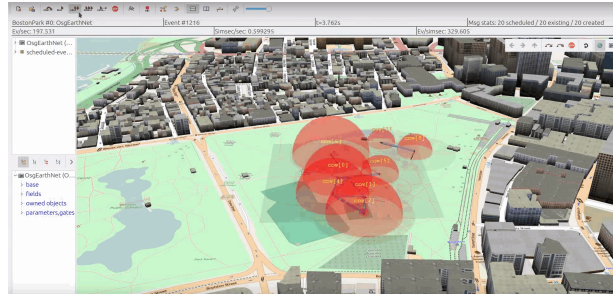


Figura 2.11: Visualización 3D de desempeño de redes en simulador OMNeT++.

la plataforma OMNeT ++ y puede ser utilizada por investigadores y desarrolladores que quieran probar sus algoritmos y/o protocolos distribuidos en modelos de radio y canales inalámbricos realistas. Castalia también se puede utilizar para evaluar diferentes características de la plataforma para aplicaciones específicas, ya que es altamente paramétrica y puede simular una amplia gama de plataformas. Las principales características de Castalia son:

- **Modelo de canal avanzado basado en datos empíricamente medidos.**
 - El modelo define un mapa de pérdida de ruta, no simplemente conexiones entre nodos.
 - Modelo complejo para la variación temporal de la pérdida de trayectoria.
 - Totalmente compatible con la movilidad de los nodos.
 - La interferencia se maneja como la intensidad de la señal recibida, no como una característica separada.
- **Modelo de radio avanzado basado en radios reales para comunicación de baja potencia.**
 - Probabilidad de recepción basada en SINR, tamaño de paquete, tipo de modulación. Compatible con PSK, FSK, modulación personalizada permitida al definir la curva SNR-BER.
 - Se permiten múltiples niveles de potencia de TX con variaciones de nodo individuales.
 - Estados con diferentes consumos de energía y retrasos en el cambio entre ellos.
 - Detección de portadora, basada en sondeo y en interrupciones.
- **Disposiciones de modelado de detección extendida**
 - Modelo de proceso físico altamente flexible.
 - Sensor de ruido, polarización y consumo de energía del dispositivo.
 - Desplazamiento del reloj del nodo, consumo de energía de la CPU.
- **Disponibles la utilización de protocolos MAC y de enrutamiento**

Castalia se diseñó desde el principio para que los usuarios puedan implementar/importar fácilmente sus algoritmos y protocolos, a la vez que hacen uso de las funciones que ofrece el simulador. La modularización adecuada y un procedimiento de construcción automatizado y configurable ayudan a este fin.

2.7.2.2. FLoRa

FLoRa (Framework for LoRa) es un marco de simulación para llevar a cabo simulaciones de extremo a extremo en redes LoRa [22]. Se basa en el simulador de red `OMNeT++` y también utiliza componentes del marco `INET`. FLoRa permite la creación de redes LoRa con módulos para nodos LoRa, gateways y un servidor de red. La lógica de la aplicación se puede implementar como módulos independientes que están conectados con el servidor de red. El servidor de red y los nodos admiten la gestión dinámica de los parámetros de configuración a través de la velocidad de datos adaptativa (Adaptive Data Rate - ADR). Finalmente, las estadísticas de consumo de energía se recogen en cada nodo. Las principales características de FLoRa son:

- Modelo preciso de la capa física LoRa.
- Simulaciones con múltiples gateways en la red.
- Simulaciones end-to-end, incluyendo un modelo preciso de la red backhaul.
- Estadísticas de consumo de energía en red.

Capítulo 3

Metodología

3.1. Descripción de trabajo de memoria

El objetivo general del presente trabajo de memoria consiste en modelar una infraestructura TICAR habilitante para las Smart Cities, con foco a las redes de telecomunicaciones. Para esto, se propone como infraestructura habilitante una red de sensores en luminarias inteligentes, caracterizadas por presentar tanto sistemas de autogestión como sistemas de inteligencia, interconexión digital que se enmarca en el paradigma IoT. De esta manera, existe un enfoque de optimización en los procesos operativos asociados al funcionamiento de las luminarias públicas, en conjunto con la adición de inteligencia a la red.

3.2. Metodología para el desarrollo de modelos

El enfoque principal de la red de sensores propuesta en luminarias inteligentes, como infraestructura habilitante en calles inteligentes de las Smart Cities, es en telecomunicaciones. Para esto, debe existir en primera instancia un modelo que cumpla con los criterios mínimos impuestos por la IETF a través de la RFC, en cada una de las capas del modelo OSI: capa física, capa de enlace de datos, capa de red, capa de transporte, capa de sesión, capa de presentación y capa de aplicación. Para lograr esto, el modelamiento de la red de sensores debe incluir el análisis de parámetros tales como banda de frecuencia de operación, *data rate* asociado, *symbols rate*, exponente de backoff, protocolo de enrutamiento, protocolos de transporte, aplicaciones y servicios. Configurados los escenarios correspondientes, debe existir un análisis y estudio de sensibilidad de las variables de la red, tales como velocidad, latencia, jitter, throughput, pérdida de paquetes, corrupción de paquetes, duplicación de paquetes, entre los más destacables. Además, como referencia en cuanto al modelamiento de la red de tal manera que cumpla con las diversas regulaciones estipuladas, estarán presentes modelos de entes internacionales, tales como modelos de OFCOM (Reino Unido), modelos de la FCC (Estados Unidos), modelos de la ARCEP (Francia), modelos del MIC (Japón), y modelos de la SUBTEL (Chile). Finalmente, el modelamiento debe cumplir con las norma-

tivas que están siendo desarrolladas para servicios IoT, en conjunto con la utilización de los protocolos correspondientes. De esta manera, la integración conjunta de cada uno de estos puntos permitirá el modelamiento de la red de sensores.

3.3. Metodología para el desarrollo de simulaciones

El desarrollo de modelos y simulaciones para el presente trabajo de memoria contempla tres escenarios. El primero se encuentra relacionado con la simulación de la red de sensores en redes de telecomunicaciones tradicionales, sobre las cuales se añadirán sistemas de autogestión e inteligencia, como servicios adicionales. Para esto, importante será la evaluación de las redes existentes, a fin de determinar el cumplimiento de estas bajo los criterios a definir. El segundo escenario se encuentra relacionado con la simulación de la red de sensores bajo una perspectiva de IoT, lo cual corresponde a incluir inteligencia sobre la red, y la conexión de la misma a servicios cloud. Para esto, importante será disponer de una red operativa sobre la cual se añadan dispositivos de hardware virtualizados, que permitan simular la red de sensores propuesta.

La simulación de los diversos escenarios propuestos debe ser realizado con softwares de alta precisión y desempeño, de manera de evaluar rigurosamente el modelo integral de la red de sensores a desarrollar. De esta manera, a continuación se presentan cada uno de las herramientas computacionales a disponer en cada uno de los escenarios previamente mencionados.

3.4. Planificación de trabajo

Para el desarrollo del presente trabajo de memoria y el cumplimiento del objetivo general del mismo, a continuación se presenta la planificación de trabajo propuesta, con las fechas estipuladas para su óptima realización.

- 1. Elaboración metodología.**
- 2. Modelamiento de infraestructura propuesta.**
- 3. 1ª Aplicación de metodología y obtención de resultados preliminares.**
- 4. Análisis y discusión de resultados preliminares.**
- 5. 2ª Aplicación de metodología y obtención de resultados finales.**
- 6. Análisis y discusión de resultados finales.**
- 7. Elaboración de conclusiones.**

3.5. Modelamiento

3.5.1. Comparativa entre tecnologías IoT

En el contexto de tecnologías habilitantes para IoT se dispone de múltiples variantes que responden a distintas necesidades y requerimientos, sustentados en redes WSN. Entre estas tecnologías podemos encontrar NB-IoT, LoRa y Zigbee.

Para el diseño de una infraestructura TICAR habilitante para Calles Inteligentes, desde el punto de vista de las telecomunicaciones, se requiere de una tecnología que ofrezca robustez y simplicidad en el diseño de la arquitectura y topología de red, un correcto manejo de colisiones en la capa MAC (garantizando un buen desempeño de la red), bajo consumo energético (propio de redes LPWAN), fácil integración a ordenadores de placa reducida (tales como Raspberry y Arduino), y que permita el despliegue masivo de una red de sensores en un radio de cobertura de a lo menos $1[km]$, en ambiente urbano. Las características anteriormente descritas señalan a LoRa como la tecnología habilitante en telecomunicaciones, en el diseño de la infraestructura TICAR. Ahora bien, ¿cómo validar que LoRa es la tecnología IoT óptima para este diseño?

Existen múltiples enfoques en telecomunicaciones que permitirían establecer la validez de la presente tecnología IoT como tecnología adecuada para el diseño de una infraestructura TICAR, desde el análisis propio de las características detalladas por el proveedor hasta el análisis específico capa por capa. Teniendo bajo consideración que las tecnologías previamente mencionadas se encuentran licenciadas principalmente en capa física y en capa de aplicación, aumenta la complejidad en el estudio comparativo que se procura realizar. Ahora bien, una de las características distintivas de cada tecnología yace en el protocolo utilizado en la capa MAC para el manejo de colisiones, aspecto de relevancia al considerar un escenario donde existe un despliegue masivo de sensores, y que potencialmente puede albergar una sobrecarga de tráfico en la red. Referido a los protocolos distintivos en capa MAC, Zigbee (y un conjunto adicional de otras tecnologías) se basan en el estándar IEEE 802.15.4 [5], mientras que LoRa hace uso de Slotted ALOHA [13]. Considerando que la utilización de diversos protocolos en la capa MAC incide en el desempeño de las tecnologías en las redes WSN, resulta válido realizar un estudio preliminar comparativo que permita analizar las diferencias que se presentan entre estos protocolos, y por ende en las tecnologías de interés.

Para realizar este estudio, resulta fundamental hacer uso de simuladores que permitan establecer comparaciones en los protocolos correspondientes. Actualmente existen dos frameworks para el desarrollo de soluciones IoT, con enfoque en telecomunicaciones, basados en el simulador de eventos discretos OMNeT++. El primero es Castalia, simulador enfocado en el análisis de dataframe en topologías IoT, que presenta robustos modelos de canal en la capa física, modelos de radio (para transmisión y recepción), y modelamiento de sensores (CPU power, Power consumption), disponiendo de los protocolos ZigBee MAC (Static GTS IEEE 802.15.4), Basic IEEE 802.15.4, T-MAC, S-MAC. Este framework está desarrollado sobre OMNeT++ 4.6. La segunda plataforma corresponde a FLoRa, framework enfocado en el diseño de topologías IoT para redes LoRa, incluyendo nodos, sensores, gateways, Network Server, entre otros. Este framework está desarrollado sobre OMNeT++ 5.2.

Los códigos utilizados e implementados en los frameworks Castalia y FLoRa se encuentran disponibles en <https://github.com/ibuguenoc/memoria-titulo>.

Definidas las plataformas de simulación, adquiere una gran relevancia el diseño del escenario de simulación sobre el cual se realizará el estudio comparativo de los protocolos MAC. Para esto, se considerará una red WSN simple: un nodo transmisor y un nodo receptor, separados a una distancia d , la cual varía desde los 10 metros hasta los 30 metros (tal como se puede visualizar en la Figura 3.1).

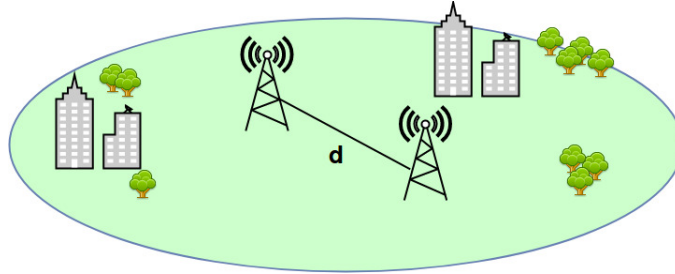


Figura 3.1: Escenario de simulación para comparación de tecnologías IoT.

Un aspecto importante en este escenario de simulación para el análisis comparativo de tecnologías IoT es el modelamiento en la capa física del medio inalámbrico. Para esto, se considera como modelo *average path loss*, con los siguientes parámetros de diseño en base a la Ecuación 2.1.

Parámetro	Valor
$P_L(d_0)$	55 [dB]
η	2.4
d_0	1.0 [km]
σ	4.0 [dB]
$\sigma_{bidireccional}$	1.0 [dB]

Tabla 3.1: Parámetros de diseño en modelo físico del canal.

Referido a la simulación de las condiciones de sobrecarga de tráfico en la red, como parámetros de diseño se considera un envío constante de paquetes desde el nodo transmisor al nodo receptor, con un **packet rate** de 5 paquetes por segundo. Además, se utilizarán grandes paquetes de tamaño de 2 [KB], de manera de producir colisiones. La utilización de estos paquetes requiere setear al máximo el tamaño de los paquetes y frames en cada capa del modelo de comunicación, tal como se presenta en la Tabla 3.2.

Parámetro	Valor
Maximum Network Frame Size	2500 [B]
Maximum MAC Packet Size	2500 [B]
Maximum PHY Frame Size	2500 [B]

Tabla 3.2: Parámetros de diseño en tamaño de paquetes y frames.

Finalmente, una vez realizado el diseño de los parámetros de la red, se procede a comparar los protocolos en capa MAC Static GTS IEEE 802.15.4 (Zigbee MAC), Slotted Aloha (LoRa MAC) y Timeout MAC (T-MAC), por medio de la ejecución del escenario diseñado en la plataforma de simulación Castalia (en un tiempo de simulación de 300 [s]), a fin de establecer una comparativa preliminar entre las tecnologías IoT.

3.5.2. Diseño de infraestructura TICAR habilitante

El objetivo general del presente trabajo consiste en el diseño de una infraestructura TICAR habilitante para las Smart Cities, con foco a las redes de telecomunicaciones. Para esto, se propone como infraestructura habilitante una red de sensores en luminarias inteligentes, caracterizadas por presentar tanto sistemas de autogestión como sistemas de inteligencia, interconexión digital que se enmarca en el paradigma IoT. De esta manera, existe un enfoque de optimización en los procesos operativos asociados al funcionamiento de las luminarias públicas, en conjunto con la adición de inteligencia a la red.

Para el diseño de una red de luminarias inteligentes, los diversos casos de estudio e implementación a nivel internacional establecen como parte del modelamiento las siguientes etapas:

- **Levantamiento de información.**
- **Desarrollo conceptual del diseño.**
- Desarrollo de cálculos de iluminación.
- Especificaciones técnicas de equipos de iluminación.
- Determinación de modelos de negocio.
- **Determinación de solución técnica para IoT.**
- **Verificación de disponibilidad de equipos.**
- Elaboración de dossiers de mantenimiento.
- Diseño preliminar de modelo de negocios.

Los puntos destacados corresponden a las aristas sobre las cuales el presente trabajo se centra.

3.5.2.1. Levantamiento de información

La red de sensores a desplegar en el sistema de alumbrado público (como parte de la red de luminarias inteligentes) será diseñada inicialmente para ser ubicada en avenidas y calles de la ciudad de Santiago de Chile. Referido a las tipologías de calles, en Chile no existe una regulación específica en cuanto al tamaño de una cuadra urbana, vale decir, al espacio lineal que abarca desde las dos esquinas formadas por la intersección de una calle con otra. El único precedente existente se remonta a 1848, donde el Congreso Nacional estableció que una cuadra urbana corresponde a 100 [m] [10]. Considerando el significativo desarrollo urbano a lo largo de las décadas siguientes, resulta factible observar que dicha medida de longitud no se condice

con el desarrollo urbano actual, medida que hoy en día es variable y sin regulación. Por este motivo, para establecer una medida de normalización, se realizaron diferentes mediciones de cuadras en la ciudad de Santiago, a través de la aplicación Google Earth, estableciendo que en promedio una cuadra equivale aproximadamente a 300 [m].

Por su parte, en cuanto a la disposición física de las luminarias públicas, la Superintendencia de Electricidad y Combustibles (SEC) establece que la distancia de separación entre dos postes del alumbrado público debe encontrarse entre 25 [m] a 35 [m], así como también es estipulado en diversas ordenanzas municipales [11].

En este contexto, para el presente trabajo se recrea un ambiente urbano, de una intersección, donde cada cuadra tiene una distancia $D = 300$ [m], y la separación entre los postes del alumbrado público es de $d = 25$ [m], tal como se puede visualizar en la Figura 3.2.

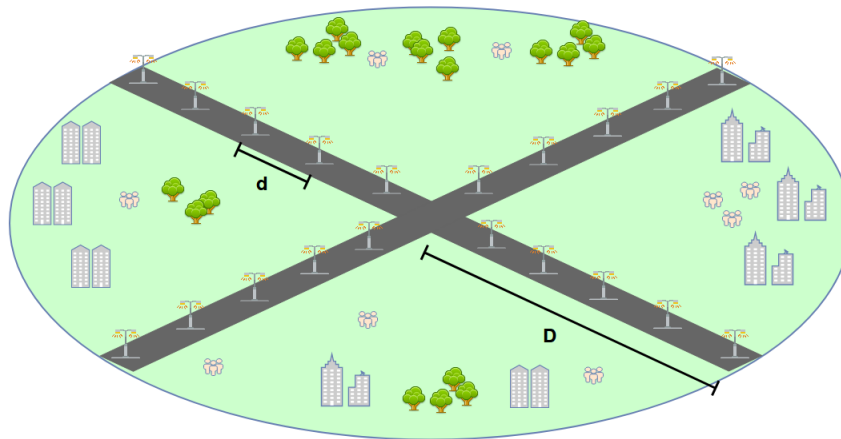


Figura 3.2: Recreación de ambiente urbano y sistema de iluminación pública.

3.5.2.2. Desarrollo conceptual del diseño

Referido al tipo de luminaria a utilizar, para la infraestructura habilitante propuesta se considera la utilización de luminarias vehiculares, con la integración de tecnologías de iluminación LED.

Al momento de seleccionar fuentes de iluminación LED, es importante disponer de información técnica documentada y respaldada por laboratorios competentes que avalen dicha información. Aspectos importantes a tener en consideración son eficacia, vida útil, consistencia del color, disipación térmica, corriente de alimentación y condiciones de fotometría (control óptico-lumínico, distribución lumínica, grado de protección IP, condiciones mecánicas y eléctricas en base a la normativa de la IEC).

3.5.2.3. Determinación de solución técnica para IoT

La infraestructura TICAR habilitante propuesta se encuentra conformada por dos aristas: por un lado se presenta la inteligencia de la red de sensores, y por otro el modelamiento con

enfoque en telecomunicaciones. En este contexto, en las siguientes subsecciones se realiza una descripción correspondiente de la solución técnica propuesta.

3.5.2.3.1 Revisión de la infraestructura propuesta desde el punto de vista de la inteligencia de sensores

Las luminarias inteligentes se constituyen como una infraestructura habilitante de comunicaciones preferida para generar soluciones en las Smart Cities, por un conjunto de características tales como su ubicuidad (presentes en toda la ciudad), su versatilidad para incorporar distintos tipos de tecnologías de comunicaciones, la integración de tecnologías IoT (WiFi, cámaras, sensores de luz y temperatura, alarmas sonoras, entre otros), y su acceso a la energía.

Referido a IoT, las soluciones existentes son aquellas en que un conjunto de sensores y actuadores emplean una red de comunicaciones para implementar aplicaciones, en las cuales la información recolectada es procesada para generar comportamiento inteligente de los sistemas, resultando en una mejora de los procesos.

Para integrar sensores y actuadores al alumbrado público, existen múltiples ordenadores de placa reducida sobre los cuales es factible realizar esta integración. Plataformas de desarrollo populares son Arduino y Raspberry Pi, caracterizados por ser unidades de procesamiento de bajo costo, de fácil integración con diversas tecnologías de comunicaciones, y de reducido tamaño. Un gran número de sensores pueden ser conectados a estas plataformas, lo que permite realizar un procesamiento y análisis local de datos desde diversas fuentes de información, caracterizando con mayor robustez las condiciones que rodean al sistema. A partir de este procesamiento se presentan dos escenarios: el primero consiste en la ejecución de alguna acción específica, por parte de alguno de los actuadores disponibles (gestión de nivel de iluminación, emisión de alarma sonora), y el segundo escenario consiste en la transmisión de dicha información a servidores de red dedicados, en los cuales se logre un monitoreo de la red y su entorno, permitiendo asimismo la telegestión de las luminarias.

A fin de visualizar el tamaño y versatilidad en la integración de las plataformas de desarrollo anteriormente mencionadas al sistema de alumbrado público, se presenta en la Figura 3.3 un case de protección para la plataforma Raspberry Pi.

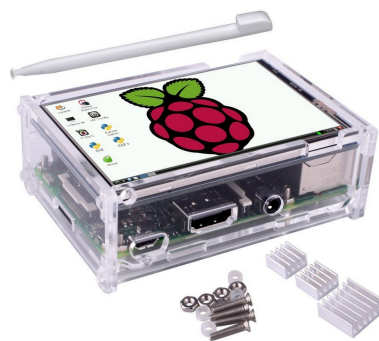


Figura 3.3: Case de protección para plataforma de desarrollo Raspberry Pi.

3.5.2.3.2 Modelamiento de la infraestructura propuesta desde el punto de vista de las telecomunicaciones

Una Smart City es una ciudad que utiliza las Tecnologías de la Información y Comunicación para mejorar la toma de decisiones, la eficiencia de las operaciones, y la prestación de servicios urbanos. El enfoque en los últimos años se ha centrado en tres grandes temas: aumento de capacidades para absorber la creciente demanda de servicios, aumento de calidad de experiencia (teniendo en cuenta throughput y latencia como indicadores fundamentales), y la convergencia de redes y servicios para lograr una conectividad universal. Así, adquieren un rol fundamental las redes inalámbricas, donde las tecnologías respectivas se han optimizado para su uso en IoT.

En este contexto, el diseño de la red de comunicaciones para la red de luminarias inteligentes debe ser riguroso, a fin de cumplir cada uno de los requerimientos expuestos a lo largo del presente documento, por lo cual es importante ceñirse a un plan de diseño, tal como se presenta a continuación.

Pre-Planning La primera etapa en el diseño de la red de comunicaciones considera la definición de los elementos de red, arquitectura y topología de red, y disposición física de la misma. Una red de luminarias inteligentes contempla como arquitectura principal una arquitectura WSN, basada en IoT. Esta arquitectura implica:

- **Capa de objetos:** en la cual los dispositivos físicos deben adaptarse a las limitaciones del entorno en el cual serán implementados, debiendo asimismo ser capaces de proporcionar información necesaria (sensores) y ejecutar acciones específicas (actuadores)
- **Capa de red de comunicaciones:** comunicación inalámbrica entre dispositivos de la red inteligente. Esta capa incluye una subcapa de acceso al medio, subcapa de gateways & backhaul, subcapa de transporte y subcapa de gestión.
- **Capa de aplicación y análisis:** procesamiento de datos recopilados para tomar una decisión inteligente. Se requiere de unidades de procesamiento local y externas (servidores dedicados).

Sobre cada uno de los postes del alumbrado público se dispondrá de un módulo de la red de luminaria inteligentes, el cual se encontrará conformado por una unidad de procesamiento Raspberry Pi, conjunto de sensores de interés, actuadores específicos (como control de nivel de luminosidad), y módulo transceiver asociado a las tecnologías de comunicación inalámbrica. La información generada en estos módulos será transmitida por el medio inalámbrico a un gateway dispuesto en la topología, el cual se encargará de enviar la información recibida al servidor dedicado, a fin de monitorear y telegestionar la red de luminarias públicas. Entre el gateway y el servidor dedicado se integran diversos elementos de la red, que permitan el correcto funcionamiento del sistema diseñado, otorgando seguridad a los datos y eficiencia en el desempeño de la red. Una ilustración de la arquitectura propuesta se presenta en la Figura 3.4.

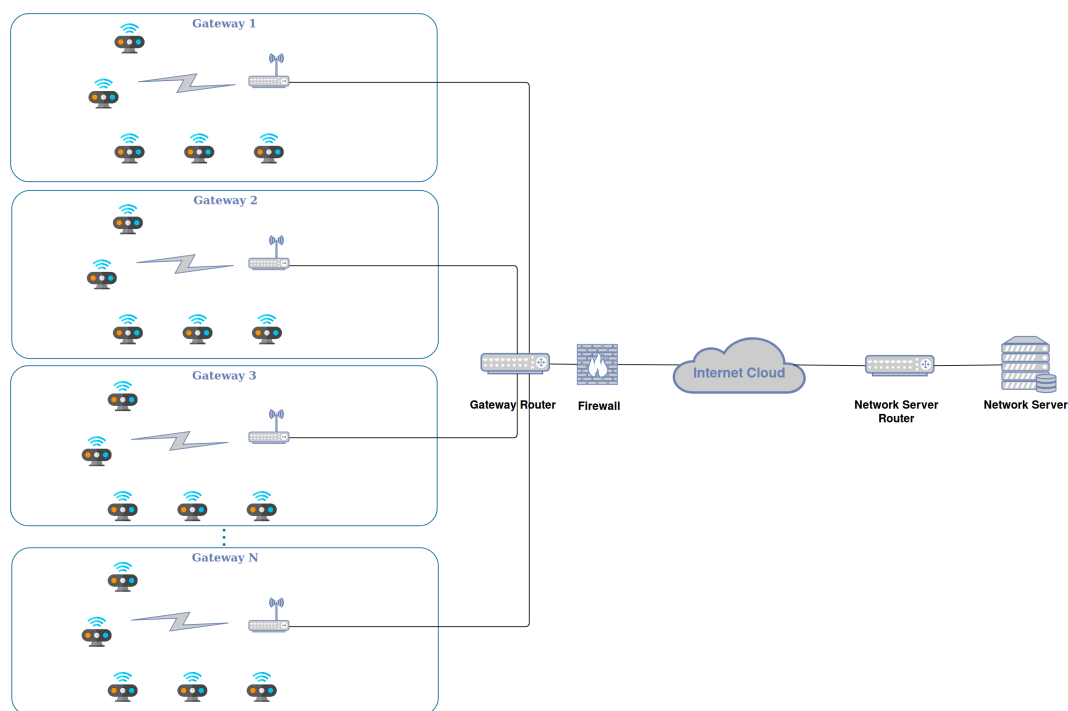


Figura 3.4: Luminarias Inteligentes: Arquitectura de red de sensores WSN, basado en IoT.

En cuanto a los elementos de la arquitectura de la red propuesta, estos son:

- Módulos de tecnologías de comunicación inalámbrica en los postes del alumbrado público, los cuales se encargan de transmitir la información sensada.
- Gateways, encargado de recibir la información transmitida por los módulos y enviarla al servidor dedicado.
- Gateway Router, proporcionando conectividad a nivel de red entre los gateways.
- Firewall, bloqueando el acceso no autorizado a la red.
- Internet Cloud, servicios de computación en Internet, donde se realiza procesamiento y análisis inteligente de los datos (tales como Amazon Machine Learning, Azure ML).
- Network Server Router, proporcionando conectividad a nivel de red entre Internet Cloud y los servidores dedicados.
- Network Server, servidor en el cual se almacenará y monitoreará la información recopilada y su análisis correspondiente.

Referido a la topología de red, adquiere un rol relevante la disposición física de los gateways, considerando que entre los módulos de comunicación de los postes del alumbrado público y los gateways se empleará una tecnología de comunicación inalámbrica, en un medio de propagación en el cual se presentan variables aleatorias que afectan la transmisión y recepción de los datos. De esta manera, la ubicación física de estos elementos influirá directamente en el desempeño de la red de sensores, debiendo existir un Line of Sight (línea de mira) adecuado entre el nodo emisor y el nodo receptor que permita el correcto funcionamiento del sistema de comunicación digital, a fin de disminuir la probabilidad de difracción, refracción, y/o reflexión de la señal en el medio. En este contexto, se presentan tres propuestas de topologías de red, tal como se logra visualizar en las Figuras 3.5, 3.6 y 3.7.

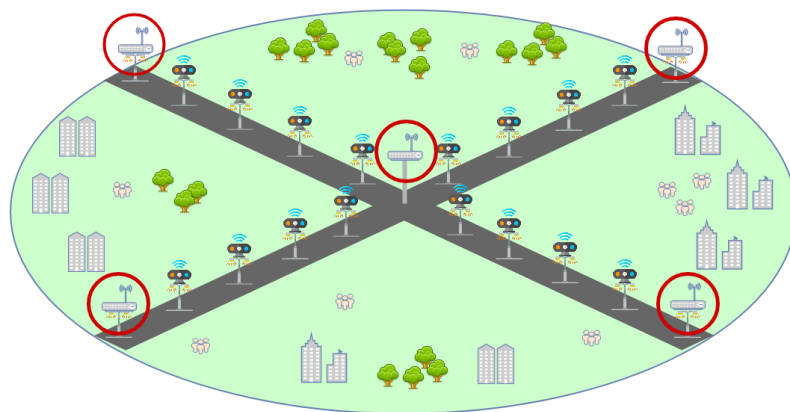


Figura 3.5: Luminarias Inteligentes: Diseño de red WSN - Topología A.

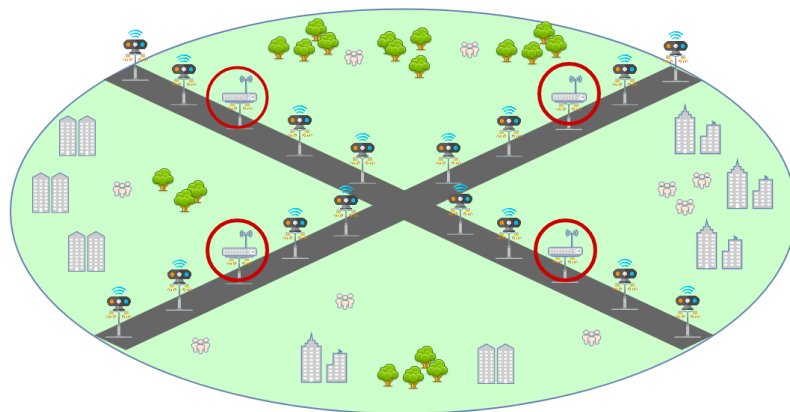


Figura 3.6: Luminarias Inteligentes: Diseño de red WSN - Topología B.

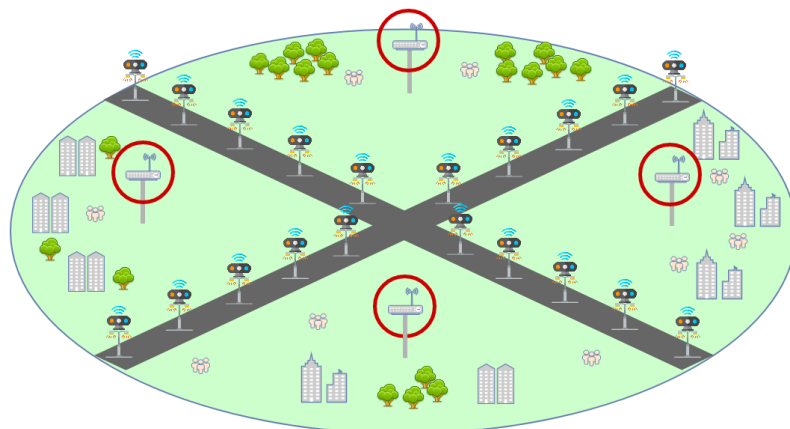


Figura 3.7: Luminarias Inteligentes: Diseño de red WSN - Topología C.

- **Topología A:** contempla la ubicación de los gateways de comunicación inalámbrica en las esquinas de intersección de las calles.
- **Topología B:** contempla la ubicación de los gateways de comunicación inalámbrica en los puntos medios de las calles.
- **Topología C:** contempla la ubicación de los gateways de comunicación inalámbrica en los puntos centrales de las cuadradas urbanas, requiriendo de una infraestructura adicional para su implementación.

Planificación detallada La segunda etapa en el diseño de la red de comunicaciones considera la configuración de la red, diseño de parámetros de configuración, y análisis y determinación de cobertura y capacidad.

Desde el punto de vista de las telecomunicaciones, el diseño de la infraestructura TICAR habilitante para Calles Inteligentes requiere de una tecnología que ofrezca simplicidad en el diseño de la arquitectura y topología de red, un correcto manejo de colisiones en la capa MAC, bajo consumo energético, fácil integración a ordenadores de placa reducida, y que soporte el despliegue masivo de una red de sensores en un radio de cobertura de a lo menos 1 km, en ambiente urbano. Basado en estas características, es elegida LoRa como la tecnología de comunicación principal de esta red.

LoRa es una tecnología de comunicación inalámbrica digital en sistemas IoT, que ofrece una alta tolerancia a las interferencias, alta sensibilidad para recibir datos, bajo consumo energético (hasta 10 años con una batería), baja transferencia de datos (hasta 255 bytes), conexión punto a punto, y múltiples frecuencias de operación (915[Mhz] en América). Al tratarse de una tecnología LPWAN, presenta un radio de cobertura en ambientes urbanos de 1[km] a 5[km] [17]. A nivel de capa MAC, su protocolo de acceso al medio se encuentra basado en ALOHA (principalmente en la variante Slotted ALOHA [13]).

La utilización de LoRa permite disponer de un dataframe que si bien es de tamaño reducido (hasta 255 bytes), permite la sencilla adición y modificación de los mensajes (en el **payload**), lo cual resulta de utilidad para la transmisión de la información sensada. Una ilustración de la estructura de los dataframe de LoRa se presenta en la Figura 3.8.

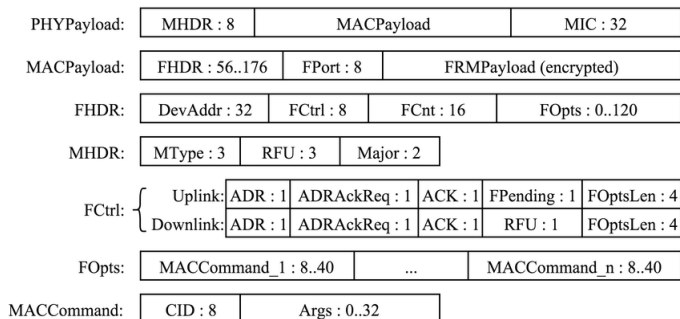


Figura 3.8: LoRa Dataframe [1].

Una vez elegida LoRa como la tecnología de comunicación en el diseño de la infraestructura TICAR habilitante, se procede a realizar la configuración de la red, junto al diseño de los parámetros de configuración, tal como se presenta en las Tablas 3.3, 3.4, 3.5 y 3.6. Estos parámetros de diseño serán utilizados para la posterior simulación de la red de sensores inteligentes (en luminarias inteligentes) en el framework FLoRa.

Optimización La tercera y última etapa en el diseño de la red de comunicaciones considera la verificación, monitoreo y ajuste de parámetros del diseño de la red propuesta.

Para validar el diseño de la infraestructura TICAR propuesta se hará uso de la plataforma de simulación FLoRa, framework enfocado en el diseño de topologías IoT para redes LoRa,

Parámetro	Valor
Frequency Carrier	915 [MHz]
Bandwith	0.25 [MHz]

Tabla 3.3: Parámetros de diseño en configuración de frecuencia de operación de módulos LoRa.

Parámetro	Valor
Radio Transmitter Bitrate	250 [Kbps]
Radio Transmitter Power	25.118 [mW]
Radio Energy Detection	-90 [dBm]
Radio Receiver Sensitivity	-137 [dBm]
Radio Receiver SNIR Threshold	-8 [dB]
Radio Antenna Type	Isotropic Antenna

Tabla 3.4: Parámetros de diseño en configuración de antenas para Transceiver LoRa.

Parámetro	Valor
MAC Bitrate	250 [bps]
MAC Header Length	8 [B]
MAC Sifs Time	10 [μ s]
MAC Slot Time	20 [μ s]

Tabla 3.5: Parámetros de diseño en configuración de capa MAC.

Parámetro	Valor
Gateway Network Layer Type	IPv4 Network Layer
Gateway Routing Table Type	IPv4 Routing Table
Gateway Ethernet MAC Type	Ethernet MAC Full Duplex
Gateway Queue Frame Capacity	100

Tabla 3.6: Parámetros de diseño en configuración de Gateways LoRa.

incluyendo nodos, sensores, gateways, Network Server, entre otros, el cual ha sido desarrollado sobre OMNeT++ 5.2. En este, se recrearán los escenarios propuestos bajo cada una de las disposiciones físicas previamente mencionadas, a fin de implementar la arquitectura de red propuesta, evaluar cada una de las topologías de red diseñadas, y determinar cual es la solución que presenta mejor desempeño de red.

Referido a las simulaciones, para recrear con mayor robustez las condiciones de la red se harán dos consideraciones. La primera consistirá en variar la desviación estándar asociada a la Ecuación 2.1 en $\sigma = \{0, 1.785, 3.57\}$ [dB], y la segunda en diseñar dos condiciones de red: condiciones de red normal, donde cada nodo envía un dataframe en una distribución exponencial con media de 60[s], y sobrecarga de tráfico en la red, donde cada nodo envía un dataframe en una distribución exponencial con media de 900[s]. El tiempo de simulación para cada escenario contemplará 7 días, a fin de otorgar mayor representatividad a los casos de estudio en el análisis de comportamiento de la red.

3.5.2.4. Verificación de disponibilidad de equipos

Al tratarse de tecnologías de vanguardia, tanto Raspberry Pi como LoRa se encuentran disponibles en el mercado, por lo cual existe disponibilidad de los equipos mencionados a lo largo del presente capítulo.

Capítulo 4

Resultados

El presente capítulo tiene por objetivo presentar cada uno de los resultados obtenidos tras la aplicación del marco metodológico, en cada uno de los escenarios propuestos para la validación del diseño de la infraestructura TICAR habilitante para la red de luminarias inteligentes, en el contexto de las Smart Cities. Los códigos utilizados e implementados en los frameworks Castalia y FLoRa se encuentran disponibles en <https://github.com/ibuguenoc/memoria-titulo>.

4.1. Comparativa de tecnologías IoT

En el presente apartado se muestran los resultados obtenidos para cada una de las métricas de interés en el contexto de tecnologías de comunicación, tras la simulación del escenario propuesto en la Figura 3.1, en conjunto con la utilización de los parámetros de diseño propuesto (ver Tabla 3.1 y 3.2), en el framework de simulación Castalia.

4.1.1. Métricas en recepción

Simulado el escenario propuesto en la Figura 3.1, se obtiene como métricas en recepción para cada uno de los protocolos MAC en estudio el número de paquetes recibidos, `packet reception rate` y `packet loss rate`, en función de la distancia de separación d , resultados tabulados en las Tablas 4.1, 4.2 y 4.3, respectivamente.

	$d = 10$ [m]	$d = 15$ [m]	$d = 20$ [m]	$d = 25$ [m]	$d = 30$ [m]
Slotted ALOHA	1495	1284	1219	998	800
T-MAC	1498	1283	1248	1031	819
Zigbee MAC	1248	1070	1048	874	693

Tabla 4.1: Número de paquetes recibidos, en función de la distancia de separación d y protocolo MAC.

	d = 10 [m]	d = 15 [m]	d = 20 [m]	d = 25 [m]	d = 30 [m]
Slotted ALOHA	0.99705	0.85657	0.81286	0.66562	0.53390
T-MAC	0.99895	0.85600	0.83248	0.68751	0.54646
Zigbee MAC	0.83281	0.71387	0.69911	0.58292	0.46225

Tabla 4.2: Packet reception rate, en función de la distancia de separación d y protocolo MAC.

	d = 10 [m]	d = 15 [m]	d = 20 [m]	d = 25 [m]	d = 30 [m]
Slotted ALOHA	0.00295	0.14343	0.18714	0.33438	0.46610
T-MAC	0.00105	0.14400	0.16752	0.31249	0.45354
Zigbee MAC	0.16719	0.28613	0.30089	0.41708	0.53775

Tabla 4.3: Packet loss rate, en función de la distancia de separación d y protocolo MAC.

4.1.2. Desglose de los paquetes recibidos

Asimismo, se obtiene un desglose en la recepción de los paquetes para cada uno de los protocolos MAC en estudio, utilizando como tasas de análisis `failed with no interference`, `failed, below sensitivity`, `failed, with non RX state` y `received with no interference`, en función de la distancia de separación d , resultados tabulados en las Tablas 4.4, 4.5, 4.6, 4.7 y 4.8.

	Failed with no interference	Failed, below sensitivity	Failed, non RX state	Received with no interference
Slotted ALOHA	0.00295	0.14343	0.18714	0.33438
T-MAC	0.00105	0.14400	0.16752	0.31249
Zigbee MAC	0.16719	0.28613	0.30089	0.41708

Tabla 4.4: Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 10$ [m].

	Failed with no interference	Failed, below sensitivity	Failed, non RX state	Received with no interference
Slotted ALOHA	0.01122	0.0	0.57640	0.41238
T-MAC	0.07874	0.0	0.02994	0.89132
Zigbee MAC	0.10418	0.0	0.0	0.89582

Tabla 4.5: Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 15$ [m].

	Failed with no interference	Failed, below sensitivity	Failed, non RX state	Received with no interference
Slotted ALOHA	0.00285	0.01396	0.58800	0.39519
T-MAC	0.02191	0.0	0.14857	0.82952
Zigbee MAC	0.04548	0.0	0.0	0.95452

Tabla 4.6: Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 20[m]$.

	Failed with no interference	Failed, below sensitivity	Failed, non RX state	Received with no interference
Slotted ALOHA	0.01591	0.01396	0.63723	0.33290
T-MAC	0.08889	0.0	0.16385	0.74727
Zigbee MAC	0.15656	0.0	0.0	0.84344

Tabla 4.7: Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 25[m]$.

	Failed with no interference	Failed, below sensitivity	Failed, non RX state	Received with no interference
Slotted ALOHA	0.02147	0.01396	0.69277	0.27179
T-MAC	0.10232	0.0	0.30441	0.59328
Zigbee MAC	0.24844	0.0	0.0	0.75156

Tabla 4.8: Desglose en la recepción de los paquetes, en tasas de recepción, para distancia $d = 30[m]$.

4.1.3. Latencia del nivel de aplicación

Finalmente, se obtiene un desglose en latencia de los paquetes recibidos para cada uno de los protocolos MAC en estudio, en función de la distancia de separación d , resultados tabulados en las Tablas 4.9, 4.10, 4.11, 4.12 y 4.13.

	Latencia [ms]					
	[100,120)	[120,140)	[140,160)	[160,180)	[180,200)	[200,inf)
Slotted ALOHA	0	0	1495	0	0	0
T-MAC	65	73	73	65	59	1072
Zigbee MAC	0	0	0	0	0	1246

Tabla 4.9: Desglose en latencia de número de paquetes recibidos, para distancia $d=10[m]$.

	Latencia [ms]					
	[100,120)	[120,140)	[140,160)	[160,180)	[180,200)	[200,inf)
Slotted ALOHA	0	0	1284	0	0	0
T-MAC	54	63	62	56	52	918
Zigbee MAC	0	0	0	0	0	1068

Tabla 4.10: Desglose en latencia de número de paquetes recibidos, para distancia $d=15[m]$.

	Latencia [ms]					
	[100,120)	[120,140)	[140,160)	[160,180)	[180,200)	[200,inf)
Slotted ALOHA	0	0	1219	0	0	0
T-MAC	53	57	57	53	51	895
Zigbee MAC	0	0	0	0	0	1046

Tabla 4.11: Desglose en latencia de número de paquetes recibidos, para distancia $d=20[m]$.

	Latencia [ms]					
	[100,120)	[120,140)	[140,160)	[160,180)	[180,200)	[200,inf)
Slotted ALOHA	0	0	998	0	0	0
T-MAC	40	42	48	46	44	743
Zigbee MAC	0	0	0	0	0	872

Tabla 4.12: Desglose en latencia de número de paquetes recibidos, para distancia $d=25[m]$.

	Latencia [ms]					
	[100,120)	[120,140)	[140,160)	[160,180)	[180,200)	[200,inf)
Slotted ALOHA	0	0	800	0	0	0
T-MAC	34	36	40	38	36	584
Zigbee MAC	0	0	0	0	0	691

Tabla 4.13: Desglose en latencia de número de paquetes recibidos, para distancia $d=30[m]$.

4.2. Diseño de infraestructura TICAR habilitante

En el presente apartado se muestran los resultados obtenidos para cada una de las métricas de interés en el contexto de tecnologías de comunicación, tras la simulación de la arquitectura WSN propuesta en la Figura 3.4, las topologías de red de las Figuras 3.5, 3.6 y 3.7, en conjunto con la utilización de los parámetros de diseño de red (ver Tabla 3.3, 3.4, 3.5 y 3.6), en el framework de simulación FLoRa, para la validación de la red de luminarias inteligentes propuesta, como infraestructura TICAR habilitante para Calles Inteligentes.

4.2.1. Frame Delivery Ratio

Simulado tanto la arquitectura WSN como las topologías de red propuestas, se obtiene el número total de frames enviados por todos los módulos LoRa de la red, número total de frames recibidos por todos los gateways de la red, y el frame delivery ratio respectivo, para cada topología y configuración de escenario, resultados tabulados en las Tablas 4.14 (condiciones normales de red) y 4.15 (sobrecarga de tráfico en la red).

		Frames enviados	Frames recibidos	Frame Delivery Ratio
Topología A	$\sigma = 0$ [dB]	31094	22504	0.7237
	$\sigma = 1.785$ [dB]	31524	24284	0.7703
	$\sigma = 3.57$ [dB]	31136	24283	0.7799
Topología B	$\sigma = 0$ [dB]	31148	23183	0.7443
	$\sigma = 1.785$ [dB]	31437	23733	0.7549
	$\sigma = 3.57$ [dB]	31223	23818	0.7628
Topología C	$\sigma = 0$ [dB]	31149	12954	0.4159
	$\sigma = 1.785$ [dB]	31219	15115	0.4842
	$\sigma = 3.57$ [dB]	31303	16590	0.53

Tabla 4.14: Frame Delivery Ratio - Condiciones normales de la red.

		Frames enviados	Frames recibidos	Frame Delivery Ratio
Topología A	$\sigma = 0$ [dB]	302329	203996	0.6747
	$\sigma = 1.785$ [dB]	302277	217700	0.7202
	$\sigma = 3.57$ [dB]	302648	222975	0.7367
Topología B	$\sigma = 0$ [dB]	302839	204168	0.6742
	$\sigma = 1.785$ [dB]	302288	209711	0.6937
	$\sigma = 3.57$ [dB]	302591	210759	0.6965
Topología C	$\sigma = 0$ [dB]	302627	103843	0.3431
	$\sigma = 1.785$ [dB]	302611	122872	0.406
	$\sigma = 3.57$ [dB]	302360	135485	0.4481

Tabla 4.15: Frame Delivery Ratio - Sobrecarga de tráfico en la red.

4.2.2. End to End Delay

Asimismo, se obtiene el end to end delay asociado a cada topología y configuración de escenario, en conjunto con los estadísticas correspondientes, resultados tabulados en las Tablas 4.16 (condiciones normales de red) y 4.17 (sobrecarga de tráfico en la red).

		End to End Delay [s]			
		Promedio	Desviación Estándar	Mínimo	Máximo
Topología A	$\sigma = 0$ [dB]	1.7736	0.4565	1.2881	2.9221
	$\sigma = 1.785$ [dB]	1.7583	0.455	1.2881	2.9221
	$\sigma = 3.57$ [dB]	1.7573	0.4501	1.2881	2.9221
Topología B	$\sigma = 0$ [dB]	1.808	0.4532	1.2881	2.9221
	$\sigma = 1.785$ [dB]	1.7906	0.4468	1.2881	2.9221
	$\sigma = 3.57$ [dB]	1.7914	0.4504	1.2881	2.9221
Topología C	$\sigma = 0$ [dB]	1.9155	0.4586	1.3498	2.9221
	$\sigma = 1.785$ [dB]	1.8819	0.4506	1.3498	2.9221
	$\sigma = 3.57$ [dB]	1.8798	0.4556	1.2881	2.9221

Tabla 4.16: End to End Delay - Condiciones normales de la red.

		End to End Delay [s]			
		Promedio	Desviación Estándar	Mínimo	Máximo
Topología A	$\sigma = 0$ [dB]	1.6594	0.3861	1.2881	2.9221
	$\sigma = 1.785$ [dB]	1.6441	0.3806	1.2881	2.9221
	$\sigma = 3.57$ [dB]	1.6413	0.3775	1.2881	2.9221
Topología B	$\sigma = 0$ [dB]	1.6891	0.3858	1.2881	2.9221
	$\sigma = 1.785$ [dB]	1.678	0.3827	1.2881	2.9221
	$\sigma = 3.57$ [dB]	1.6752	0.382	1.2881	2.9221
Topología C	$\sigma = 0$ [dB]	1.7985	0.4052	1.3498	2.9221
	$\sigma = 1.785$ [dB]	1.7699	0.3901	1.3498	2.9221
	$\sigma = 3.57$ [dB]	1.7645	0.3954	1.3498	2.9221

Tabla 4.17: End to End Delay - Sobrecarga de tráfico en la red.

4.2.3. RSSI Recibido

Además, se obtiene el RSSI recibido por cada gateway de la red, asociado a cada topología y configuración de escenario, en conjunto con los estadísticas correspondientes, resultados tabulados en las Tablas 4.18 (condiciones normales de red) y 4.19 (sobrecarga de tráfico en la red).

		RSSI Recibido [dBm]			
		Promedio	Desviación Estándar	Mínimo	Máximo
Topología A	$\sigma = 0$ [dB]	-106.8903	42.8985	-134.6113	11.0
	$\sigma = 1.785$ [dB]	-107.4249	42.2188	-136.9966	11.0
	$\sigma = 3.57$ [dB]	-107.0329	42.4815	-136.9888	11.0
Topología B	$\sigma = 0$ [dB]	-110.7235	38.6099	-135.7029	11.0
	$\sigma = 1.785$ [dB]	-111.5273	37.6802	-137.0000	11.0
	$\sigma = 3.57$ [dB]	-111.0508	37.9498	-136.9972	11.0
Topología C	$\sigma = 0$ [dB]	-129.8543	2.8008	-134.8043	-125.4736
	$\sigma = 1.785$ [dB]	-129.7534	3.0256	-136.9892	-119.8853
	$\sigma = 3.57$ [dB]	-128.9659	3.7309	-136.9929	-113.6202

Tabla 4.18: RSSI Recibido - Condiciones normales de la red.

		RSSI Recibido [dBm]			
		Promedio	Desviación Estándar	Mínimo	Máximo
Topología A	$\sigma = 0$ [dB]	-101.8403	45.7626	-134.6113	11.0
	$\sigma = 1.785$ [dB]	-103.2045	44.824	-136.9872	11.0
	$\sigma = 3.57$ [dB]	-103.4356	44.4036	-136.9968	11.0
Topología B	$\sigma = 0$ [dB]	-110.5423	36.9574	-135.7029	11.0
	$\sigma = 1.785$ [dB]	-110.8836	36.5784	-136.9989	11.0
	$\sigma = 3.57$ [dB]	-110.6631	36.5558	-136.9998	11.0
Topología C	$\sigma = 0$ [dB]	-129.2065	2.7284	-134.8043	-125.4736
	$\sigma = 1.785$ [dB]	-129.1086	2.9886	-136.9998	-117.8104
	$\sigma = 3.57$ [dB]	-128.2269	3.6992	-136.9999	-109.6952

Tabla 4.19: RSSI Recibido - Sobrecarga de tráfico en la red.

4.2.4. Internet Cloud Queue

Por su parte, se obtiene el **Internet Cloud Queue** (indicador del tiempo de encolamiento para el procesamiento de los datos en Internet Cloud) asociado a cada topología y configuración de escenario, en conjunto con los estadísticas correspondientes, resultados tabulados en las Tablas 4.20 (condiciones normales de red) y 4.21 (sobrecarga de tráfico en la red).

		Internet Cloud Queue [s]			
		Promedio	Desviación Estándar	Mínimo	Máximo
Topología A	$\sigma = 0$ [dB]	0.2251e-07	0.7791e-07	0.0	5.7183e-07
	$\sigma = 1.785$ [dB]	0.2124e-07	0.7427e-07	0.0	6.2435e-07
	$\sigma = 3.57$ [dB]	0.2302e-07	0.7719e-07	0.0	7.4246e-07
Topología B	$\sigma = 0$ [dB]	0.9861e-07	1.7735e-07	0.0	8.6522e-07
	$\sigma = 1.785$ [dB]	0.8344e-07	1.6741e-07	0.0	8.6522e-07
	$\sigma = 3.57$ [dB]	0.6703e-07	1.4875e-07	0.0	8.6522e-07
Topología C	$\sigma = 0$ [dB]	1.3665e-07	1.8412e-07	0.0	9.1447e-07
	$\sigma = 1.785$ [dB]	1.0972e-07	1.6942e-07	0.0	9.1447e-07
	$\sigma = 3.57$ [dB]	0.8955e-07	1.5641e-07	0.0	9.1447e-07

Tabla 4.20: Internet Cloud Queue - Condiciones normales de la red.

		Internet Cloud Queue [s]			
		Promedio	Desviación Estándar	Mínimo	Máximo
Topología A	$\sigma = 0$ [dB]	0.2625e-07	0.8131e-07	0.0	5.7183e-07
	$\sigma = 1.785$ [dB]	0.2247e-07	0.7504e-07	0.0	8.6522e-07
	$\sigma = 3.57$ [dB]	0.2460e-07	0.8009e-07	0.0	8.6522e-07
Topología B	$\sigma = 0$ [dB]	1.2770e-07	2.0319e-07	0.0	8.6522e-07
	$\sigma = 1.785$ [dB]	1.0811e-07	1.9243e-07	0.0	8.6522e-07
	$\sigma = 3.57$ [dB]	0.8865e-07	1.7404e-07	0.0	8.6522e-07
Topología C	$\sigma = 0$ [dB]	1.7028e-07	2.0126e-07	0.0	9.1447e-07
	$\sigma = 1.785$ [dB]	1.4480e-07	1.8624e-07	0.0	9.1447e-07
	$\sigma = 3.57$ [dB]	1.2248e-07	1.7633e-07	0.0	9.1447e-07

Tabla 4.21: Internet Cloud Queue - Sobrecarga de tráfico en la red.

4.2.5. Energía total consumida por nodo

Finalmente, se obtiene la Energía total consumida por nodo (durante los 7 días de simulación) asociado a cada topología y configuración de escenario, en conjunto con los estadísticas correspondientes, resultados tabulados en las Tablas 4.22 (condiciones normales de red) y 4.23 (sobrecarga de tráfico en la red).

		Energía total consumida por nodo [J]			
		Promedio	Desviación Estándar	Mínimo	Máximo
Topología A	$\sigma=0$ [dB]	69.9461	21.325	44.9263	131.8401
	$\sigma=1.785$ [dB]	71.0885	22.3443	44.4538	135.8971
	$\sigma=3.57$ [dB]	70.3534	22.3725	43.9497	133.4202
Topología B	$\sigma=0$ [dB]	70.4695	23.2142	44.4490	142.1822
	$\sigma=1.785$ [dB]	70.7071	21.7465	46.7671	134.3279
	$\sigma=3.57$ [dB]	70.5744	22.9164	44.7610	141.7428
Topología C	$\sigma=0$ [dB]	70.3592	22.4076	45.2052	137.6307
	$\sigma=1.785$ [dB]	70.6767	22.8745	44.1782	133.6919
	$\sigma=3.57$ [dB]	70.766	23.0157	45.9604	137.5795

Tabla 4.22: Energía total consumida por nodo - Condiciones normales de la red.

		Energía total consumida por nodo [J]			
		Promedio	Desviación Estándar	Mínimo	Máximo
Topología A	$\sigma=0$ [dB]	638.1675	58.6789	529.4222	794.8488
	$\sigma=1.785$ [dB]	637.7344	59.4263	538.5837	799.7195
	$\sigma=3.57$ [dB]	638.5461	59.1286	536.1966	794.008
Topología B	$\sigma=0$ [dB]	638.9769	59.8677	529.5503	802.5721
	$\sigma=1.785$ [dB]	637.5285	58.7354	533.3864	800.6523
	$\sigma=3.57$ [dB]	637.8912	59.2445	526.8849	792.4836
Topología C	$\sigma=0$ [dB]	638.622	60.0915	526.605	800.9321
	$\sigma=1.785$ [dB]	638.3976	59.1718	525.0526	798.9938
	$\sigma=3.57$ [dB]	637.9318	59.2984	531.4712	798.2419

Tabla 4.23: Energía total consumida por nodo - Sobrecarga de tráfico en la red.

Capítulo 5

Discusión

El presente capítulo tiene por objetivo realizar un análisis interpretativo de cada uno de los resultados obtenidos, tanto en la comparativa de tecnología IoT como en la validación del diseño de la infraestructura TICAR habilitante propuesta. Los códigos desarrollados y utilizados para la generación de los gráficos de los resultados obtenidos en los frameworks de simulación Castalia y FLoRa se encuentran disponibles en <https://github.com/ibuguenoc/memoria-titulo>.

5.1. Comparativa de tecnologías IoT

5.1.1. Métricas en recepción

De los resultados obtenidos en las Tablas 4.1, 4.2 y 4.3 se elaboran de manera respectiva lo gráficos de las Figuras 5.1, 5.2 y 5.3.

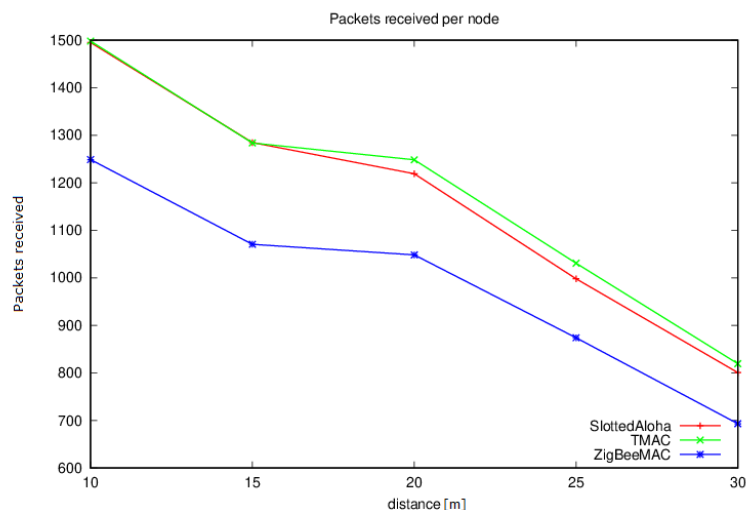


Figura 5.1: Número de paquetes recibidos, en función de la distancia.

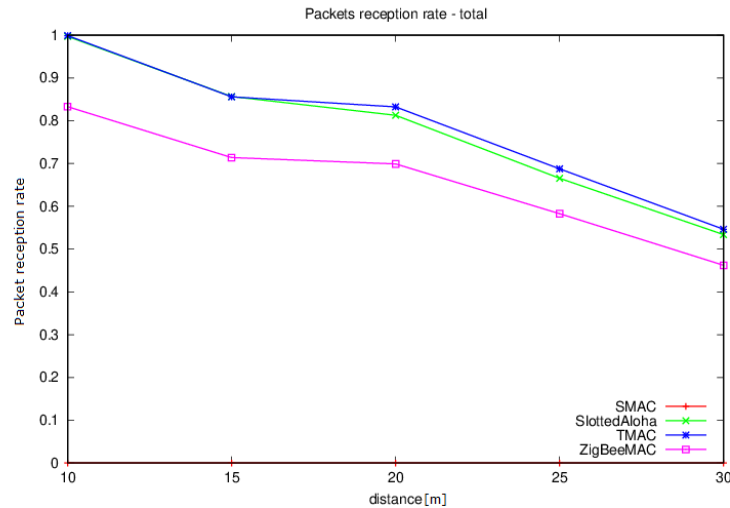


Figura 5.2: Tasa de paquetes recibidos, en función de la distancia.

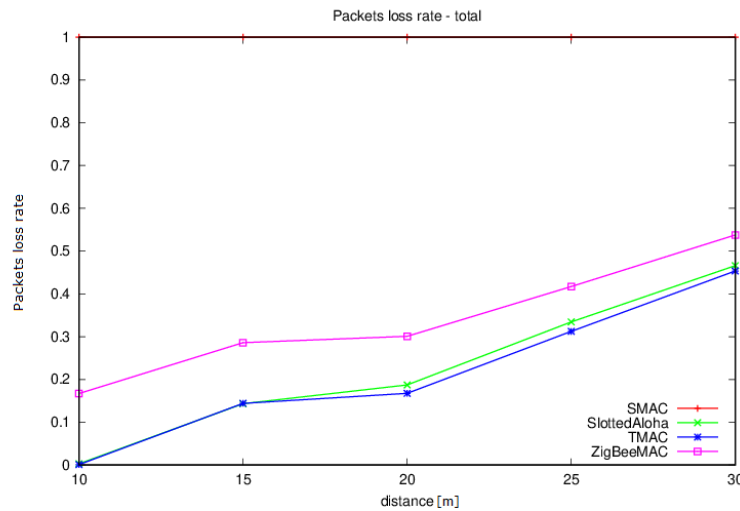


Figura 5.3: Tasa de paquetes perdidos, en función de la distancia.

A partir de la Figura 5.1, se observa que el protocolo de acceso al medio con mayor cantidad de paquetes correctamente recibidos corresponde al protocolo T-MAC, seguido por Slotted ALOHA y finalizando con Zigbee MAC (basado en el estándar IEEE 802.15.4, el cual hace uso de CSMA/CA), siendo consecuente con el **packet reception rate** y el **packet loss rate** de las Figuras 5.2 y 5.3. Ahora bien, basado en [7], se ha demostrado teóricamente que siempre CSMA/CA supera en desempeño al protocolo Slotted ALOHA, por lo cual los resultados obtenidos a nivel de simulación resultan ser contradictorios. Sin embargo, teniendo bajo consideración que el protocolo MAC asociado a la tecnología Zigbee es una variante del estándar IEEE 802.15.4, es probable este protocolo presente un desempeño inferior a Slotted ALOHA en condiciones de sobrecarga de tráfico de la red. Asimismo, la simplicidad del escenario de simulación puede ser una variable que conlleva a que el orden de desempeño de los protocolos sea T-MAC, Slotted ALOHA y Zigbee MAC.

5.1.2. Desglose en la recepción de los paquetes

De los resultados obtenidos en las Tablas 4.4, 4.5, 4.6, 4.7 y 4.8 se elaboran de manera respectiva lo gráficos de las Figuras 5.4, 5.5, 5.6 y 5.7.

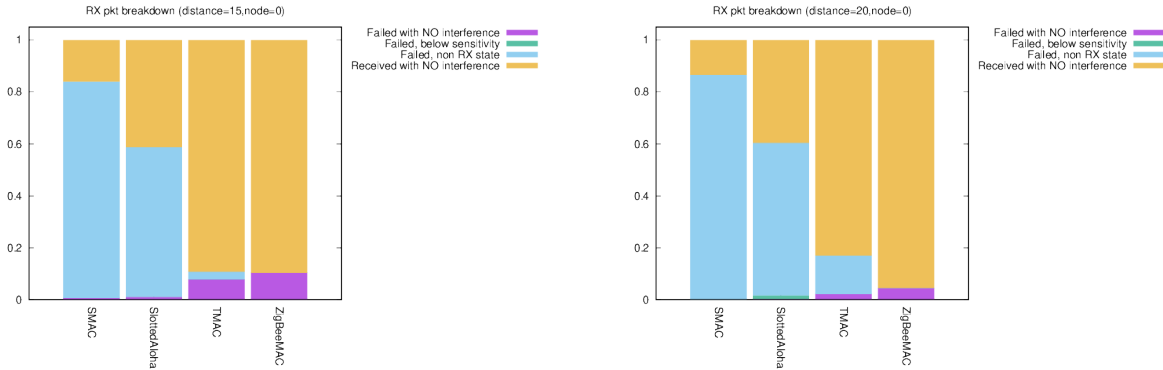


Figura 5.4: Desglose paquetes recibidos, con distancia de separación $d = 15[m]$.

Figura 5.5: Desglose paquetes recibidos, con distancia de separación $d = 20[m]$.

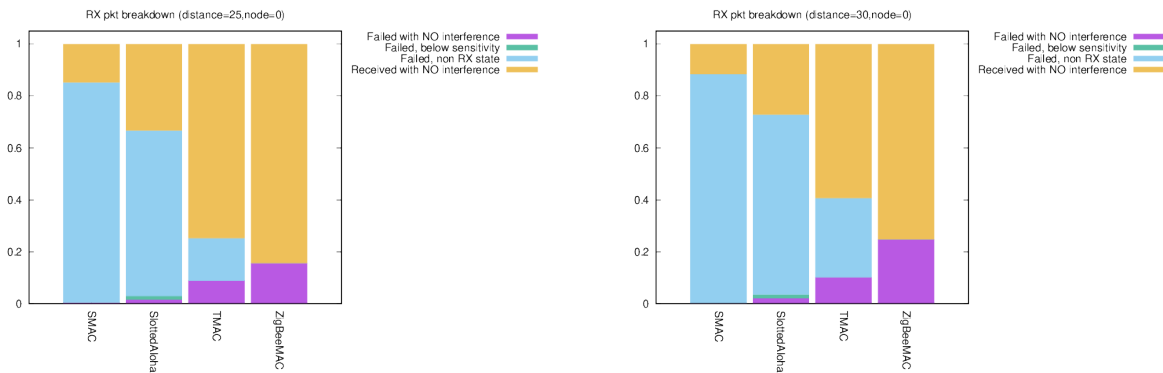


Figura 5.6: Desglose paquetes recibidos, con distancia de separación $d = 25[m]$.

Figura 5.7: Desglose paquetes recibidos, con distancia de separación $d = 30[m]$.

Referido al desglose en la recepción de los paquetes, los resultados obtenidos en las Figuras 5.4, 5.5, 5.6 y 5.7 muestran que a medida que aumenta la distancia de separación entre los nodos de la red, en todos los protocolos aumenta el número de paquetes cuya recepción fue fallida, no dependiendo de la interferencia (**failed with no interference**). Asimismo, en cuanto a la tasa de recepción correcta de los paquetes (**received with no interference**), los porcentajes más altos se presentan siempre en Zigbee MAC, aun cuando presente las mayores tasas **failed with no interference**. De esta manera, Zigbee MAC presenta mejor desempeño en el manejo de colisiones en comparación al protocolo T-MAC y Slotted ALOHA, siendo coherente con [7].

5.1.3. Latencia del nivel de aplicación

De los resultados obtenidos en las Tablas 4.9, 4.10, 4.11, 4.12 y 4.13 se elaboran de manera respectiva lo gráficos de las Figuras 5.8, 5.9, 5.10 y 5.11.

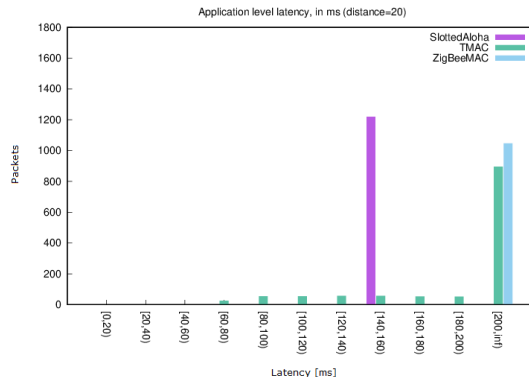
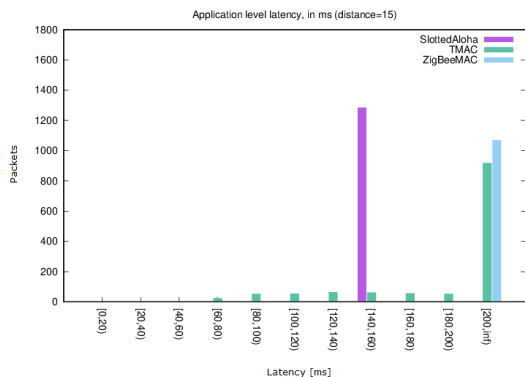


Figura 5.8: Latencia del nivel de aplicación, con distancia de separación $d = 15[m]$. Figura 5.9: Latencia del nivel de aplicación, con distancia de separación $d = 20[m]$.

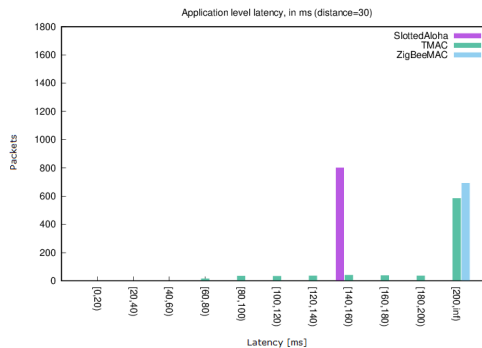
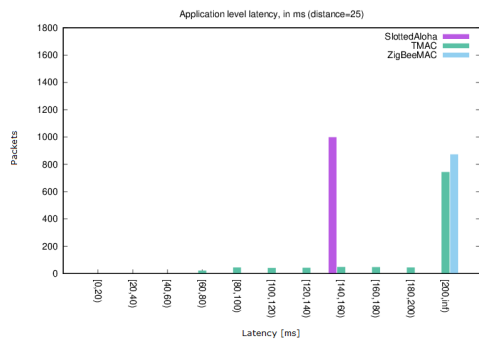


Figura 5.10: Latencia del nivel de aplicación, con distancia de separación $d = 25[m]$. Figura 5.11: Latencia del nivel de aplicación, con distancia de separación $d = 30[m]$.

Por último, en cuanto a la latencia de las tecnologías, a partir de las Figuras 5.8, 5.9, 5.10 y 5.11 se observa que el protocolo que presenta menor latencia asociada frente a la variación de la distancia de separación viene a ser Slotted ALOHA, lo cual se puede explicar en base a la simplicidad del protocolo (en comparación con IEEE 802.15.4). Si bien este aspecto no mejora el desempeño de red del protocolo, es una variable siempre a tener en consideración en sistemas de comunicación que requieren de bajas latencias en la transmisión de sus paquetes.

5.2. Diseño de infraestructura TICAR habilitante

5.2.1. Frame Delivery Ratio

De los resultados obtenidos en las Tablas 4.14 y 4.15 se elaboran de manera respectiva los gráficos de las Figuras 5.12 y 5.13.

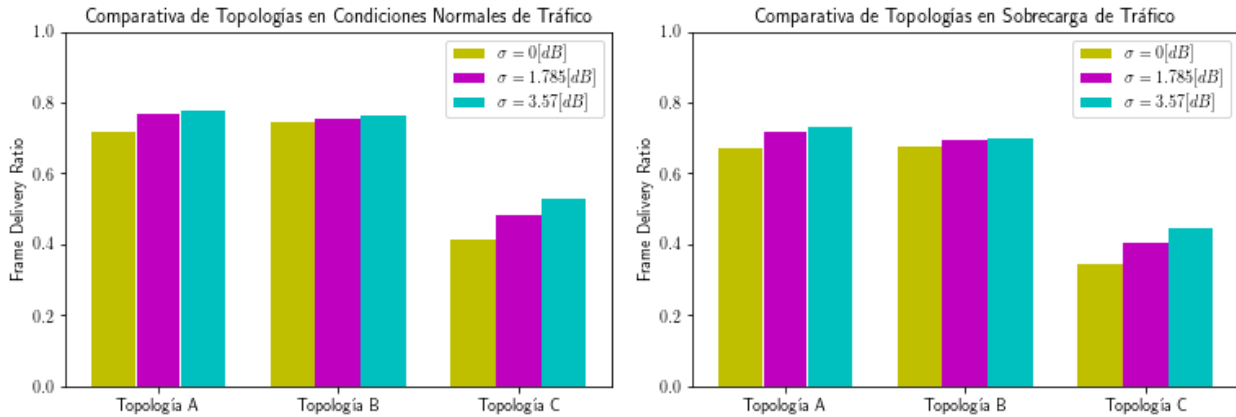


Figura 5.12: Frame Delivery Ratio, en red con condiciones normales de tráfico.

Figura 5.13: Frame Delivery Ratio, en red con sobrecarga de tráfico.

A partir de las Figuras 5.12 y 5.13, se observa que la topología de red que presenta en general el **frame delivery ratio** más alto (tanto en condiciones normales de la red como en sobrecarga de tráfico) corresponde a la Topología A, en la cual los gateways se ubican en las intersecciones de las calles (tal como se puede visualizar en la Figura 3.5). Ahora bien, la diferencia en comparación con la Topología B es mínima, esto principalmente debido a que los gateways de ambas topologías se disponen en la misma línea vista (la cual coincide con la dirección propia de las calles), ubicándose cercano a los módulos de transmisión LoRa. Donde no hay discrepancia es en establecer que la topología que presenta peor desempeño corresponde a la Topología C, en la cual los gateways se ubican en el punto central de las cuadras. Este bajo desempeño se explica por el hecho de que los gateways se encuentran a distancias técnicamente equidistante de los nodos transmisores, implicando que no existan nodos muy cercanos a estos dispositivos, como si sucede en las Topologías A y B, aumentando de esta manera la probabilidad de pérdida de frames en la propagación de las señales de transmisión al aumentar la distancia entre los nodos transmisores y los gateway respectivos.

5.2.2. End to End Delay

De los resultados obtenidos en las Tablas 4.16 y 4.17 se elaboran de manera respectiva los gráficos de las Figuras 5.14 y 5.15.

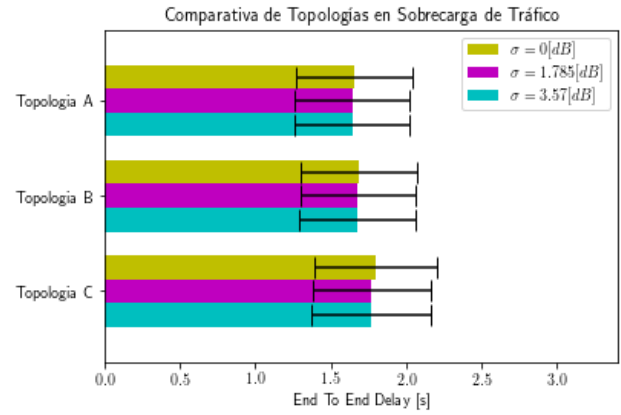
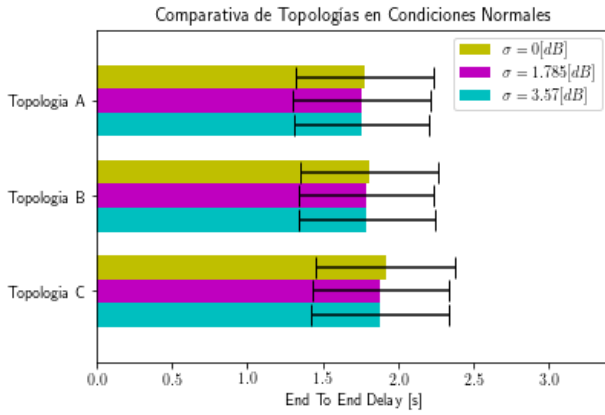


Figura 5.14: End-to-End Delay promedio, en red con condiciones normales de tráfico. Figura 5.15: End-to-End Delay promedio, en red con sobrecarga de tráfico.

Referido al end to end delay, de las Figuras 5.14 y 5.15 se observa que la topología de red que presenta menor tiempo de transmisión promedio desde el nodo transmisor hasta el Network Server corresponde a la Topología A. Al encontrarse ubicados los gateways de manera más cercana a los nodos de transmisión, el tiempo de que requiere un mensaje desde que es transmitido hasta que es recibido en el Network Server disminuye, y en caso contrario aumenta. Esto explica el aumento del end to end delay en la Topología C, en la cual la distancia mínima de separación de los nodos con el gateway aumenta al cambiar la disposición física de este elemento de red.

5.2.3. RSSI Recibido

De los resultados obtenidos en las Tablas 4.18 y 4.19 se elaboran de manera respectiva los gráficos de las Figuras 5.16 y 5.17.

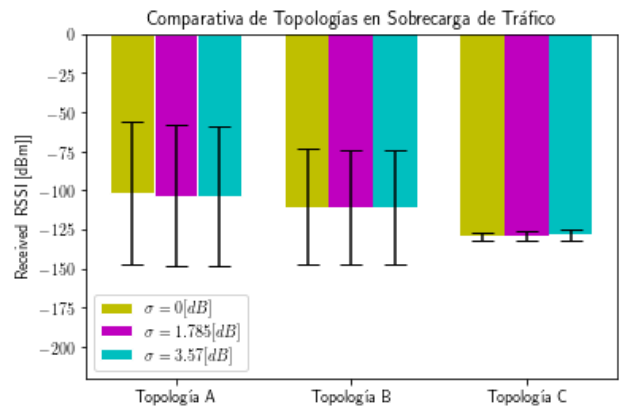
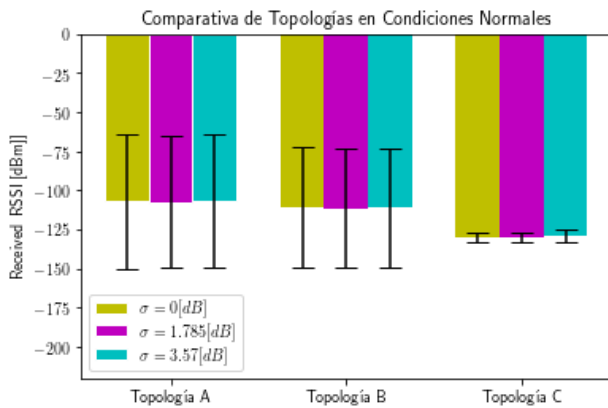


Figura 5.16: RSSI recibido promedio, en red con condiciones normales de tráfico. Figura 5.17: RSSI recibido promedio, en red con sobrecarga de tráfico.

En cuanto al **RSSI recibido**, a partir de las Figuras 5.16 y 5.17 se observa que la topología que presenta menor RSSI recibido promedio en los gateways de la red corresponde a la Topología A. Si bien el RSSI promedio se encuentra en un rango interpretado como de baja cobertura, resulta importante analizar estadísticas adicionales. Para esto, analizando las Tablas 4.18 y 4.19, se determina que existe una desviación estándar considerable en la Topología A y en la Topología B, lo cual conlleva a que el promedio no sea una métrica suficientemente representativa. Esto se puede comprobar al observar ambas topologías previamente mencionadas alcanzan un RSSI máximo de 11[dBm]. Por lo tanto, aún cuando el RSSI promedio asociado a la Topología A es bajo (con una desviación estándar que muestra que existe una distribución de las mediciones a tener bajo consideración), esta topología presenta el mejor indicador promedio de fuerza de las señales recibidas, en comparación con las Topologías B y C.

Manteniéndose en el **RSSI recibido**, resulta de interés visualizar que la todas las topologías presentan un mejor RSSI cuando la red presenta sobrecarga de tráfico, en comparación a condiciones normales de red. Este comportamiento se puede explicar al considerar que frente a una sobrecarga la red, mayor es el número de colisiones de los frames transmitidos, perdiendo fuerza de señal aquellos frames que han sido transmitidos desde los nodos más lejanos. De esta manera, los gateways recibirán aquellos frames que presentan una fuerza de señal mayor, y que corresponden a los que han sido transmitidos desde los nodos más cercanos. De esta manera, disminuirá el número de paquetes recibidos desde los nodos más lejanos, aumentando el RSSI promedio.

5.2.4. Internet Cloud Queue

De los resultados obtenidos en las Tablas 4.20 y 4.21 se elaboran de manera respectiva los gráficos de las Figuras 5.18 y 5.19.

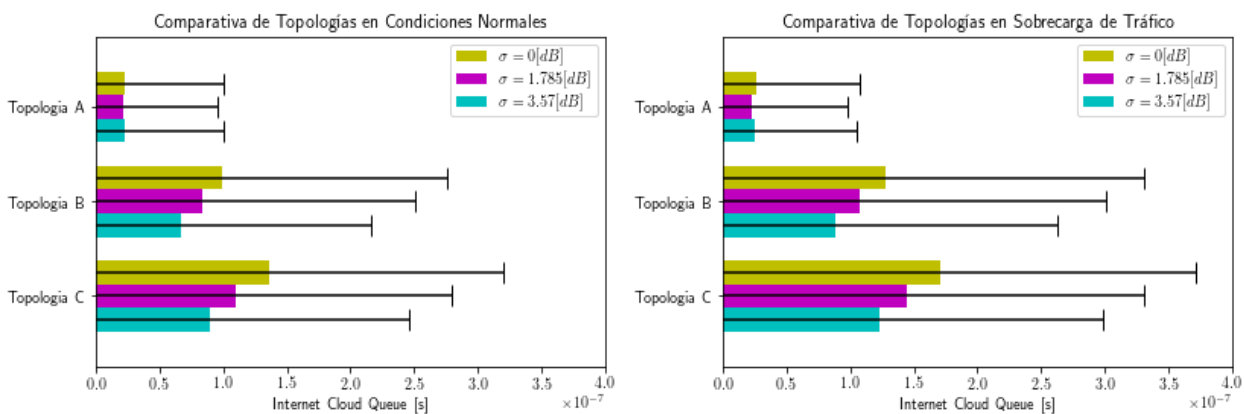


Figura 5.18: Internet Cloud Queue promedio, en red con condiciones normales de tráfico. Figura 5.19: Internet Cloud Queue promedio, en red con sobrecarga de tráfico.

Referido al **Internet Cloud Queue**, a partir de las Figuras 5.18 y 5.19 se visualiza que la Topología A presenta los menores tiempos de encolamiento asociados al Internet Cloud.

La explicación de esto se puede deber al hecho de que al presentar menor cantidad de colisiones, menos es el número de datos en el caché asociado, y por ende menor es el tiempo de encolamiento de los datos. Ahora bien, es importante destacar las unidades y magnitudes asociadas a estas mediciones, en las cuales técnicamente se está evaluando el desempeño en el orden de los $[\mu s]$

5.2.5. Energía total consumida por nodo

Finalmente, referido a la **Energía total consumida por nodo**, a partir de las Tablas 4.22 y 4.23 se observa que no existen discrepancias significativas entre topologías de red. Ahora bien, es sumamente destacable el bajo consumo energético promedio de cada uno de los nodos (incluyendo la desviación estándar asociada), considerando que estos indicadores corresponden a la energía total consumida en 7 días de simulación. De esta manera, bajo la perspectiva de IoT, la tecnología LoRa cumple con los criterios de bajo consumo energético requerido, siendo altamente probable una vida útil de 10 años para las baterías, como fuente de alimentación de los módulos.

Capítulo 6

Conclusiones

Las Smart Cities son un tipo de desarrollo urbano capaz de responder a las necesidades básicas de instituciones, empresas y de los propios habitantes de las ciudades. Las tecnologías de la información y comunicación adquieren un rol clave, puesto que facilitan la comunicación entre los diferentes servicios de la ciudad. Asimismo, la introducción de nuevas tecnologías permiten dotar de inteligencia a estos servicios, optimizando procesos asociados al funcionamiento propio de las urbes. De esta manera, un pilar fundamental de estas ciudades viene a ser la utilización de tecnologías TICAR.

Para el diseño de una infraestructura TICAR habilitante para Calles Inteligentes, en el contexto de Smart Cities, resulta de gran relevancia tener una panorámica general de los elementos conformantes de este paradigma. La definición misma de una Ciudad Inteligente integra a las tecnologías de comunicación como un ente de gran importancia, motivo por el cual es necesario comprender el background teórico de las mismas en cada uno de sus enfoques. Asimismo, Internet of Things adquiere un rol protagónico en las Smart Cities, al ofrecer una interconexión digital de dispositivos inteligentes, siendo estos capaces de analizar el entorno y tomar decisiones inteligentes. Para sostener esta interconexión, las telecomunicaciones presentan a las redes de sensores inalámbricos como la arquitectura óptima sobre la cual desarrollar, implementar y operar esta interconexión digital. Cada desarrollo tecnológico requiere de una regulación, motivo por el cual en Chile existe un marco y entes regulatorios que procuran la normalización en la integración de estas nuevas tecnologías. Finalmente, la revisión de infraestructuras públicas actuales permite analizar la factibilidad en la integración de una red de sensores inteligentes.

En la misma línea, previo a la presentación de la infraestructura propuesta, es necesario analizar cuáles son las tecnologías de comunicación actuales que permite el desarrollo de sistemas inteligentes. En IoT, dos tecnologías emergentes están adquiriendo un rol protagónico: LoRa y Zigbee. Ambas tecnologías se encuentran patentadas, por lo cual múltiples especificaciones a nivel de comunicaciones se encuentran con acceso restringido, sin embargo, esto no ha sido contraproducente. Gracias a las diversas características que estas tecnologías ofrecen, es que se han convertido en las tecnologías de mayor uso a nivel de IoT. Ahora bien, ¿qué hace que una tecnología sea preferida por sobre otra? Los protocolos asociados a la capa de acceso al medio.

La tecnología LoRa basa su protocolo de la capa MAC en el protocolo Slotted ALOHA, mientras que Zigbee basa su protocolo en el estándar IEEE 802.15.4 (principalmente en CSMA/CA). A partir del background teórico, es de conocimiento que CSMA/CA supera teóricamente en desempeño de red a Slotted ALOHA, por lo cual Zigbee debiese ser la tecnología de preferencia en el diseño de redes WSN. Sin embargo, al simular escenarios que recrean ambientes urbanos e integran estos protocolos, se visualiza que Slotted ALOHA presenta mejores métricas a nivel de **packet reception rate**, **packet loss rate** y **latencia**. Aún cuando esto puede resultar contradictorio, técnicamente no lo es, considerando que Zigbee es una tecnología enfocada para soluciones IoT de corto alcance, mientras que LoRa está enfocada para otorgar amplios rangos de cobertura. De esta manera, la utilización de una tecnología sobre la otra dependerá exclusivamente de las condiciones propias del entorno, y la respuesta frente a estos requerimientos de cada tecnología.

Referido al diseño de la infraestructura TICAR habilitante para Calles Inteligentes, se propone en base a la revisión de antecedentes como solución montar una red de sensores inteligentes sobre el alumbrado público, dotando de inteligencia a una infraestructura física ya existente. Para esto, bajo el paradigma IoT, hay dos aristas a tener en consideración para el diseño de una solución tecnológica. La primera arista hace referencia a la inteligencia de los sensores, y para esto se propone como módulo de integración de los sensores y actuadores plataformas de desarrollo tales como Raspberry Pi y Arduino, ordenadores de placa reducida que ofrecen una versátil y sencilla integración con tecnologías IoT, junto a capacidades locales de procesamiento. La segunda arista hace referencia al modelamiento de la infraestructura desde el punto de vista de las telecomunicaciones.

Para el modelamiento de la infraestructura propuesta desde el punto de vista de las telecomunicaciones, inicialmente se realiza una definición de los elementos propios de la red, arquitectura y topología de la misma, en conjunto a su disposición física. Considerando que la red de luminarias inteligentes se basa en la implementación de una red de sensores, la arquitectura principal de la infraestructura propuesta se basa en las arquitecturas WSN, las cuales contemplan una capa de objetos, una capa de red de comunicaciones, y una capa de aplicación y análisis. La definición de esta arquitectura conlleva a la definición de los elementos propios de la red, determinando la necesidad de disponer de módulos de tecnologías de comunicación inalámbrica en los postes del alumbrado público, gateways encargado de recibir la información transmitida y enviarla a los sistemas informáticos respectivos, enrutadores para proporcionar conectividad a nivel de red, Firewall como elemento de seguridad, Internet Cloud como servicio en red para el procesamiento y análisis inteligente de datos, y servidores de red externos.

Una vez definida la arquitectura de la red, se presentan tres topologías de red que consideran cada una de las condiciones existentes en canales de comunicación inalámbricos. La diferencia principal entre estas topologías radica principalmente en la disposición física de los gateways asociados, variable que afectará el comportamiento y desempeño de la red.

Considerando que las características propias del entorno del alumbrado público exigen la necesidad de utilizar una tecnología de ofrezca simplicidad en el diseño de la arquitectura y topología de la red, un correcto manejo de colisiones en la capa MAC, bajo consumo energético, baja transferencia de datos, fácil integración a plataformas de desarrollo, y soporte al despliegue masivo de red de sensores en un amplio rango de cobertura, es que se elige a LoRa como la tecnología de comunicación para la presente infraestructura habilitante.

Para la validación de la infraestructura TICAR diseñada bajo el enfoque de las telecomunicaciones, se hace uso del framework de simulación LoRa, enfocado en entornos IoT que integran tecnología LoRa. A partir de las simulaciones de los diversos escenarios que caracterizan la variabilidad en el modelo del medio de propagación, junto a la evaluación de las topologías propuestas, se determina tras la obtención de cinco métricas de interés tanto en el aspecto de telecomunicaciones como de sistemas IoT (`frame delivery ratio`, `end to end delay`, `RSSI recibido`, `Internet Cloud Queue` y `Energía total consumida por nodo`) que la Topología A presenta el mejor desempeño de red, en condiciones normales de la red y en sobrecarga de tráfico. Esta topología dispone a los gateways en las intersecciones de las calles, presentando un alto desempeño como red de sensores inteligentes, haciendo uso de la tecnología LoRa.

De esta manera, se logra presentar una propuesta de infraestructura TICAR habilitante para Calles Inteligentes, con enfoque en telecomunicaciones: una red de luminarias inteligentes, cumpliendo con el objetivo general del presente trabajo de título. El uso de tecnologías de comunicación como LoRa permite otorgar robustez y versatilidad al sistema propuesto, siendo bajo todos los efectos necesario realizar una comparación futura con otras tecnologías IoT emergentes de largo alcance.

En cuanto al cumplimiento de los objetivos específicos, se logra modelar una infraestructura TUCAR habilitante, en redes de luminarias inteligentes, por medio de la revisión del estado del arte, y un modelamiento de la red de sensores bajo el enfoque de las telecomunicaciones e IoT. Este modelamiento logra ser posteriormente simulado, logrando realizar una validación del mismo por medio de la recreación de ambientes urbanos. Como punto pendiente queda el modelamiento bajo una perspectiva económica, lo cual vendría a ser como trabajo a futuro realizar un estudio de caso de negocio de la infraestructura propuesta, con proyecciones a nivel de mercado de la misma.

Por último, la utilización de los frameworks de simulación de entornos IoT Castalia y LoRa, permitieron realizar de manera respectiva una comparativa de las tecnologías IoT de interés, bajo la mirada de los protocolos en la capa MAC, y una validación de la infraestructura TICAR propuesta. Ambos entornos de simulación se encuentran desarrollados sobre OMNeT++, plataforma de simulación de eventos discretos por excelencia en telecomunicaciones, lo cual otorga una mayor validez a las simulaciones realizadas, junto a los resultados obtenidos.

Bibliografía

- [1] T. Clausen A. Augustin, J. Yi and W. M. Townsley. *A Study of LoRa: Long Range Low Power Networks for the Internet of Things*. Sensors 16, 2016. Disponible en <https://www.mdpi.com/1424-8220/16/9/1466>.
- [2] P. Moreno A. Ludovici and A. Calveras. *TinyCoAP: A novel Constrained Application Protocol (CoAP) implementation for embedding RESTful Web services in wireless sensor networks based on TinyOS*. Journal of Sensor and Actuator Networks, 2013.
- [3] A.A.A. Alkhatib and G.S. Baicher. *Wireless Sensor Network Architecture*. 2012 International Conference on Computer Networks and Communication Systems (CNCS 2012), IACSIT Press, Singapore, 2012.
- [4] LoRa Alliance. *LoRaWAN Specification*. 2015. Disponible en <https://lora-alliance.org/resource-hub/lorawantm-specification-v11>.
- [5] ZigBee Alliance. *ZigBee Specification Document*. 2008.
- [6] A.P. Athreya and P. Tague. *Network Self-Organization in the Internet of Things*. IEEE International Workshop on Internet-of-Things Networking and Control (IoT-NC 2013), New Orleans, LA, 2013. Disponible en <https://ieeexplore.ieee.org/document/6644956>.
- [7] S. Banaouas B. Blaszczyszyn, P. Muhlethaler. *A comparison of Aloha and CSMA in Wireless Ad-Hoc Networks under Different Channel Conditions*. INRIA, 2010. Disponible en <https://hal.inria.fr/inria-00530093v1>.
- [8] Powercast Corporation. *Lifetime Power Wireless Sensor System*. 2012. Disponible en https://www.kele.com/Catalog/14%20Network_and_Wireless/PDFs/Lifetime%20Power%20Installation.pdf.
- [9] P. Grossetete R. Barton J. Henry D. Hanes, G. Salgueiro. *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Cisco Press, Indianapolis, Indiana 46240 USA, 2017.
- [10] Congreso Nacional de Chile. *Pesos i medidas.-Lei sobre la materia*. Biblioteca del Congreso Nacional, 1848. Disponible en <https://www.leychile.cl/Navegar?idNorma=1024220>.

- [11] Consejo Municipal de la Comuna de Quilpué. *Ordenanza N 4: Instalación de Alumbrado Público en calles*. 2013. Disponible en https://www.quilpue.cl/uploads/biblioteca/biblioteca_4_ORDENANZA_ALUMBRADO_PUBLICO_.pdf.
- [12] M.O. Farooq and T. Kunz. *Operating Systems for Wireless Sensor Networks: A Survey*. Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, 2011.
- [13] G. Ferré. *Collision and Packet Loss Analysis in a LoRaWAN Network*. 2017 25th European Signal Processing Conference (EUSIPCO), Bordeaux, France, 2017.
- [14] W. Li G. Zhou, L. Huang and Z. Zhu. *Harvesting ambient environmental energy for wireless sensor networks: A survey*. Journal of Sensors 2014, 2014.
- [15] M. Raynor J. Holdowsky, M. Mahto and M. Cotteleer. *Inside the Internet of Things*. Deloitte University Press, Westlake, 2015.
- [16] X. Liang. *Lecture Note 5: Local Area Networks and Medium Access Control Protocols*. University of Waterloo, 2013.
- [17] A. Zanella M. Centenaro, L. Vangelista and M. Zorzi. *Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios*. IEEE Wireless Communications, 2016. Disponible en <https://ieeexplore.ieee.org/document/7721743>.
- [18] T. Leva M. Ilaghi and M. Komu. *Techno-Economic Feasibility: Analysis of Constrained Application Protocol*. IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, 2014. Disponible en <https://ieeexplore.ieee.org/document/6803138>.
- [19] Networks and Pervasive Computing. *Castalia: An OMNeT-based simulator for low-power wireless networks such as Wireless Sensor Networks and Body Area Networks*. National ICT Australia, 2016. Disponible en <https://github.com/boulis/Castalia>.
- [20] London: Stationery Office. *Climate Change Act 2008*. 2008.
- [21] M. Park. *IEEE 802.11ah: Sub-1-GHz license-exempt operation for the internet of things*. IEEE Communications Magazine, 2015. Disponible en <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7263359>.
- [22] Gopika Premsankar and Mariusz Slabicki. *FLoRa*. 2017. Disponible en <https://flora.aalto.fi/>.
- [23] C. Simpson. *Health impact assessment of introducing LED street lighting in Stockport*. Lighting Project Team, Stockport, England, 2015.
- [24] C. Simpson. *Street Lighting Project – Full Business Case*. Lighting Project Team, Stockport, England, 2015.
- [25] M.P. Singh S.K. Singh and D.K. Singh. *Routing protocols in wireless sensor networks: A survey*. International Journal of Computer Science and Engineering Survey

- (IJCSSES), 2010. Disponible en <http://www.airccse.org/journal/ijcses/papers/1110ijcses06.pdf>.
- [26] H. Suo and C. Zou. *Security in the Internet of Things: A Review*. 2012 International Conference on Computer Science and Electronic Engineering (ICCSEE 2012), Hangzhou, Zhejiang China, 2012. Disponible en <https://ieeexplore.ieee.org/document/6188257>.
- [27] Y.-H. Tseng C.-H. Lee V.S. Somayazulu S.-Y. Chien, W.-K. Chan and Y.-K. Chen. *Distributed Computing in IoT: System-on-a-Chip for Smart Cameras as an Example*. 20th Asia and South Pacific Design Automation Conference (ASP-DAC), Chiba, Japan, 2015.
- [28] K. Hartke Z. Shelby and C. Bormann. *The Constrained Application Protocol (CoAP)*. Internet Engineering Task Force (IETF), 2014.
- [29] Marco Zuniga and Bhaskar Krishnamachari. *Analyzing the Transitional Region in Low Power Wireless Links*. First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON), Santa Clara, CA, 2004.

Apéndice A

Antecedentes

A.1. Modelo TCP/IP

- **Capa de acceso a la red:** esta capa inferior del modelo TCP/IP permite acceder al medio físico de una red de telecomunicaciones, siendo comparable a la capa de enlace de datos y a la capa física del modelo OSI. Así, esta capa proporciona la codificación de datos, modificando el modelo de señal digital, interpreta el estado de la señal frente a secuencias de bits, y permite transmitir esta información como señales eléctricas u ópticas. Además, permite establecer el vínculo lógico entre dos nodos de una red, tener un control del tráfico de las tramas, detectar los errores presentes ocurridos la capa física, y administrar el acceso al medio por parte de los datos
- **Capa de Internet:** esta capa intermedia del modelo TCP/IP es la responsable de proporcionar el paquete de datos para la transmisión de la información en la red de telecomunicaciones. Es asimilable a la capa de red del modelo OSI.
- **Capa de transporte:** esta capa intermedia del modelo TCP/IP brinda los datos de enrutamiento, junto con los mecanismos que permiten conocer el estado de la transmisión. Es asimilable a la capa de transporte del modelo OSI.
- **Capa de aplicación:** esta capa superior del modelo TCP/IP incorpora aplicaciones de red estándar, manejando aspectos de representación, codificación y control de diálogo. Es asimilable a la capa de sesión, presentación y aplicación del modelo OSI.

El protocolo IP es el más importante en estos modelos puesto que al corresponder al número que identifica la interfaz de red de un dispositivo electrónico, permite establecer el origen o destino de la información para el proceso de comunicación entre nodos una red de telecomunicaciones.

A.2. Protocolo MAC: CSMA/CA

A.2.1. Distributed Coordination Function

En CSMA/CA, la función de coordinación distribuida (Distributed Coordination Function - DCF) controla el tiempo que una estación espera antes de iniciar la transmisión en un medio libre. Esta función también asigna ciertos intervalos de tiempo a los elementos de la red para otras acciones, creando una estructura de tiempo vinculante. Este procedimiento es el foco de la prevención de colisiones: una estructura de tiempo compleja que hace posible evitar colisiones. DCF tiene en cuenta varios intervalos al crear la estructura de tiempo.

- **DCF Interframe Space (DIFS):** en primer lugar, los nodos deben monitorear la red en base a la duración del DIFS para determinar si el medio actualmente está libre. En CSMA/CA, esto significa que ninguna otra estación dentro del rango está transmitiendo al mismo tiempo. Los resultados DIFS del SIFS casi duplican el tiempo de ranura, que es entre 28 $[\mu s]$ y 50 $[\mu s]$ de largo.
- **Contention window:** si los nodos determinan que el canal se encuentra libre, esperan una cantidad de tiempo aleatoria antes de comenzar a enviar. Esta duración corresponde a la ventana de contención, la cual se duplica con cada colisión.
- **Short Interframe Space (SIFS):** después de enviar el paquete de datos, el nodo receptor envía una notificación (siempre y cuando este nodo también utilice el procedimiento RTS/CTS), sin embargo, esta estación también espera un tiempo fijo antes de transmitir. SIFS es el tiempo que toma procesar un paquete de datos. La duración depende del estándar IEEE-802.11 y se encuentra entre los 10 $[\mu s]$ y 16 $[\mu s]$.

A.2.2. RTS/CTS

Los frames "Request to Send" (RTS - "Solicitud de envío") y "Clear to Send" (CTS - "Borrar para enviar") forman parte de la extensión opcional de CSMA/CA RTS/CTS. Este procedimiento es anterior a la transmisión de datos real. Si un nodo determina que el medio de transmisión está libre, el dispositivo primero envía un RTS al nodo receptor. Con esto, se manifiesta que el nodo quiere iniciar una transmisión, y ocupará el medio de transmisión durante un cierto tiempo.

El receptor, a su vez, envía un CTS al remitente original, informando a todos los nodos restantes que el medio se encuentra ocupado, y que el nodo transmisor está habilitado para la transmisión. Informados, el nodo transmisor comienza a transmitir los datos. Ahora, no es posible que los nodos detecten colisiones u otras interferencias durante la transmisión. Por este motivo, la estación receptora debe enviar un ACK cuando el paquete de datos ha sido recepcionado de manera correcta.

Si el ACK no aparece, el nodo transmisor supone que se ha producido una complicación y vuelve a enviar el paquete de datos. La estación tiene un derecho preferencial para usar el medio y no tiene que esperar de nuevo para que el canal sea libre.

Los frames RTS, CTS y ACK se encuentran conformados por diferentes campos en sus estructuras, los cuales se describen a continuación.

- **Frame control:** el campo Frame Control contenido en cada frame es de 2 bytes (16 bits), dividiéndose en varios elementos:
 - **Protocol version:** especifica la versión del protocolo utilizado.
 - **Type:** especifica si se trata de un frame de control (como con RTS/CTS y ACK), frame de datos o frame de administración.
 - **Sub-type:** especifica el tipo de frame definiendo una de las 25 subcategorías existentes.
 - **To distribution system:** se activa cuando el frame va a un sistema de distribución.
 - **From distribution system:** se activa si el frame proviene de un sistema de distribución.
 - **More fragments:** solo tiene contenido si siguen más frames (solo relevante para frames de datos y frames de administración).
 - **Retry:** especifica si se ha enviado el frame, y con qué frecuencia.
 - **Power management:** muestra el modo de ahorro de energía.
 - **More data:** especifica que si más datos deben ser enviados.
 - **WEP:** indica si los datos están encriptados con WEP.
 - **Order:** informa al destinatario si los datos se envían en el orden correcto.
- **Duration:** especifica el tiempo que necesita el transmisor para la transmisión de datos (esta información es crucial para el vector de asignación de red y tiene un tamaño de 2 Bytes).
- **Receiver address:** contiene la dirección MAC del receptor (6 Bytes).
- **Transmitter address:** contiene la dirección MAC del transmisor (6 Bytes); sólo se requiere para RTS, no para CTS y ACK.
- **Frame check sequence:** la secuencia de comprobación de bloque de 4 Bytes es una suma de comprobación que permite al nodo receptor determinar si el frame de datos ha llegado según lo planeado. El transmisor calcula la suma de comprobación a partir del dataframe. El mismo proceso también tiene lugar en el lado del receptor cuando el frame ha sido recibido. Si el resultado del receptor coincide con lo que el transmisor adjuntó al frame como FCS, entonces la transmisión fue exitosa.

Excepto por el campo que especifica la dirección del transmisor, todos los campos están contenidos en RTS, CTS y ACK (tal como se puede visualizar en Tabla A.1 y Tabla A.2. Sin embargo, **Transmitter address** solo es de interés cuando se contacta por primera vez con el receptor, de modo que este sea capaz de saber con qué nodo se está comunicando.

Frame Control	Duration	Receiver Address	Transmitter Address	Frame Check
2 Bytes	2 Bytes	6 Bytes	6 Bytes	4 Bytes

Tabla A.1: Estructura RTS Frame.

Al extender el protocolo CSMA/CA con RTS/CTS, es posible reducir las colisiones en la trama RTS inicial. Ahora bien, aún es probable que dos nodos envíen una solicitud para

Frame Control	Duration	Receiver Address	Frame Check
2 Bytes	2 Bytes	6 Bytes	4 Bytes

Tabla A.2: Estructura CTS & ACK Frame.

transmitir a la misma estación de manera simultánea, sin embargo, en casos como este, el receptor no envía una trama CTS porque las tramas RTS no han llegado correctamente. De esta manera, RTS/CTS puede resolver el problema del nodo oculto: incluso si los dos transmisores no se reconocen entre sí debido al rango limitado, solo las tramas RTS están en riesgo y no los datos reales. Por lo tanto, CSMA/CA entra en vigor y la transmisión puede realizarse de manera ordenada.

A.3. Protocolo TCP

TCP posibilita administrar en la capa de aplicaciones los datos que provienen desde la capa física, desatendiéndose de la operación de la capa de transporte, pues este protocolo es el encargado de monitorear la información.

Durante la comunicación la conexión entre las máquinas se llama conexión en línea, en la cual tanto el emisor (conocido también como cliente) como el receptor (conocido también como servidor) se comunican en ambas direcciones. Para lograr este proceso, los datos a transmitir se agrupan de tal manera de agregar encabezados al envío de cada uno, formando los denominados paquetes de datos, sincronizando así las transmisiones de información y garantizando la recepción de la señal emitida.

La gran característica del protocolo TCP es que permite la realización del proceso de multiplexación/demultiplexación, transmitiendo de esta manera datos desde diversas aplicaciones por el mismo canal comunicativo, gracias a la utilización de puertos (dirección IP de una máquina determinada a la cual será enviada la información).

Referido a ejemplos en donde se aplica el protocolo TCP, se encuentran:

- **TELNET:** aplicación que permite realizar una conexión remota a través de la red con una máquina multiusuario.
- **FTP:** protocolo implementado para estandarizar el proceso de copia de ficheros de una máquina a otra.
- **TFTP:** protocolo utilizado para transferencia de ficheros, donde sólo puede leer y escribir un fichero de un servidor.
- **SMTP:** protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otro tipo de dispositivos.
- **IP Móvil:** protocolo de red que permite a los usuarios de dispositivos móviles moverse de una red a otra, manteniendo la misma dirección IP.

A.4. Protocolo UDP

El contenido de los datagramas en el protocolo UDP se encuentra caracterizado por tener una cabecera para el puerto de origen (número del remitente del segmento UDP transmitido), otro para el puerto de destino (número del puerto al cual se envía el paquete), otro para la longitud del mensaje, otro para la suma de comprobación (permite controlar la integridad del mensaje transmitido) y por último por contener a los datos a transmitir. El uso principal de este protocolo se da mayormente en aquellos protocolos en donde los intercambios de paquetes de la conexión no son rentables con respecto a la información transmitida, y en transmisión de audio y video en tiempo real.

Referido a ejemplos más específicos en donde se aplica el protocolo UDP, se encuentran:

- **DHCP:** protocolo cliente/servidor que permite a los clientes de una red TCP/IP obtener información para su configuración.
- **DNS:** sistema de nomenclatura para dispositivos conectados a redes IP, el cual asocia información variada con nombres de dominios, permitiendo establecer una correlación entre las direcciones IP y el nombre de los dominios.
- **RIP:** protocolo de puerta de enlace interna utilizado por los routers para intercambiar información de las redes IP a las que se encuentran conectados.
- **SNMP:** protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red.

A.5. Diseño de redes de ordenadores

A.5.1. Paradigma Cliente-Servidor

Este tipo de diseño corresponde a un modelo de aplicación en la red de comunicaciones en el que las tareas se reparten entre los proveedores de servicios y los clientes, centralizando la gestión de la información. En esta arquitectura los clientes se conectan al servidor, en el cual se centralizan y gestionan los diversos recursos disponibles, dejándolos a servicio de los clientes cada vez que son solicitados, disponiendo de esta manera los requerimientos provenientes de los clientes que tienen mayor prioridad.

Las principales características de este modelo son que cliente y el servidor pueden actuar en conjunto o de manera independiente en la realización de tareas, las funciones de ambos pueden estar en la misma o distintas plataformas, existe una interrelación entre el hardware y el software de la arquitectura, permitiendo un acceso simple a los recursos de la red, y el usuario dispone de sus bases de datos, sin requerir del sistema central de información.

Una aplicación general de este modelo es la interacción entre clientes y servidores, por medio de la disposición de un servidor de bases de datos, un protocolo intermedio y un cliente de base de datos. Así, se tienen los siguientes ejemplos:

- Servidor Web (Apache) - Protocolo - Cliente Web (Internet Explorer, Google Chrome).
- Servidor FTP (SFTP) - Protocolo - Cliente FTP (Cute FTP).
- Servidor IRC (IRC Server) - Protocolo - Cliente IRC (Mirc).

A.5.2. Paradigma P2P

Este tipo de diseño permite comunicar y compartir información entre clientes sin la necesidad de recurrir a un servidor central que facilite la comunicación. En palabras más técnicas, este modelo corresponde a una red de ordenadores, en la cual algunos o todos funcionan sin clientes o servidores fijos.

La gran ventaja de este modelo de comunicación es que aprovecha al máximo los recursos de los clientes (ancho de banda, capacidad de almacenamiento, entre otros), ofreciendo servicios de red sin tener que depender de los recursos de otros servidores. Además, no existe una entidad central que bloquee y colapse una red P2P, lo cual permite que esta arquitectura sea de gran robustez frente a la operación de múltiples servidores.

Considerando estas características del paradigma P2P, se encuentran los siguientes ejemplos donde se emplea este modelo de red:

- **Freenet:** red de distribución de información descentralizada, la cual trabaja por medio del uso en común del ancho de banda y el espacio de almacenamiento de los equipos que componen la red, permitiendo publicar o recuperar distintos tipos de información de los usuarios de manera anónima.
- **Napster:** servicio de distribución de archivos de música entre usuarios, por medio de la utilización de una gran red de servidores basados en la arquitectura P2P.
- **P2PSP:** protocolo de comunicación de la capa de aplicación utilizado en sistemas P2P, el cual permite la realización de streaming de archivos multimedia en Internet, logrando la reproducción sincronizada del contenido por parte de los usuarios.

A.6. Protocolos de enlace de datos en IoT

A.6.1. IEEE 802.15.4

A.6.1.1. Servicios de la capa MAC

La funcionalidad de la capa MAC en el estándar IEEE 802.15.4 se divide en dos partes: transferencia de datos, que se realiza mediante MCPS (MAC common part sublayer); y funciones de administración, que son realizadas por MLME (MAC layer management entity). La función del MCPS es transferir datos desde la capa de red a la capa MAC y viceversa. También transfiere paquetes de datos desde la capa MAC a la capa física y viceversa. La

función de la MLME es gestionar la capa MAC, la cual contiene una base de datos (MIB - MAC information base) con atributos relevantes para la capa.

La Figura A.1 muestra la arquitectura del estándar IEEE 802.15.4, específicamente la capa MAC y su interfaz con la capa de red y la capa física.

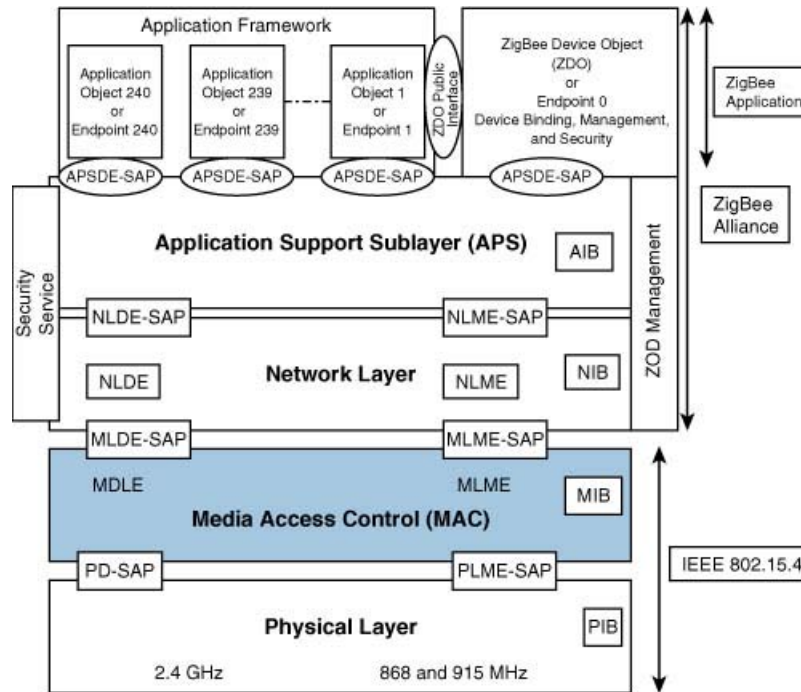


Figura A.1: Arquitectura IEEE 802.15.4 [9].

A.6.1.2. MAC Data Services

El MCPS proporciona servicios de datos utilizando las siguientes primitivas:

- MCPS Data Request:** esta primitiva es generada por la capa de red y emitida a la capa MAC para indicar la transferencia de la unidad de datos de protocolo de red (NPDU) a una capa MAC de destino. La capa MAC agrega su información de encabezado a la NPDU, y el paquete resultante se denomina unidad de datos del protocolo MAC (MPDU). La capa MAC envía la MPDU a la capa física, donde la capa física agrega su información de encabezado, formando un paquete denominado unidad de datos de protocolo de la capa física (PPDU). Finalmente, la capa física transmite la PPDU al destino. La estructura asociada a esta primitiva se presenta en la Figura A.2.

Source Address Mode	Source PANID	Source Address	Dest Address Mode	Dest PANID	Dest Address	MSDU Length	MSDU	MSDU Handle	TX Option	Security Level	Key ID Mode	Key Source	Key Index
---------------------	--------------	----------------	-------------------	------------	--------------	-------------	------	-------------	-----------	----------------	-------------	------------	-----------

Figura A.2: Estructura MCPS Data Request.

- **MCPS Data Confirm:** esta primitiva es generada por la capa MAC, y se envía a la capa de red en respuesta al MCPS Data Request. La estructura asociada a esta primitiva se presenta en la Figura A.3.

MSDU Handle	Status	Time Stamp
-------------	--------	------------

Figura A.3: Estructura MCPS Data Confirm.

- **MCPS Data Indication:** esta primitiva es generada por la capa MAC a la capa de red con el propósito de indicar la transferencia de la MSDU (unidad de datos del servicio MAC). La estructura asociada a esta primitiva se presenta en la Figura A.4.

Source Address Mode	Source PANID	Source Address	Dest Address Mode	Dest PANID	Dest Address	MSDU Length	MSDU	Link Quality	DSN	Security Level	Key ID Mode	Key Source	Key Index
---------------------	--------------	----------------	-------------------	------------	--------------	-------------	------	--------------	-----	----------------	-------------	------------	-----------

Figura A.4: Estructura MCPS Data Indication.

- **MCPS Purge Request:** esta primitiva es generada por una capa superior y se crea en la capa MAC para eliminar una unidad de datos del servicio MAC de la cola MAC.
- **MCPS Purge Confirm:** esta primitiva es generada por la capa MAC en respuesta a un MCPS Purge Request.

Referido al protocolo de transferencia de datos de la capa MAC, este se describe a continuación. El protocolo comienza con la transmisión de un **MCPS Data Request** desde la capa de red a la capa MAC. Luego, la capa MAC transmite a la capa física un **PD Data Request**. Cuando se completa la transmisión de la capa física, se emite un **PD Data Confirm** a la capa MAC. Si la capa MAC de origen activó el campo de transmisión, la capa MAC esperará recibir un paquete ACK del destino, durante un tiempo no mayor que la duración de espera ACK especificada en la MIB. Si la fuente no recibe una trama ACK, intentará retransmitir el dataframe; el número de intentos adicionales se especifica en la MIB como el valor máximo de reintentos del dataframe. Sin embargo, si la trama ACK no se recibe después de haber alcanzado el número máximo de intentos, se informa a la capa superior que el destino no está disponible. El proceso anteriormente descrito se puede visualizar en la Figura A.5.

A.6.2. IEEE 802.11ah

IEEE 802.11ah es una versión de bajo consumo de energía del estándar IEEE 802.11. Se ha diseñado con menos consumo energético para cumplir con los requisitos de IoT. Los estándares IEEE 802.11 son los estándares inalámbricos más utilizados, sin embargo, los estándares de Wi-Fi originales no son adecuados para aplicaciones de IoT debido al **frame overhead** y al consumo de energía. De esta manera, el grupo de trabajo IEEE 802.11 desarrolló un estándar que admite una comunicación de bajo costo y de fácil uso para sensores [21]. Las características básicas de la capa MAC incluyen:

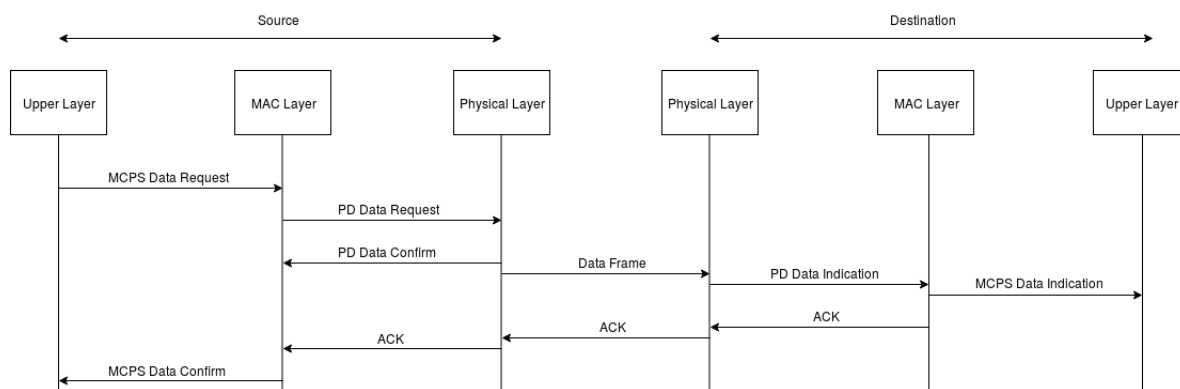


Figura A.5: Protocolo de transferencia de datos de la capa MAC, en IEEE 802.15.4 [9].

- **Frame de sincronización:** una estación no puede transmitir a menos que tenga información del canal que le permita detener el intercambio de paquetes por parte de otros nodos.
- **Intercambio de paquetes bidireccional eficiente:** esta característica permite que el sensor ahorre más energía al autorizar un **uplink** y un **downlink** entre el Access Point y el sensor.
- **Menor tamaño del MAC frame:** el tamaño normal de un frame del IEEE 802.11 es de unos 30 Bytes, lo cual es demasiado grande para las aplicaciones de IoT. IEEE 802.11ah mitiga este problema, definiendo una trama MAC corta con aproximadamente 12 Bytes.
- **Null data packet:** en IEEE 802.11, los frame de control tienen un tamaño aproximado de 14 Bytes, y no contienen datos, lo que produce una sobrecarga. IEEE 802.11ah mitiga este problema al reemplazar el frame de control con un preámbulo, correspondiente a una pequeña señal.
- **Incremento del sleep time:** IEEE 802.11ah está diseñado para sensores de baja potencia y, por lo tanto, permite un largo período de hibernación, activándose sólo para la transmisión de datos.

A.7. Infraestructura

A.7.1. Iluminación Pública: eficiencia energética

A.7.1.1. Barreras financieras

Los puntos claves de las barreras financieras son:

- Falta de comprensión de la inversión en alumbrado público EE
- Falta de confianza en las municipalidades por parte de las instituciones financieras.
- Presupuestos reducidos de las municipalidades.

- Falta de mecanismos innovadores de financiamiento, basados en el mercado, para implementar tecnologías EE.
- Inversionistas temerosos de invertir en proyectos públicos.
- Poca familiaridad de las instituciones financieras con evaluación de proyectos asociados a tecnologías EE.
- Falta de instrumentos financieros dedicados a tecnologías EE en el sector municipal.
- Tamaño propio de los proyectos (los cuales pueden ser pequeños).

A.7.1.2. Barreras de información

Los puntos claves de las barreras de información son:

- Falta de información y comprensión por parte de las municipalidades en cuanto a adoptar un consumo racional de los recursos energéticos, e invertir en la introducción de tecnologías EE.
- Falta de información organizada sobre la implementación de proyectos de alumbrado público con tecnologías EE, proveedores asociados, entre otros.
- Conocimiento técnico limitados de los desarrolladores de proyectos asociados a proyectos con tecnologías EE.

A.7.1.3. Barreras institucionales y regulaciones

Los puntos claves de las barreras institucionales y regulaciones son:

- Falta de mecanismos institucionales para alentar la inversión de tecnologías EE en municipalidades.
- Baja prioridad otorgada en el sector municipal a tecnologías EE, en relación a otros costos existentes.
- Políticas rígidas de compras y presupuestos
- Limitaciones al financiamiento público.

A.7.1.4. Barreras técnicas

Los puntos claves de las barreras técnicas son:

- Falta de sistemas efectivos de monitoreo y seguimiento de ahorro de energía.
- Capacidad insuficiente por parte de las municipalidades para la conceptualización, diseño e implementación de nuevas iniciativas.
- Poca claridad por parte de las municipalidades en cuanto a las propiedades asociadas a las tecnologías EE (ahorro energético y disminución de costos fijos).
- Habilidades de gestión de riesgos limitadas por parte de los proveedores de servicios

Apéndice B

Desarrollos

Código B.1: Procesamiento de resultados y generación de gráficos

```
1 import pandas as pd
2 import pandas
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import statistics
6 import matplotlib
7
8 list_total_received_frames = []
9 list_total_sent_frames = []
10 list_frame_delivery_ratio = []
11
12 list_endToendDelay_mean = []
13 list_endToendDelay_stddev = []
14 list_endToendDelay_min = []
15 list_endToendDelay_max = []
16
17 list_totalEnergyConsumed_mean = []
18 list_totalEnergyConsumed_stddev = []
19 list_totalEnergyConsumed_min = []
20 list_totalEnergyConsumed_max = []
21
22 list_InternetCloudQueue_mean = []
23 list_InternetCloudQueue_stddev = []
24 list_InternetCloudQueue_min = []
25 list_InternetCloudQueue_max = []
26
27 list_receivedRSSI_mean = []
28 list_receivedRSSI_stddev = []
29 list_receivedRSSI_min = []
30 list_receivedRSSI_max = []
31
32 for i in range (0, 18):
33     print (i+1)
34     lora = pd.read_csv(str(i+1) + '/lora.csv')
35
36     tmp = lora[(lora.type=='scalar') & (lora.name=='numSent')]
37     list_test = []
```

```

38     for index in tmp.value:
39         list_test.append(int(index))
40     total_frames = sum(list_test)
41
42     tmp = lora[(lora.type=='histogram') & (lora.name=='endToendDelay')]
43     received_frames = int(tmp['count'])
44     endToendDelay_mean = float(tmp['mean'])
45     endToendDelay_stddev = float(tmp['stddev'])
46     endToendDelay_min = float(tmp['min'])
47     endToendDelay_max = float(tmp['max'])
48
49     frame_delivery_ratio = received_frames/total_frames
50
51     tmp = lora[(lora.type=='scalar') & (lora.name=='totalEnergyConsumed')]
52     list_totalEnergyConsumed = []
53     for index in tmp.value:
54         list_totalEnergyConsumed.append(float(index))
55     totalEnergyConsumed_mean = sum(list_totalEnergyConsumed)/len(
list_totalEnergyConsumed)
56     totalEnergyConsumed_stddev = statistics.stdev(list_totalEnergyConsumed)
57     totalEnergyConsumed_min = min(list_totalEnergyConsumed)
58     totalEnergyConsumed_max = max(list_totalEnergyConsumed)
59
60     tmp = lora[(lora.type=='histogram') & (lora.name=='queueingTime:histogram'
) & (lora.module=='LoRaNetworkTest.internetCloud.ppp[0].queue')]
61     InternetCloudQueue_mean = float(tmp['mean'])
62     InternetCloudQueue_stddev = float(tmp['stddev'])
63     InternetCloudQueue_min = float(tmp['min'])
64     InternetCloudQueue_max = float(tmp['max'])
65
66     tmp = lora[(lora.type=='histogram') & (lora.name=='receivedRSSI')]
67     receivedRSSI_mean = float(tmp['mean'])
68     receivedRSSI_stddev = float(tmp['stddev'])
69     receivedRSSI_min = float(tmp['min'])
70     receivedRSSI_max = float(tmp['max'])
71
72     list_total_received_frames.append(received_frames)
73     list_total_sent_frames.append(total_frames)
74     list_frame_delivery_ratio.append(round(frame_delivery_ratio,4))
75
76     list_endToendDelay_mean.append(round(endToendDelay_mean,4))
77     list_endToendDelay_stddev.append(round(endToendDelay_stddev,4))
78     list_endToendDelay_min.append(round(endToendDelay_min,4))
79     list_endToendDelay_max.append(round(endToendDelay_max,4))
80
81     list_totalEnergyConsumed_mean.append(round(totalEnergyConsumed_mean,4))
82     list_totalEnergyConsumed_stddev.append(round(totalEnergyConsumed_stddev,4)
)
83     list_totalEnergyConsumed_min.append(round(totalEnergyConsumed_min,4))
84     list_totalEnergyConsumed_max.append(round(totalEnergyConsumed_max,4))
85
86     list_InternetCloudQueue_mean.append(round(InternetCloudQueue_mean,11))
87     list_InternetCloudQueue_stddev.append(round(InternetCloudQueue_stddev,11))
88     list_InternetCloudQueue_min.append(round(InternetCloudQueue_min,11))
89     list_InternetCloudQueue_max.append(round(InternetCloudQueue_max,11))
90

```

```

91 list_receivedRSSI_mean.append(round(receivedRSSI_mean,4))
92 list_receivedRSSI_stddev.append(round(receivedRSSI_stddev,4))
93 list_receivedRSSI_min.append(round(receivedRSSI_min,4))
94 list_receivedRSSI_max.append(round(receivedRSSI_max,4))
95
96 print("Frame Delivery Ratio:")
97 print(frame_delivery_ratio)
98
99 print("End to End Delay: ")
100 print(str(endToendDelay_mean) + ' [s]')
101
102 print("Total Energy Consumed per Node:")
103 print(str(totalEnergyConsumed_mean) + " [J]")
104
105 print("Internet Cloud Queue:")
106 print(str(InternetCloudQueue_mean) + ' [s]')
107
108 print("Received RSSI:")
109 print(str(receivedRSSI_mean) + ' [dBm]')
110
111 print('')
112
113 #*****#
114
115 text_file = open("Output-FrameDeliveryRatio.txt", "w")
116 text_file.write("Frame Delivery Ratio")
117 text_file.write("\nTotal Sent Frames\tTotal Received Frames\tFrame Delivery
Ratio")
118 index_order = [4, 5, 6, 10, 11, 12, 16, 17, 18, 1, 2, 3, 7, 8, 9, 13, 14, 15]
119 for index in index_order:
120     if (index == 1):
121         text_file.write("\n")
122         text_file.write("\n" + str(list_total_sent_frames[index-1]) + "\t" + str(
list_total_received_frames[index-1]) + "\t" + str(list_frame_delivery_ratio
[index-1]))
123
124 text_file.close()
125
126 text_file = open("Output-EndToEnd.txt", "w")
127 text_file.write("End to End Delay - Unit: [s]")
128 text_file.write("\nMean\tStd Dev\tMin\tMax")
129 index_order = [4, 5, 6, 10, 11, 12, 16, 17, 18, 1, 2, 3, 7, 8, 9, 13, 14, 15]
130 for index in index_order:
131     if (index == 1):
132         text_file.write("\n")
133         text_file.write("\n" + str(list_endToendDelay_mean[index-1]) + "\t" + str(
list_endToendDelay_stddev[index-1]) + "\t" + str(list_endToendDelay_min[
index-1]) + "\t" + str(list_endToendDelay_max[index-1]))
134
135 text_file.close()
136
137 text_file = open("Output-EnergyPerNode.txt", "w")
138 text_file.write("Total Energy Consumed per Node - Unit: [J]")
139 text_file.write("\nMean\tStd Dev\tMin\tMax")
140 index_order = [4, 5, 6, 10, 11, 12, 16, 17, 18, 1, 2, 3, 7, 8, 9, 13, 14, 15]
141 for index in index_order:

```

```

142     if (index == 1):
143         text_file.write("\n")
144         text_file.write("\n" + str(list_totalEnergyConsumed_mean[index-1]) + "\t"
+ str(list_totalEnergyConsumed_stddev[index-1]) + "\t" + str(
list_totalEnergyConsumed_min[index-1]) + "\t" + str(
list_totalEnergyConsumed_max[index-1]))
145
146 text_file.close()
147
148 text_file = open("Output-InternetCloudQueue.txt", "w")
149 text_file.write("Internet Cloud Queue - Unit: [s]")
150 text_file.write("\nMean\tStd Dev\tMin\tMax")
151 index_order = [4, 5, 6, 10, 11, 12, 16, 17, 18, 1, 2, 3, 7, 8, 9, 13, 14, 15]
152 for index in index_order:
153     if (index == 1):
154         text_file.write("\n")
155
156     IC_mean = '{:0=5.4f}e-07'.format(list_InternetCloudQueue_mean[index-1]/1e
-07)
157     IC_stddev = '{:0=5.4f}e-07'.format(list_InternetCloudQueue_stddev[index
-1]/1e-07)
158     IC_min = '{:0=5.4f}e-07'.format(list_InternetCloudQueue_min[index-1]/1e
-07)
159     IC_max = '{:0=5.4f}e-07'.format(list_InternetCloudQueue_max[index-1]/1e
-07)
160     text_file.write("\n" + IC_mean + "\t" + IC_stddev + "\t" + IC_min + "\t" +
IC_max)
161
162 text_file.close()
163
164 text_file = open("Output-ReceivedRSSI.txt", "w")
165 text_file.write("Received RSSI - Unit: [dBm]")
166 text_file.write("\nMin\tMax\tMean\tStd Dev")
167 index_order = [4, 5, 6, 10, 11, 12, 16, 17, 18, 1, 2, 3, 7, 8, 9, 13, 14, 15]
168 for index in index_order:
169     if (index == 1):
170         text_file.write("\n")
171         text_file.write("\n" + str(list_receivedRSSI_mean[index-1]) + "\t" + str(
list_receivedRSSI_stddev[index-1]) + "\t" + str(list_receivedRSSI_min[index
-1]) + "\t" + str(list_receivedRSSI_max[index-1]))
172
173 text_file.close()
174
175 #*****#
176
177 #Frame Delivery Ratio
178 #Overload traffic conditions
179
180 matplotlib.rcParams['text.usetex'] = True
181 matplotlib.rcParams['text.latex.unicode'] = True
182
183 # set width of bar
184 barWidth = 0.25
185
186 # set height of bar
187 sigma_0 = [list_frame_delivery_ratio[0], list_frame_delivery_ratio[6],

```

```

188 list_frame_delivery_ratio[12]]
189 sigma_1785 = [list_frame_delivery_ratio[1], list_frame_delivery_ratio[7],
190               list_frame_delivery_ratio[13]]
191 sigma_357 = [list_frame_delivery_ratio[2], list_frame_delivery_ratio[8],
192              list_frame_delivery_ratio[14]]
193
194 # Set position of bar on X axis
195 r1 = np.arange(len(sigma_0))
196 r2 = [x + barWidth for x in r1]
197 r3 = [x + barWidth for x in r2]
198
199 # Make the plot
200 plt.bar(r1, sigma_0, color='y', width=barWidth, edgecolor='white', label=r'\
201 sigma'+'+=0'+'+[dB]')
202 plt.bar(r2, sigma_1785, color='m', width=barWidth, edgecolor='white', label=r'\
203 sigma'+'+=1.785'+'+[dB]')
204 plt.bar(r3, sigma_357, color='c', width=barWidth, edgecolor='white', label=r'\
205 sigma'+'+=3.57'+'+[dB]')
206
207 # Add xticks on the middle of the group bars
208 plt.ylabel('Frame Delivery Ratio', fontweight='bold')
209 plt.xticks([r + barWidth for r in range(len(sigma_0))], ['Topolog a A', '
210 Topolog a B', 'Topolog a C'])
211
212 #Prueba de carga
213 plt.title('Comparativa de Topolog as en Sobrecarga de Tr fico')
214 plt.ylim([0, 1])
215
216 # Create legend & Show graphic
217 plt.legend()
218 plt.savefig('frame_delivery_ratio_sobrecarga.png', bbox_inches='tight')
219 plt.show()
220
221 #Normal conditions
222
223 # set width of bar
224 barWidth = 0.25
225
226 # set height of bar
227 sigma_0 = [list_frame_delivery_ratio[3], list_frame_delivery_ratio[9],
228            list_frame_delivery_ratio[15]]
229 sigma_1785 = [list_frame_delivery_ratio[4], list_frame_delivery_ratio[10],
230              list_frame_delivery_ratio[16]]
231 sigma_357 = [list_frame_delivery_ratio[5], list_frame_delivery_ratio[11],
232              list_frame_delivery_ratio[17]]
233
234 # Set position of bar on X axis
235 r1 = np.arange(len(sigma_0))
236 r2 = [x + barWidth for x in r1]
237 r3 = [x + barWidth for x in r2]
238
239 # Make the plot
240 plt.bar(r1, sigma_0, color='y', width=barWidth, edgecolor='white', label=r'\
241 sigma'+'+=0'+'+[dB]')
242 plt.bar(r2, sigma_1785, color='m', width=barWidth, edgecolor='white', label=r'\
243 sigma'+'+=1.785'+'+[dB]')

```

```

232 plt.bar(r3, sigma_357, color='c', width=barWidth, edgecolor='white', label=r'\
sigma '+'=3.57 '+'[dB] ')
233
234 # Add xticks on the middle of the group bars
235 plt.xticks([r + barWidth for r in range(len(sigma_0))], ['Topolog a A', '
Topolog a B', 'Topolog a C'])
236
237 plt.title('Comparativa de Topolog as en Condiciones Normales de Tr fico')
238 plt.ylabel('Frame Delivery Ratio', fontweight='bold')
239 plt.ylim([0, 1])
240
241 # Create legend & Show graphic
242 plt.legend()
243 plt.savefig('frame_delivery_ratio_normal.png', bbox_inches='tight')
244 plt.show()
245
246 #*****#
247
248 #End to End Delay
249 #Overload traffic conditions
250
251 sigma_0 = [list_endToendDelay_mean[12], list_endToendDelay_mean[6],
list_endToendDelay_mean[0]]
252 sigma_1785 = [list_endToendDelay_mean[13], list_endToendDelay_mean[7],
list_endToendDelay_mean[1]]
253 sigma_357 = [list_endToendDelay_mean[14], list_endToendDelay_mean[8],
list_endToendDelay_mean[2]]
254
255 df = pandas.DataFrame(dict(graph=['Topologia C', 'Topologia B', 'Topologia A'
],
256 s357 = sigma_357, s1785 = sigma_1785, s0 = sigma_0)
)
257
258 ind = np.arange(len(df))
259 width = 0.25
260
261 fig, ax = plt.subplots()
262
263 ax.barh(ind + 2*width, df.s0, width, color='y', label=r'\sigma '+'=0 '+'[dB] ')
264 ax.barh(ind + width, df.s1785, width, color='m', label=r'\sigma '+'=1.785 '+'[dB
] ')
265 ax.barh(ind, df.s357, width, color='c', label=r'\sigma '+'=3.57 '+'[dB] ')
266
267 ax.set(yticks=ind + width, yticklabels=df.graph, ylim=[2*width - 1, len(df)])
268 ax.legend()
269
270 plt.title('Comparativa de Topolog as en Sobrecarga de Tr fico')
271 plt.xlabel('End To End Delay [s]', fontweight='bold')
272
273 plt.xlim([0, 2.6])
274 plt.savefig('end_to_end_delay_sobrecarga.png', bbox_inches='tight')
275 plt.show()
276
277 #Normal conditions
278
279 sigma_0 = [list_endToendDelay_mean[15], list_endToendDelay_mean[9],

```

```

list_endToendDelay_mean[3]]
280 sigma_1785 = [list_endToendDelay_mean[16], list_endToendDelay_mean[10],
list_endToendDelay_mean[4]]
281 sigma_357 = [list_endToendDelay_mean[17], list_endToendDelay_mean[11],
list_endToendDelay_mean[5]]
282
283 df = pandas.DataFrame(dict(graph=['Topologia C', 'Topologia B', 'Topologia A'
],
284 s357 = sigma_357, s1785 = sigma_1785, s0 = sigma_0)
)
285
286 ind = np.arange(len(df))
287 width = 0.25
288
289 fig, ax = plt.subplots()
290
291 ax.barh(ind + 2*width, df.s0, width, color='y', label=r'\sigma'+ '=0'+ '[dB]')
292 ax.barh(ind + width, df.s1785, width, color='m', label=r'\sigma'+ '=1.785'+ '[dB]')
293 ax.barh(ind, df.s357, width, color='c', label=r'\sigma'+ '=3.57'+ '[dB]')
294
295 ax.set(yticks=ind + width, yticklabels=df.graph, ylim=[2*width - 1, len(df)])
296 ax.legend()
297
298 plt.title('Comparativa de Topolog as en Condiciones Normales')
299 plt.xlabel('End To End Delay [s]', fontweight='bold')
300
301 plt.xlim([0, 2.6])
302 plt.savefig('end_to_end_delay_normal.png', bbox_inches='tight')
303 plt.show()
304
305 #*****#
306
307 #Received RSSI
308 #Overload traffic conditions
309
310 # set width of bar
311 barWidth = 0.25
312
313 # set height of bar
314 sigma_0 = [list_receivedRSSI_mean[0], list_receivedRSSI_mean[6],
list_receivedRSSI_mean[12]]
315 sigma_1785 = [list_receivedRSSI_mean[1], list_receivedRSSI_mean[7],
list_receivedRSSI_mean[13]]
316 sigma_357 = [list_receivedRSSI_mean[2], list_receivedRSSI_mean[8],
list_receivedRSSI_mean[14]]
317
318 # Set position of bar on X axis
319 r1 = np.arange(len(sigma_0))
320 r2 = [x + barWidth for x in r1]
321 r3 = [x + barWidth for x in r2]
322
323 # Make the plot
324 plt.bar(r1, sigma_0, color='y', width=barWidth, edgecolor='white', label=r'\
sigma'+ '=0'+ '[dB]')
325 plt.bar(r2, sigma_1785, color='m', width=barWidth, edgecolor='white', label=r'

```

```

326 plt.bar(r3, sigma_357, color='c', width=barWidth, edgecolor='white', label=r'\
    sigma '+'=3.57 '+'[dB] ')
327
328 # Add xticks on the middle of the group bars
329 plt.ylabel('Received RSSI [dBm]', fontweight='bold')
330 plt.xticks([r + barWidth for r in range(len(sigma_0))], ['Topolog a A', '
    Topolog a B', 'Topolog a C'])
331
332 #Prueba de carga
333 plt.title('Comparativa de Topolog as en Sobrecarga de Tr fico')
334 plt.ylim([-150, 0])
335
336 # Create legend & Show graphic
337 plt.legend()
338 plt.savefig('received_rssi_sobrecarga.png', bbox_inches='tight')
339 plt.show()
340
341 #Normal conditions
342
343 # set width of bar
344 barWidth = 0.25
345
346 # set height of bar
347 sigma_0 = [list_receivedRSSI_mean[3], list_receivedRSSI_mean[9],
    list_receivedRSSI_mean[15]]
348 sigma_1785 = [list_receivedRSSI_mean[4], list_receivedRSSI_mean[10],
    list_receivedRSSI_mean[16]]
349 sigma_357 = [list_receivedRSSI_mean[5], list_receivedRSSI_mean[11],
    list_receivedRSSI_mean[17]]
350
351 # Set position of bar on X axis
352 r1 = np.arange(len(sigma_0))
353 r2 = [x + barWidth for x in r1]
354 r3 = [x + barWidth for x in r2]
355
356 # Make the plot
357 plt.bar(r1, sigma_0, color='y', width=barWidth, edgecolor='white', label=r'\
    sigma '+'=0 '+'[dB] ')
358 plt.bar(r2, sigma_1785, color='m', width=barWidth, edgecolor='white', label=r'
    \sigma '+'=1.785 '+'[dB] ')
359 plt.bar(r3, sigma_357, color='c', width=barWidth, edgecolor='white', label=r'\
    sigma '+'=3.57 '+'[dB] ')
360
361 # Add xticks on the middle of the group bars
362 plt.ylabel('Received RSSI [dBm]', fontweight='bold')
363 plt.xticks([r + barWidth for r in range(len(sigma_0))], ['Topolog a A', '
    Topolog a B', 'Topolog a C'])
364
365 #Prueba de carga
366 plt.title('Comparativa de Topolog as en Condiciones Normales')
367 plt.ylim([-150, 0])
368
369 # Create legend & Show graphic
370 plt.legend()
371 plt.savefig('received_rssi_normal.png', bbox_inches='tight')

```



```

372 plt.show()
373
374 #*****#
375
376 #Internet Cloud Queue
377 #Overload traffic conditions
378
379 sigma_0 = [list_InternetCloudQueue_mean[12], list_InternetCloudQueue_mean[6],
            list_InternetCloudQueue_mean[0]]
380 sigma_1785 = [list_InternetCloudQueue_mean[13], list_InternetCloudQueue_mean
              [7], list_InternetCloudQueue_mean[1]]
381 sigma_357 = [list_InternetCloudQueue_mean[14], list_InternetCloudQueue_mean
              [8], list_InternetCloudQueue_mean[2]]
382
383 df = pandas.DataFrame(dict(graph=['Topologia C', 'Topologia B', 'Topologia A'
384                                ],
385                                s357 = sigma_357, s1785 = sigma_1785, s0 = sigma_0)
386 )
387
388 ind = np.arange(len(df))
389 width = 0.25
390
391 fig, ax = plt.subplots()
392
393 ax.barh(ind + 2*width, df.s0, width, color='y', label=r'\sigma'+ '=0'+ '[dB]')
394 ax.barh(ind + width, df.s1785, width, color='m', label=r'\sigma'+ '=1.785'+ '[dB]')
395 ax.barh(ind, df.s357, width, color='c', label=r'\sigma'+ '=3.57'+ '[dB]')
396
397 ax.set(yticks=ind + width, yticklabels=df.graph, ylim=[2*width - 1, len(df)])
398 ax.legend()
399
400 plt.title('Comparativa de Topologías en Sobrecarga de Tráfico')
401 plt.xlabel('Internet Cloud Queue [s]', fontweight='bold')
402
403 plt.xlim([0, 1.8e-7])
404 plt.savefig('internet_cloud_queue_sobrecarga.png', bbox_inches='tight')
405 plt.show()
406
407 #Normal conditions
408
409 sigma_0 = [list_InternetCloudQueue_mean[15], list_InternetCloudQueue_mean[9],
            list_InternetCloudQueue_mean[3]]
410 sigma_1785 = [list_InternetCloudQueue_mean[16], list_InternetCloudQueue_mean
              [10], list_InternetCloudQueue_mean[4]]
411 sigma_357 = [list_InternetCloudQueue_mean[17], list_InternetCloudQueue_mean
              [11], list_InternetCloudQueue_mean[5]]
412
413 df = pandas.DataFrame(dict(graph=['Topologia C', 'Topologia B', 'Topologia A'
414                                ],
415                                s357 = sigma_357, s1785 = sigma_1785, s0 = sigma_0)
416 )
417
418 ind = np.arange(len(df))
419 width = 0.25

```

```

417 fig, ax = plt.subplots()
418
419 ax.barh(ind + 2*width, df.s0, width, color='y', label=r'\sigma'+ '=0'+ '[dB]')
420 ax.barh(ind + width, df.s1785, width, color='m', label=r'\sigma'+ '=1.785'+ '[dB
421 |')
421 ax.barh(ind, df.s357, width, color='c', label=r'\sigma'+ '=3.57'+ '[dB]')
422
423 ax.set(yticks=ind + width, yticklabels=df.graph, ylim=[2*width - 1, len(df)])
424 ax.legend()
425
426 plt.title('Comparativa de Topolog as en Condiciones Normales')
427 plt.xlabel('Internet Cloud Queue [s]', fontweight='bold')
428
429 plt.xlim([0, 1.8e-7])
430 plt.savefig('internet_cloud_queue_normal.png', bbox_inches='tight')
431 plt.show()

```