

La investigación reportada en esta tesis es parte de los programas de investigación del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California).

La investigación fue financiada por el CONAHCYT (Consejo Nacional de Ciencias Humanidades y Tecnología).

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México). El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo o titular de los Derechos Autor.

**Centro de Investigación Científica y de Educación  
Superior de Ensenada, Baja California**



---

**Maestría en Ciencias  
Electrónica y Telecomunicaciones  
con orientación en Telecomunicaciones**

---

**Evaluación de algoritmos broadcast sobre redes ad-hoc con  
ciclos de trabajo.**

Tesis  
para cubrir parcialmente los requisitos necesarios para obtener el grado de  
Maestro en Ciencias

Presenta:

**Carlos David Nimo Trujillo**

Ensenada, Baja California, México  
2024

Tesis defendida por  
**Carlos David Nimo Trujillo**

y aprobada por el siguiente Comité

---

**Dr. Gabriel Alejandro Galaviz Mosqueda**

Codirector de tesis

---

**Dr. Salvador Villarreal Reyes**

Codirector de tesis

Miembros del Comité

**Dr. Raúl Rivera Rodríguez**

**Dr. Ismael Edrein Espinosa Curiel**

**Dr. José Lozano Risk**

**MC. Enrique Guerrero Arbona**



---

**Dra. María del Carmen Maya Sánchez**

Coordinadora del Posgrado en Electrónica y Telecomunicaciones

---

**Dra. Ana Denise Re Araujo**

Directora de Estudios de Posgrado

Resumen de la tesis que presenta **Carlos David Nimo Trujillo** como requisito parcial para la obtención del grado de Maestro en Ciencias en Electrónica y Telecomunicaciones con orientación en Telecomunicaciones

### **Evaluación de algoritmos broadcast sobre redes ad-hoc con ciclos de trabajo**

Resumen aprobado por:

---

Dr. Gabriel Alejandro Galaviz Mosqueda  
Codirector de tesis

---

Dr. Salvador Villarreal Reyes  
Codirector de tesis

La diseminación de paquetes tipo broadcast es fundamental en las redes de comunicación, ya que son la base de procesos cruciales como la conformación de la topología de red, la actualización de los nodos y la difusión de información en tiempo real. Un problema importante en la diseminación de paquetes broadcast es el consumo de energía, debido que se busca que todos los nodos estén informados. Este problema es de mayor relevancia en Redes de Sensores Inalámbricos (WSN por sus siglas en inglés), donde la eficiencia energética es esencial por las limitaciones de batería que alimentan los nodos. En este marco, se introducen dos mecanismos importantes: los ciclos de trabajo y los algoritmos de diseminación broadcast. Los ciclos de trabajo son una estrategia de capa física para prolongar las baterías en los nodos en WSN. Este enfoque implica alternar períodos de actividad y descanso para minimizar el consumo de energía, siendo el módulo de radio el principal responsable del agotamiento energético en estos nodos. Los algoritmos de diseminación broadcast son una estrategia de capa 3 para disminuir el número de nodos retransmisores, mitigando el problema del broadcast storm. Sin embargo, la identificación del impacto que los ciclos de trabajo pueden tener en los algoritmos de transmisión es un campo de investigación en desarrollo. Este trabajo de investigación se enfoca en determinar cómo impactan los ciclos de trabajo en los algoritmos de broadcast para WSN. Para el estudio realizado se utilizó un simulador de redes previamente validado (Cardenas, 2019; Cruz, 2017) que se fundamenta en el estándar de comunicación IEEE 802.15.4, utilizado en WSN. La evaluación del impacto de los ciclos de trabajo en los algoritmos de broadcast se realiza utilizando las métricas de retardo en la transmisión (**Delay**), la capacidad de alcanzar nodos receptores (**Reachability**) y la eficacia en la evitación de retransmisiones innecesarias (**SRB**). Los resultados proporcionarán una comprensión de la interacción entre los ciclos de trabajo y los algoritmos de broadcast en WSN, lo que podría tener implicaciones para el diseño y la implementación de estas redes, especialmente en términos de conservación de energía y optimización de la comunicación.

**Palabras clave:** Broadcast, Ciclos de trabajo, WSN, Métricas de desempeño

Abstract of the thesis presented by **Carlos David Nimo Trujillo** as a partial requirement to obtain the Master of Science degree in Electronics and Telecommunications with orientation in Telecommunications

### **Evaluation of broadcast algorithms on ad-hoc networks with duty cycles**

Abstract approved by:

---

Dr. Gabriel Alejandro Galaviz Mosqueda  
Codirector de tesis

---

Dr. Salvador Villarreal Reyes  
Codirector de tesis

The dissemination of broadcast-type packets is fundamental in communication networks, as they form the basis for crucial processes such as network topology formation, node updates, and real-time information dissemination. An important issue in broadcast packet dissemination is energy consumption, as it is essential for ensuring that all nodes are informed. This problem is of greater relevance in Wireless Sensor Networks (WSN), where energy efficiency is critical due to the battery limitations that power the nodes. In this context, two important mechanisms are introduced: duty cycles and broadcast dissemination algorithms. Duty cycles are a physical layer strategy to prolong the batteries in WSN nodes. This approach involves alternating periods of activity and rest to minimize energy consumption, with the radio module being the main contributor to energy depletion in these nodes. Broadcast dissemination algorithms are a layer 3 strategy to reduce the number of relay nodes, mitigating the broadcast storm problem. However, understanding the impact that duty cycles can have on transmission algorithms is a developing research field. This research focuses on determining how duty cycles impact broadcast algorithms for WSN. For the conducted study, a previously validated network simulator (Cardenas, 2019; Cruz, 2017) based on the IEEE 802.15.4 communication standard used in WSN was employed. The evaluation of the impact of duty cycles on broadcast algorithms is performed using metrics such as transmission delay (**Delay**), the ability to reach receiving nodes (**Reachability**), and efficiency in avoiding unnecessary retransmissions (**SRB**). The results will provide an understanding of the interaction between duty cycles and broadcast algorithms in WSN, potentially influencing the design and implementation of these networks, especially in terms of energy conservation and communication optimization.

**Keywords: Broadcast, Duty Cycles, WSN, Performance Metrics**

## **Dedicatoria**

A mi madre **Santa Margarita Trujillo Hernández** y a mi abuela **Orfelina Hernández Pérez** por siempre impulsarme y motivarme a crecer como persona y superarme profesionalmente.

**-Never Give Up-**

**Sergio (Checo) Pérez**

## Agradecimientos

Doy gracias primeramente al Consejo Nacional de Humanidades Ciencia y Tecnologías (CONAHCYT) con su número de becario 811289 por el apoyo económico proporcionado durante estos dos años que duraron mis estudios de maestría.

Al Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CICESE) por abrirme las puertas y acogerme en su seno por más de dos años junto con la Unidad Foránea de Monterrey donde he encontrado conocimientos y a partir de ahí seguir creciendo científicamente, también al Proyecto de Ciencia básica CONAHCYT CB-2016-285276 por permitirme usar sus recursos computacionales para mi trabajo de investigación.

Agradecer inmensamente y de manera sincera a mis directores de tesis, al Dr. Gabriel Alejandro Galaviz Mosqueda y al Dr. Salvador Villarreal Reyes, por sus aportes de conocimiento y experiencias, por la dedicación y seguimiento que tuvieron ambos en todo el proceso, no solo de investigación sino también como académicos, por la palmadita en la espalda y el empujón a seguir en los momentos difíciles, por abrirme las puertas del conocimiento y lograr este reto en mi vida que es convertirme en Maestro en Ciencias.

A los miembros del comité de tesis Dr. Raúl Rivera Rodríguez, Dr. José Lozano Risk, Dr. Ismael Edrein Espinosa Curiel y al M. en C. Enrique Guerrero Arbona por sus consejos, comentarios y observaciones durante todo el proceso de investigación las que ayudaron a lograr un trabajo más prolijo.

A los profesores del posgrado de Electrónica y Telecomunicaciones que durante estos dos años me compartieron sus enseñanzas y me abrieron el panorama científico de las Telecomunicaciones y la Electrónica.

A mis padres, abuela, hermano y familia por apoyarme en esta etapa de mi vida la cual representó un salto enorme al alejarme de ellos y de mis amigos, al venir a un país nuevo con una cultura distinta.

A mis compañeros y amigos que realicé durante el posgrado, muchas gracias por todos los momentos compartidos, ustedes han sido de ayuda para mi crecimiento profesional, especial agradecimiento a Alfredo Ramírez, Itzel Mota, Paulina Díaz, Gerardo Soto, Carlos Villalobos, Daniel Gastelum, Arlen González.

Por último, pero no menos importante a los amigos que realicé durante toda mi estancia tanto en Ensenada como en Monterrey y los lugares que tuve la oportunidad de visitar en este hermoso país, muchas gracias por dejarme ser parte de su vida: Miguel, Jenny, Camila, Erica, Néstor, Eduardo, Maricela, Jorge, Malena, Sergio Villa, Rodrigo González Andrea, Fernando, Rodrigo, Alexis, Martín, Iliana, Amir, Daniel, Zeidel, Enrique y Elvis.



## Tabla de contenido

	Página
Resumen en español.....	ii
Resumen en inglés.....	iii
Dedicatoria .....	iv
Agradecimientos.....	v
Lista de figuras.....	x
Lista de tablas .....	xiv
<b>Capítulo 1. Introducción.....</b>	<b>1</b>
1.1 Planteamiento del problema .....	3
1.2 Justificación .....	4
1.3 Objetivos .....	4
1.3.1 Objetivo general .....	4
1.3.2 Objetivos específicos.....	5
1.4 Antecedentes .....	5
1.5 Metodología .....	8
<b>Capítulo 2. Marco Teórico .....</b>	<b>10</b>
2.1 Tecnología de WSN .....	10
2.1.1 Capa Física (PHY) .....	11
2.1.2 Capa MAC.....	13
2.1.2.1 CSMA/CA No ranurado del protocolo 802.15.4 .....	14
2.1.3 Ciclo de trabajo .....	16
2.1.4 6LoWPAN.....	17
2.2 Simuladores de redes de telecomunicaciones.....	18

2.2.1	Contiki .....	20
2.2.1.1	COOJA.....	21
2.2.1.2	UDGM (Unit Disk Graph Medium).....	23
2.3	ContikiMAC.....	24
2.3.1	ContikiMAC Timing.....	26
2.3.2	Detección de paquetes y sueño rápido (fast sleep).....	27
2.4	Algoritmos de diseminación broadcast.....	29
2.4.1	Técnicas de mitigación del broadcast storm.....	30
2.4.1.1	Esquemas de probabilidad de reenvío fija.....	31
2.4.1.2	Esquemas probabilísticos adaptativos.....	34
2.4.1.3	Esquemas basados en contadores.....	37
2.5	Métricas de desempeño.....	39
2.5.1	Reachability.....	40
2.5.2	Saved Rebroadcast (SRB).....	40
2.5.3	Delay.....	41
2.6	Método de Montecarlo.....	42
<b>Capítulo 3. Validación y Configuración del Simulador.....</b>		<b>44</b>
3.1	Escenario.....	44
3.2	Validación de la cama de pruebas.....	46
3.2.1	Escenario de validación.....	47
3.3	Simulación.....	51
3.3.1	Configuración de la capa Física.....	51
3.3.2	Configuración de la capa MAC.....	52
3.3.3	Configuración de la capa RDC.....	52
3.3.4	Proceso de simulación.....	53

<b>Capítulo 4. Resultados y Discusión.....</b>	<b>56</b>
4.1 Análisis de resultados para algoritmo GOSSIP1(p).....	56
4.2 Análisis de resultados para algoritmo DAPF .....	61
4.3 Análisis de resultados para algoritmo Counter-Based Scheme. ....	66
<b>Capítulo 5. Conclusiones, Contribuciones y Trabajo futuro .....</b>	<b>70</b>
5.1 Conclusiones.....	70
5.2 Aportaciones del trabajo de investigación.....	71
5.3 Trabajo futuro .....	72
<b>Literatura Citada.....</b>	<b>74</b>
<b>Anexos .....</b>	<b>81</b>

## Lista de figuras

Figura	Página
1. Representación de topología estrella y malla(CARDENAS, 2019).....	6
2. Arquitectura típica de un nodo de una WSN. ....	10
3. Mecanismo de CSMA/CA no ranurado(CARDENAS, 2019) .....	15
4. Descripción general del mecanismo de los ciclos de trabajo en WSN.....	17
5. Stack de protocolos de Contiki.....	21
6. Diagrama de acción para el transmisor y el receptor usando ContikiMAC en transmisiones de tipo unicast.....	25
7. Diagrama de acción para el transmisor y el receptor usando ContikiMAC en transmisiones de tipo broadcast .....	25
8. Distribución de los timing de ContikiMAC. ....	26
9. El tiempo de transmisión de un paquete $t_s$ debe ser lo suficientemente larga para que no se encuentre entre dos CCA. ....	27
10. Bloque de transmisión de fase, el transmisor aprende sobre las fases donde se despierta el receptor y la acomoda a la siguiente transmisión, lo cual causa que el transmisor necesite enviar menos paquetes.....	29
11. Diagrama de flujo ( <b>Flooding</b> ). ....	32
12. Diagrama de flujo GOSSIP1(p).....	33
13. Pseudo Código simple counter-based.....	37
14. Escenario disperso 10 nodos.....	45
15. Escenario disperso 20 nodos.....	45
16. Escenario disperso 30 nodos.....	45
17. Escenario disperso 40 nodos.....	45
18. Escenario disperso 50 nodos.....	45
19. Escenario denso 10 nodos.....	45
20. Escenario denso 20 nodos.....	46
21. Escenario denso 30 nodos.....	46

22. Escenario denso 40 nodos.....	46
23. Escenario denso 50 nodos.....	46
24. Topología de validación de la cama de pruebas. ....	48
25. Comparativa de pérdida de paquetes promedio entre los simuladores OMNET++ y COOJA .	51
26. Ejemplo de datos mostrados por la herramienta <b>Mote Output</b> de COOJA.....	54
27. Resultados obtenidos del algoritmo GOSSIP(p), en donde el <b>Delay</b> $p = 0.8$ , el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un <b>Delay</b> más pequeño siendo un resultado deseado. ....	57
28. Ejemplo <b>Time Delay</b> .....	58
29. Resultados obtenidos del algoritmo GOSSIP(p), en donde la <b>Reachability</b> $p = 0.8$ , el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una <b>Reachability</b> más grande siendo un resultado deseado. ....	59
30. Ejemplo <b>Timeline Reachability</b> .....	61
31. Resultado de la métrica <b>Saved Rebroadcast</b> para el esquema GOSSIP1(p). ....	61
32. Resultados obtenidos del algoritmo DAPF, en donde <b>Delay</b> $k=3$ , el eje X se refiere a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más claro es significado de un <b>Delay</b> más pequeño siendo un resultado deseado.....	62
33. Resultados obtenidos del algoritmo DAPF, en donde la <b>Reachability</b> $k = 3$ , el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una <b>Reachability</b> más grande siendo un resultado deseado.....	63
34. a-) Rango de Tx-Rx de los nodos. b-) Captura Timeline COOJA. ....	64
35. a-) Gráfico de bigotes <b>Delay</b> . b-) Gráfico de bigotes <b>Reachability</b> . ....	65
36. Resultado de la métrica <b>Saved Rebroadcast</b> para el esquema DAPF.....	66
37. Resultados obtenidos del algoritmo CBS, en donde <b>Delay</b> $Cth=2$ $RAD=0.001$ , el eje X se refiere a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más claro es significado de un <b>Delay</b> más pequeño siendo un resultado deseado. ....	67
38. Resultados obtenidos del algoritmo CBS, en donde la <b>Reachability</b> $Cth=2$ $RAD=0.001$ , el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una <b>Reachability</b> más grande siendo un resultado deseado. ....	68
39. Resultado de la métrica <b>Saved Rebroadcast</b> $Cth=2$ $RAD=0.001$ .....	69
40. Resultados obtenidos del algoritmo GOSSIP(p) en escenarios densos, en donde el <b>Delay</b> $p = 0.6$ , $p=0.8$ , $p=1$ respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un <b>Delay</b> más pequeño siendo un resultado deseado.....	81

41. Resultados obtenidos del algoritmo GOSSIP(p) en escenarios dispersos, en donde el **Delay**  $p = 0.6, p=0.8, p=1$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado..... 81
42. Resultados obtenidos del algoritmo GOSSIP(p) en escenarios densos, en donde la **Reachability**  $p = 0.6, p=0.8, p=1$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado..... 82
43. Resultados obtenidos del algoritmo GOSSIP(p) en escenarios dispersos, en donde la **Reachability**  $p = 0.6, p=0.8, p=1$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado..... 82
44. Resultado de la métrica **Saved Rebroadcast** para el esquema GOSSIP1(p) para  $p=0.6, p=0.8, p=1$  respectivamente para escenarios densos..... 83
45. Resultado de la métrica **Saved Rebroadcast** para el esquema GOSSIP1(p) para  $p=0.6, p=0.8, p=1$  respectivamente para escenarios dispersos..... 83
46. Resultados obtenidos del algoritmo DAPF en escenarios densos, en donde el **Delay**  $k=2, k=3, k=5$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado..... 84
47. Resultados obtenidos del algoritmo DAPF en escenarios disperso, en donde el **Delay**  $k=2, k=3, k=5$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado..... 84
48. Resultados obtenidos del algoritmo DAPF en escenarios densos, en donde la **Reachability**  $k=2, k=3, k=5$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado..... 85
49. Resultados obtenidos del algoritmo DAPF en escenarios dispersos, en donde la **Reachability**  $k=2, k=3, k=5$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado..... 85
50. Resultado de la métrica **Saved Rebroadcast** para el esquema DAPF para  $K=2, K=3, K=5$  respectivamente para escenarios densos..... 86
51. Resultado de la métrica **Saved Rebroadcast** para el esquema DAPF para  $K=2, K=3, K=5$  respectivamente para escenarios dispersos..... 86
52. Resultados obtenidos del algoritmo CBS en escenarios densos, en donde el **Delay**  $Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado..... 87

53. Resultados obtenidos del algoritmo CBS en escenarios dispersos, en donde el **Delay** Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado. .... 87
54. Resultados obtenidos del algoritmo CBS en escenarios densos, en donde la **Reachability** Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado. .... 88
55. Resultados obtenidos del algoritmo CBS en escenarios dispersos, en donde la **Reachability** Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado... 88
56. Resultado de la métrica **Saved Rebroadcast** para el esquema CBS para Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente para escenarios densos. .... 89
57. Resultado de la métrica **Saved Rebroadcast** para el esquema CBS para Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente para escenarios densos. .... 89

## Lista de tablas

Tabla	Página
1. Parámetros de CSMA/CA no ranurado. ....	47
2. Parámetros de simulación para la validación del simulador. ....	49
3. Análisis del empaquetamiento para validación del simulador COOJA. ....	50
4. Parámetros de la Capa física que se utilizarán en la simulación.....	51
5. Parámetros de la capa MAC que se utilizarán en la simulación. ....	52
6. Parámetros de la capa RDC que se utilizarán en la simulación.....	52
7. Estructura de una línea de la herramienta <i>Mote Output</i> . ....	54
8. Valores utilizados para los parámetros a variar dentro del escenario de simulación. ....	56
9. Descripción de los tiempos de acuerdo a la frecuencia del ciclo de trabajo. ....	58



## Capítulo 1. Introducción

---

Desde que surgió la interconexión entre dispositivos e internet a partir de la segunda mitad del siglo XX(Zakon, 1997), IPv4 es el protocolo de direccionamiento en red más importante y ampliamente desplegado que existe(Czyz et al., 2014). Sin embargo, IPv4 posee una capacidad de conexión limitada y para mediados de la segunda década del presente siglo, la organización de Registros Regionales de Internet (RIR) en varios periodos declaró las direcciones IPv4 como agotadas. IPv6 emerge como una solución a las limitaciones que presentaba IPv4, abordando tanto las capacidades de conexión como la capacidad para resolver problemas de conexión y descubrimiento de vecinos. Mientras que IPv4 soporta  $2^{32}$  direcciones, IPv6 soporta  $2^{128}$  direcciones, lo que equivale a millones de direcciones. A esta capacidad de direccionamiento que se ha logrado, le sumamos el abaratamiento de los costos de producción de los microcontroladores, dispositivos embebidos, sensores y sistemas microelectromecánicos (MEMS), además de la miniaturización de todos estos antes mencionado. Esto ha traído consigo un incremento en el uso y aplicaciones de las redes inalámbricas de sensores (WSN).

Las WSN están conformadas por dispositivos con poder limitado de procesamiento y de almacenamiento de energía. Estos dispositivos tienen la capacidad de interconexión de forma centralizada o distribuida, es decir, sin la necesidad de tener una entidad centralizada, como con un enrutador.

La vida útil de los nodos dentro de la red se ve limitada por el tiempo de vida de su batería, ya sea debido a la dificultad o imposibilidad de cambiarlas. En este sentido, la extensión de la vida útil representa uno de los desafíos principales en las WSN. Los módulos a los cuales se les atribuye un consumo de energía en el nodo sensor son: monitorización, comunicación y procesamiento de datos.

El módulo que más consumo de energía presenta es el de radio de comunicación tanto para transmisión como recepción(Jurdak et al., 2007; Shnayder et al., 2004). Especialmente el consumo que tiene este módulo durante el tiempo de *idle listening*. Aunque no es el evento que más consume por sí solo, su prolongada duración implica que la suma de estos consumos constituye un gasto excesivo de energía. Este periodo se refiere al tiempo en el que la radio permanece activa, a la espera de recibir alguna transmisión o paquete, incluso en situaciones en las que dicho paquete puede nunca llegar(Ye et al., 2004). El análisis de un nodo sensor específico, el Micaz de Crossbow, revela un consumo de corriente para la transmisión que varía entre 3.7 mA y 21.5 mA, dependiendo de la potencia de transmisión. Sin embargo, el consumo

de corriente en el modo *idle listening* es de 7 mA, independientemente de si se está transmitiendo o no. Este estado de escucha consume la mayor parte de la energía de la batería(Zhou & Guan, 2009).

Como respuesta a la problemática del consumo excesivo de energía atribuido al módulo de radio, se adopta la implementación de ciclos de trabajo en los nodos de las WSN(Rout & Ghosh, 2012). Este enfoque implica alternar entre periodos de activación y desactivación del radio, con el objetivo de optimizar el ahorro de energía. La optimización de los periodos de actividad y reposo de los nodos sensores no solo extiende la vida útil de las baterías, sino que también contribuye a la sostenibilidad y eficiencia energética. Sin embargo, este enfoque plantea un desafío para la disseminación de información. La necesidad de sincronizar los ciclos de activación puede generar retrasos en la transmisión de datos, lo que podría impactar la velocidad y la confiabilidad en la entrega de información crítica. Esto implica un equilibrio delicado entre el ahorro de energía y la eficacia en la comunicación, destacando así la importancia de estrategias de disseminación eficientes para abordar el desafío de optimizar el rendimiento general de las WSN.

Un reto particularmente importante es el de la comunicación *broadcast*, i.e., envío de paquetes de datos dirigidos a todos los dispositivos o nodos de una red, por ejemplo, para anuncios, actualizaciones o solicitudes de descubrimiento en los procesos de conformación de la red(Benrhaïem & Hafid, 2020; Ibrahim et al., 2020). El uso indebido o indiscriminado de este tipo de paquetes de *broadcast* puede traer consigo el conocido fenómeno de *broadcast storm* o tormenta de difusión(Ni et al., 1999), el cual incrementa de forma significativa el retardo, la contención del medio y la pérdida de paquetes, entre otros efectos negativos. Además, en las WSN con ciclo de trabajo, la falta de sincronización de los ciclos de trabajo de los nodos podría disminuir el número de nodos receptores del paquete.

En la literatura científica, se ha explorado la mitigación de este fenómeno mediante la aplicación de técnicas llamadas algoritmos de disseminación *broadcast* o algoritmos de *broadcast*. Estos algoritmos están concebidos con el objetivo principal de disminuir el nivel de redundancia, cuidando de no perder la eficiencia en la disseminación de los paquetes en la red(Shu et al., 2019) (Jung et al., 2018). Estos algoritmos de disseminación contribuyen a mejorar parámetros como la contención, la pérdida de paquetes por colisiones, entre otros aspectos. Sin embargo, en el contexto de las WSNs con ciclo de trabajo, disminuir la redundancia podría disminuir la cantidad de nodos que reciben un mensaje. Este fenómeno se atribuye, como se explicó anteriormente, a la carencia de sincronización en los ciclos de trabajo. Este trabajo de tesis se enfoca en evaluar el impacto del ciclo de trabajo en la eficiencia de los algoritmos de disseminación *broadcast* para WSNs.

## 1.1 Planteamiento del problema

El despliegue de redes WSNs para habilitar diversas aplicaciones críticas, como la monitorización ambiental y la salud, ha resaltado la importancia de la eficiencia de la comunicación en estos entornos. Principalmente de la comunicación **broadcast**, que se utiliza tanto para información de control (p. ej., en la formación de la red), como para mensajes de aplicación (p.ej., advertencias de accidente en una carretera).

Uno de los factores determinantes en la gestión de energía de las WSN es el ciclo de trabajo, que regula los períodos activos e inactivos de los nodos sensores y sus transceptores de radio para conservar la energía de las baterías. Sin embargo, también puede afectar el desempeño de los algoritmos de disseminación. Esto, debido a las pérdidas y retardos adicionales introducidos. De la revisión de la literatura realizada, no se encontró un trabajo que abordar este relevante tema.

El estudio del impacto de la variación del ciclo de trabajo en los algoritmos de disseminación **broadcast** es de especial interés, ya que la eficacia de la difusión **broadcast** es crucial para la transmisión oportuna y confiable de información en aplicaciones críticas. La presente investigación tiene como objetivo abordar este problema, evaluando de manera sistemática y cuantitativa el impacto del ciclo de trabajo en la eficiencia de los algoritmos de disseminación **broadcast** en WSN. Particularmente, se evaluará con las métricas de retransmisiones evitadas (**SRB**), la **Reachability** y el **Delay**.

Medir el impacto que tiene el ciclo de trabajo de las WSN en el desempeño de los algoritmos de disseminación **broadcast** no es una tarea trivial. Es necesario considerar aspectos relacionados a los escenarios de red, p.ej., la densidad de nodos, y también el funcionamiento de los algoritmos de disseminación. Los resultados de este trabajo servirán de guía para el diseño de WSN con ciclos de trabajo que usen algoritmos de **broadcast**, teniendo en cuenta las métricas de desempeño utilizadas, así como futuros diseños de algoritmos de **broadcast** en WSN teniendo en cuenta las afectaciones que el ciclo de trabajo pueda tener en su desempeño. Lo que contribuye al desarrollo de estrategias de disseminación más eficientes para WSN, a través de información relevante para la comunidad de investigación y desarrolladores de sistemas de WSNs

Existen varias tesis y trabajos los cuales exhiben técnicas para mitigar estos fenómenos en redes sin ciclo de trabajo, pero, no se encontró un trabajo previo el cual analice el comportamiento de estos algoritmos

en redes que cuenten con estados *on/off* de los nodos de esta red, las cuales presentan mayores retos para establecer una comunicación satisfactoria con sus nodos vecinos.

## 1.2 Justificación

En la actualidad, con el crecimiento del internet, la aparición de IPv6 y el desarrollo que han alcanzado las redes de sensores en el Internet de las Cosas (Internet of Things, IoT por sus siglas en inglés) ha expandido el uso de las WSNs. La topología distribuida es de especial interés en escenarios como agricultura de precisión y de Internet de las Cosas. En WSNs con topología distribuida. Si no se tiene control sobre la misma, puede generar saturación en la red, ineficiencia de ésta y que aparezca el fenómeno de *broadcast storm*.

La investigación de este trabajo de tesis se centra en la evaluación del impacto del ciclo de trabajo en WSNs sobre la eficiencia de algoritmos de disseminación *broadcast*. Las WSNs desempeñan un papel crucial en diversas aplicaciones, desde monitoreo ambiental hasta sistemas de salud, donde la comunicación eficiente es esencial. El ciclo de trabajo, que incluye periodos de actividad y de inactividad para conservar la energía, puede afectar significativamente la capacidad de disseminación de paquetes *broadcast* en estas redes. Estudiar cómo diferentes configuraciones de ciclo de trabajo influyen en la eficiencia de los algoritmos de disseminación *broadcast* permitirá obtener información relevante que sea la base para que, en posteriores trabajos, se optimice el rendimiento de las WSNs con ciclos de trabajo. Por lo que los resultados de este trabajo de tesis contribuyen al desarrollo de algoritmos de *broadcast* para WSN que utilicen ciclos de trabajo más eficientes, teniendo como base las métricas empleadas en esta investigación y los resultados que arrojen las mismas.

## 1.3 Objetivos

### 1.3.1 Objetivo general

Evaluar el impacto del uso de ciclos de trabajo en los algoritmos de disseminación de *broadcast* sobre redes inalámbricas de sensores.

### 1.3.2 Objetivos específicos

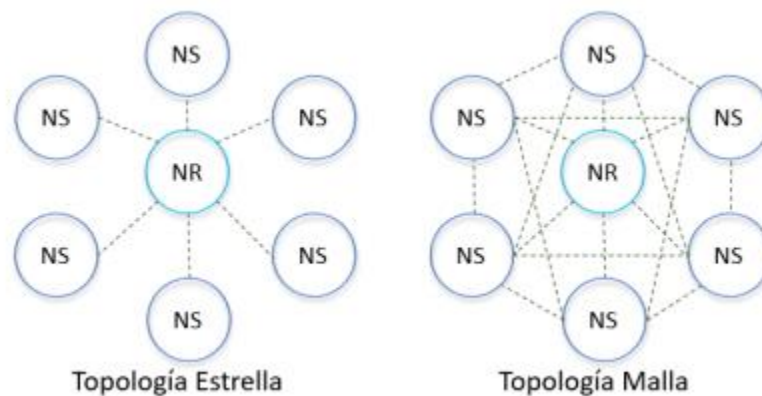
- Seleccionar e implementar los algoritmos de disseminación **broadcast** más relevantes en la literatura para su evaluación.
- Diseñar, implementar y validar una cama de pruebas para la evaluación de algoritmos de disseminación **broadcast** en WSN con ciclo de trabajo.
- Analizar las ventajas y desventajas de cada enfoque a través de la experimentación en la cama de pruebas.

## 1.4 Antecedentes

Las WSN están formadas por numerosos nodos individuales que operan en conjunto para recopilar información del entorno en el que están desplegados. Estos nodos suelen estar equipados con sensores que les permiten medir una variedad de parámetros físicos, como temperatura, presión, humedad, luz, movimiento, entre otros, según la aplicación para la que fueron desarrollados (Dludla et al., 2013). Los datos recopilados por estos sensores se transmiten de un nodo a otro a través de conexiones inalámbricas, lo que permite la recolección de información en tiempo real desde diversos puntos de una ubicación remota. Las WSNs se utilizan en una amplia gama de aplicaciones, como monitoreo ambiental, control industrial, atención médica, agricultura de precisión y más (Baggio, 2005; Catenazzo et al., 2018; Majid et al., 2022; Ojha et al., 2015). La capacidad de desplegar numerosos nodos pequeños y económicos en áreas dispersas o de difícil acceso hace que estas redes sean muy versátiles y útiles para la recopilación de datos y la toma de decisiones basada en datos en diversas industrias y entornos.

Las WSN pueden implementarse en diversas topologías, y dos de las más comunes son la topología de estrella y la topología de malla como se muestra en la Figura 1. En la topología de estrella, se simplifica el proceso de configuración de la red y el enrutamiento, ya que existe una entidad central (el Nodo Resumidero, NR) que se encarga de gestionar la incorporación de nuevos nodos a la red. Sin embargo, esta topología presenta un desafío en términos de escalabilidad. A medida que el número de nodos en la red aumenta, se requieren más recursos por parte del nodo resumidero para administrar eficazmente la red. Además, la topología de estrella limita el alcance de la red a la zona de cobertura del nodo resumidero, lo que puede ser un inconveniente en redes más grandes o en áreas extensas (Cardenas, 2019). Por otro lado,

la topología de malla aborda algunos de estos desafíos de escalabilidad y alcance. En una topología de malla, los nodos pueden comunicarse directamente entre sí en lugar de depender de un nodo resumidero central. Esto distribuye la carga de gestión de la red entre los propios nodos y permite que la red se extienda a áreas más grandes. Sin embargo, la configuración y el enrutamiento en una topología de malla suelen ser más complejos que en una topología de estrella. La elección entre estas topologías dependerá de las necesidades específicas de la aplicación y de los recursos disponibles. Cada una tiene sus ventajas y desventajas, y la decisión se basará en consideraciones como la escalabilidad, el alcance, la complejidad y la eficiencia en el consumo de energía(Sharma et al., 2013).



**Figura 1.** Representación de topología estrella y malla(CARDENAS, 2019).

La condición de "todos los nodos activos" no es práctica para las WSN con limitación de energía. Los nodos sensores ahorran energía alternando entre estados activos y dormidos (es decir, en modo de suspensión). La proporción entre el tiempo durante el cual un nodo sensor está en estado activo y el tiempo total de estados activos/dormidos se denomina ciclo de trabajo. El ciclo de trabajo depende de la densidad de nodos en el área monitoreada para lograr una mejor cobertura y conectividad. Por lo general, en una WSN densa, el ciclo de trabajo de un nodo es muy bajo. Una WSN con ciclo de trabajo puede clasificarse de manera general en dos tipos principales: WSN con ciclo de trabajo aleatorio(Hsin & Liu, 2006) y WSN con ciclo de trabajo coordinado(Wang et al., 2010). En el primero, los nodos sensores se encienden y apagan de manera independiente de forma aleatoria. En el segundo, los nodos sensores se coordinan entre sí mediante intercambios de mensajes de comunicación y control. Esta coordinación puede ser eficiente para la comunicación, pero requiere un intercambio adicional de información para difundir el horario activo/dormido de cada nodo. Las WSN con ciclo de trabajo aleatorio son más simples de diseñar, ya que no requieren sobrecarga adicional(Rout & Ghosh, 2012).

Los paquetes **broadcast**, en el contexto de las redes de telecomunicaciones, son una forma de comunicación que se utiliza para enviar paquete de datos desde una fuente a todos los dispositivos o nodos en la red, sin importar su ubicación. Su función principal es lograr que múltiples dispositivos reciban la misma información al mismo tiempo, lo que resulta especialmente útil en situaciones que requieren sincronización, como actualizaciones de software, resolución de problemas de red, anuncios críticos o alarmas. Estos paquetes se envían a todas las direcciones en la red, asegurando que cada dispositivo los reciba. A pesar de su simplicidad y efectividad en la distribución de datos, los paquetes de difusión pueden generar tráfico innecesario en la red, lo que consume ancho de banda y energía, dando aparición al fenómeno del **broadcast storm** el cual acentúa los problemas de demora, pérdida de paquetes, contención del medio entre otros. Por esta razón, se utilizan con moderación y se combinan con otros algoritmos de **broadcast** para hacer más eficiente este tipo de transmisiones de **broadcast**.

De manera general, los algoritmos de **broadcast** son técnicas que se utilizan para mitigar el fenómeno del **broadcast storm**. Dándole así a los nodos la capacidad de retransmitir o no el paquete de **broadcast** basándose en parámetros previamente establecidos por el algoritmo. Uno de los algoritmos de **broadcast** más simples es el **flooding**, en el cual, cada nodo en la red retransmite un mensaje entrante una vez. La principal ventaja de utilizar el **flooding** radica en su facilidad de implementación (Haas et al., 2006; Reina et al., 2015). Los algoritmos de **broadcast** pueden clasificarse como de probabilidad fija, adaptativos y basados en contador.

Los algoritmos de **broadcast** de probabilidad fija usan una probabilidad de reenvío constante, por lo que cada nodo tiene la misma probabilidad de reenvío en la red. Los esquemas probabilísticos fijos se han estudiado utilizando la teoría de percolación y el fenómeno de transición de fase utilizado en redes aleatorias (Sasson et al., 2003; Shen et al., 2006). Sin embargo, existen diferencias importantes entre las redes aleatorias y las redes ad-hoc, por lo que los resultados observados en las redes aleatorias no pueden considerarse ciertos para las redes ad-hoc. En consecuencia, no existe una probabilidad de reenvío óptima para todos los escenarios posibles. Esta probabilidad de reenvío óptima puede depender de muchos parámetros, como la densidad, la distancia entre nodos, la velocidad y la aplicación (Haas et al., 2002, 2006; Reina et al., 2016).

Los algoritmos de **broadcast** adaptativos utilizan parámetros locales o globales, como métricas de densidad, distancia y energía, para determinar la probabilidad de reenvío. En (Cartigny & Simplot, 2003) el autor propone un algoritmo de **broadcast** con una probabilidad de reenvío adaptativa basándose en la densidad del vecindario que rodea al nodo que desea retransmitir. En (Camp et al., 2002; Williams & Camp,

2002) proponen un algoritmo de **broadcast** con probabilidad adaptativa donde la idea principal es utilizar la distancia relativa entre dos nodos para ajustar la probabilidad de reenvío, de modo que los nodos no necesitan información global de otros nodos en la red.

Los algoritmos de **broadcast** basados en contadores a diferencia de los probabilísticos se rigen por umbrales de conteo y demoras de acceso aleatorio (RAD por sus siglas en inglés). Estos esperan un RAD y durante ese intervalo de tiempo cuentan los paquetes que reciben iguales, si terminado este tiempo la cantidad de paquetes recibidos no supera el umbral, este se retransmitirá, de lo contrario será descartado el paquete. En (al-Humoud et al., 2008) el autor propone el cálculo del RAD basándose en la densidad del vecindario que tenga un nodo, donde un nodo que tenga mayor cantidad de vecinos (caso denso) tendrá un RAD más pequeño y por lo tanto mayor chance de retransmisión. El autor (Yassein et al., 2011) propone un algoritmo de **broadcast** donde su parámetro de referencia es el remanente de la batería de los nodos. En este enfoque, se calcula el RAD de cada nodo en función de la carga restante en su batería. Como resultado, los nodos con una mayor carga de batería tienen mayores posibilidades de realizar retransmisiones, lo que contribuye a prolongar la vida útil de la red. En (Jung et al., 2018) propone un algoritmo de **broadcast** donde se rige por varios parámetros para el cálculo del RAD, tales como batería y área de convergencia, asíndolo más completos que otros existentes en la literatura.

## 1.5 Metodología

Para el desarrollo del presente trabajo de tesis se siguió una metodología en seis fases, las cuales se describen a continuación.

1. Investigar el estado del arte de los algoritmos de *broadcast*.
  - Se realizó una investigación sobre los algoritmos de broadcast existentes en la literatura y la aplicación de estos para mitigar el fenómeno de *broadcast storm*.
2. Seleccionar los algoritmos para realizar su evaluación.
  - Se seleccionó un grupo de algoritmos de los existentes en la literatura para realizar la evaluación de los mismos y ver el impacto de los ciclos de trabajo en los mismos.



3. Definir los escenarios y métricas de evaluación de los algoritmos sobre redes con ciclos de trabajo.
  - Se seleccionaron varias estructuras de escenarios para evaluar el comportamiento de los algoritmos anteriormente escogidos, escenarios con una estructura perfectamente organizado en cuadrículas como si se trataran del trazado de las calles de Manhattan y otras estructuras representadas de manera que presente más densidad de nodos por unidad de área.
4. Generar un modelo de simulación validado para redes WSN con ciclo de trabajo.
  - Es de vital importancia emplear herramientas que garanticen la fiabilidad de los resultados obtenidos. En este sentido, se llevó a cabo la validación del simulador de ContikiOs-Cooja a través del análisis de la capa MAC del estándar IEEE 802.15.4 utilizado en las WSNs. El objetivo fue establecer un fundamento sólido y un método de comparación mediante un esquema de simulación validado. Al utilizar este enfoque, se logra proyectar los resultados hacia escenarios más complejos, lo que permite obtener datos confiables y verídicos.
5. Evaluar el desempeño de los algoritmos y el impacto del ciclo de trabajo en los mismos a través de la cama de pruebas basada en el simulador de WSN Cooja.
  - Se decidió utilizar el modelador de redes Cooja del sistema operativo de Contiki, debido a las características que presenta, para determinar la interacción de los nodos para formar la red y añadir las técnicas de ciclo de trabajo. Debido a que una única realización de un experimento no se puede considerar como un resultado definitivo, es necesario por el método Monte Carlo realizar un gran número de experimentos con distintas semillas y luego de los resultados obtenidos realizar un promedio para obtener un resultado definitivo.
6. Elaborar conclusiones y documentar resultados obtenidos.
  - Una vez evaluado el desempeño de los diferentes algoritmos de *broadcast* y evaluar el impacto de los ciclos de trabajos sobre los mismos, se generaron conclusiones de los resultados mostrados, basándonos en métricas de desempeño, tales como las retransmisiones guardadas (*SRB*), la capacidad de alcance a toda la red (*Reachability*) y la demora de las transmisiones (*Delay*).

## Capítulo 2. Marco teórico

---

En esta sección se describen los conceptos teóricos base para este trabajo de tesis, tales como: las capas física y MAC, el sistema operativo utilizado para las simulaciones, incluyendo el funcionamiento de la capa RDC (Radio Duty Cycle por sus siglas en inglés) para el manejo del ciclo de trabajo en WSNs. Se presentará a detalle los algoritmos de **broadcast** existentes en la literatura y el funcionamiento de estos.

### 2.1 Tecnología de WSN

Una WSNs es un conjunto de nodos sensores dispersos espacialmente que están interconectados mediante comunicación inalámbrica. Como se muestra en la Figura 2, un nodo sensor, también llamado **mote**, es un dispositivo electrónico que consta de un procesador junto con una unidad de almacenamiento, un módulo transceptor, uno o varios sensores, junto con un convertidor analógico a digital (ADC), y una fuente de alimentación, que normalmente es una batería. Opcionalmente, puede incluir una unidad de posicionamiento y/o una unidad de movilización (Kandris et al., 2020). Los sensores se clasifican en función de su idoneidad para despliegue en campo, teniendo en cuenta factores como el costo, la escalabilidad y la eficiencia en la ingeniería. Las principales categorías de sensores son sensores físicos, químicos y biológicos (Ramson & Moni, 2017).

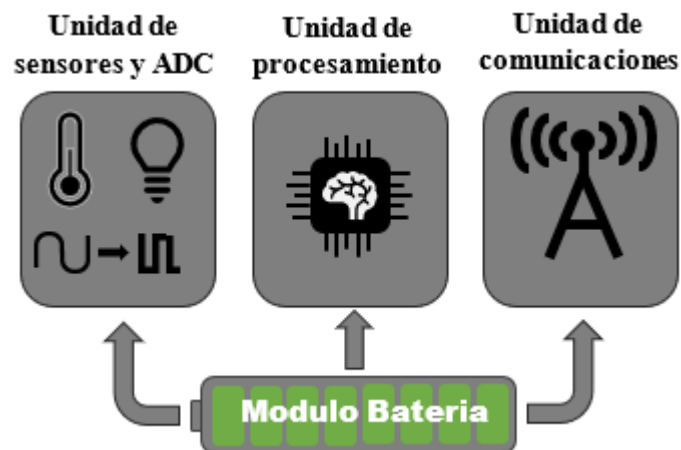


Figura 2. Arquitectura típica de un nodo de una WSN.

Las WSN pueden clasificarse en dos estructuras principales: centralizada y descentralizada. En una WSN

centralizada, todos los nodos sensores envían sus datos a un nodo central o estación base. Las ventajas de esta estructura incluyen un control centralizado que simplifica la administración de los protocolos de red y el procesamiento de datos, así como una mayor eficiencia en la transmisión de datos ya que se evitan colisiones. Sin embargo, las desventajas pueden incluir un mayor consumo de energía en los nodos sensores debido a la necesidad de transmitir datos largos a distancias más largas y una mayor vulnerabilidad a fallos en el nodo central.

Por otro lado, en una WSN descentralizada o redes ad-hoc, los nodos sensores pueden comunicarse directamente entre sí o a través de otros nodos en la red sin depender de una estación base central, en otras palabras, los dispositivos actúan tanto como nodos finales como enrutadores para retransmitir datos a otros nodos en la red. Esto permite una mayor flexibilidad y escalabilidad, ya que los nodos pueden adaptarse a cambios en la topología de la red. Sin embargo, las desventajas incluyen una mayor complejidad en la gestión de la red y la posibilidad de congestión en la comunicación entre nodos, lo que puede llevar a un consumo de energía no uniforme y problemas de latencia en la transmisión de datos (Al-Obaisat & Braun, 2007; Kandris et al., 2020; Ramson & Moni, 2017).

La eficiencia de una red de comunicación ad-hoc depende no solo de sus protocolos de control, sino también de su topología. Una red inalámbrica ad-hoc, o simplemente una red ad-hoc, consiste en una colección de nodos distribuidos geográficamente que se comunican entre sí a través de un medio inalámbrico. Estas redes por lo general tienen una estructura descentralizada y las capacidades de comunicación de la red están limitadas por la energía de la batería de los nodos de la red. En las redes inalámbricas, como las redes ad-hoc, la capacidad de movilidad y conectividad es fundamental y una de las principales ventajas que presentan sobre redes cableadas. No podemos aplicar un modelo de topología ideal o estándar, ya que estas redes se construyen y desarrollan según las necesidades de la aplicación, las características geográficas y los requisitos específicos. Sin embargo, es posible dividir una red en secciones o fragmentos de topología que pudieran ser analizadas y catalogadas.

### 2.1.1 Capa Física (PHY)

La capa física (PHY, abreviación en inglés) es el nivel más bajo del modelo Interconexión de Sistemas Abiertos (OSI por sus siglas en inglés) o la primera capa. La capa física incluye los medios físicos como un conector de acoplamiento de cable y las técnicas requeridas para la transmisión en estos medios físicos, como la modulación. La función principal de la capa física es completar el flujo de bits original entre nodos

adyacentes(Kumar et al., 2014); si la transmisión de datos es en ambas direcciones o solo en una. Cómo establecer la integridad de la comunicación desde el origen hasta el destino. Define un protocolo para proporcionar una conexión (no necesariamente confiable) entre dos nodos directamente conectados y la modulación o conversión entre la representación de datos digitales en equipos de usuario y las señales correspondientes transmitidas sobre el canal de comunicación física(Li et al., 2011). Este canal puede involucrar un medio físico (como cobre y fibra óptica) o un medio inalámbrico. Aunque la capa física no proporciona servicios de corrección de errores, puede establecer la velocidad de transferencia de datos y monitorear la tasa de error de los datos. La capa física de Parallel SCSI opera en esta capa, al igual que las capas físicas de Ethernet y otras redes de área local, como Token Ring, FDDI, ITU-T G.hn y IEEE 802.11, así como redes de área personal como Bluetooth e IEEE 802.15.4(Angueira et al., 2022).

En este trabajo de tesis se utiliza la capa física del estándar IEEE 802.15.4. La cual se usa comúnmente para redes con baja potencia, baja velocidad de transmisión de datos y aplicaciones de menor costo. Este estándar técnico sirve como base para otros estándares como Zigbee, que lo extienden mediante el desarrollo de capas superiores que no están definidas en el estándar. Utilizando mecanismos de encapsulación y compresión de encabezados definidos por la tecnología 6LoWPAN, también se puede permitir el envío y recepción de paquetes IPv6 a través de IEEE 802.15.4, que es el acrónimo de Redes de Área Personal Inalámbricas de Baja Potencia (LoWPAN) para IPv6. El estándar IEEE 802.15.4 se centra principalmente en proporcionar comunicación ad-hoc de baja velocidad y bajo costo, ofreciendo una red inalámbrica de área personal(Baronti et al., 2007; Khanafer et al., 2013). En comparación con otros enfoques como Wi-Fi, que requieren más energía y ofrecen mayor ancho de banda, el énfasis en el estándar IEEE 802.15.4 está en la comunicación entre dispositivos cercanos sin infraestructura subyacente, con el menor costo y consumo de energía posible.

La capa física definida por este estándar IEEE 802.15.4 es responsable de gestionar el transceptor físico de radiofrecuencia. También realiza funciones de selección y gestión de canales para la energía y las señales. La capa física admite tres bandas de frecuencia: una banda de 2450 MHz (con 16 canales), una banda de 915 MHz (con 10 canales) y una banda de 868 MHz (1 canal), todas utilizando el modo de acceso de espectro ensanchado de secuencia directa (DSSS). La banda de 2450 MHz emplea la modulación de desplazamiento de fase en cuadratura con desfase (O-QPSK) mientras que las bandas de 868/915 MHz dependen de la modulación de desplazamiento de fase binaria (BPSK)(Baronti et al., 2007).

### 2.1.2 Capa MAC

La capa de control de acceso al medio, o capa MAC por sus siglas en inglés, es una parte esencial del modelo OSI en las redes de telecomunicaciones. Se ubica en el nivel de enlace de datos y desempeña un papel fundamental en la administración del acceso a un medio de transmisión compartido, como un canal de comunicación o una red compartida. La función principal de la capa MAC es regular cómo los dispositivos comparten y transmiten datos de manera eficiente en dicho medio, evitando colisiones. Una característica principal de los diferentes tipos de redes inalámbricas es la forma en que controlan el acceso al medio de transmisión. Se conocen varios mecanismos de acceso, como el Acceso Múltiple por División de Tiempo (TDMA por sus siglas en inglés), el Acceso Múltiple por División de Frecuencia (FDMA por sus siglas en inglés) y el Acceso Múltiple por División de Código (CDMA por sus siglas en inglés).

En el caso de un medio compartido en el que cada nodo puede transmitir a todos los nodos dentro de su rango de transmisión, el FDMA es difícil de implementar y el CDMA requiere un esquema para distribuir los códigos. El TDMA no es posible sin una sincronización externa, como el GPS(Lauwens et al., 2010). Los fallos en la transmisión pueden ser mitigados mediante un esquema de reconocimiento positivo (ACK por sus siglas en inglés), en el que un paquete se retransmite si el remitente no recibe una confirmación del receptor. Este enfoque, aunque efectivo, reduce la tasa de datos general debido a las retransmisiones de paquetes y la transmisión de paquetes ACK. Para tomar decisiones más informadas, un nodo puede detectar el estado del medio compartido antes de iniciar una transmisión. El Acceso Múltiple con Detección de Portadora (CSMA) ayuda a prevenir colisiones al abstenerse de transmitir si se percibe que el medio está ocupado. Sin embargo, las colisiones debidas a nodos ocultos aún pueden ocurrir.

En las redes inalámbricas, la detección de colisiones, como la utilizada en el Acceso Múltiple con Detección de Colisión (***Carrier Sensing Multiple Access with Collision Detection***, CSMA/CD por sus siglas en inglés) para redes cableadas(Beck, 2005), no es factible. Para reducir aún más las colisiones en canales inalámbricos, se puede emplear un esquema de espera aleatoria. Este esquema limita la probabilidad de que varios nodos inicien transmisiones simultáneamente y se conoce como Acceso Múltiple con Evitación de Colisiones (***Carrier Sensing Multiple Access with Collision Avoidance***, CSMA/CA por sus siglas en inglés). (Nasipuri et al., 1999; Tay et al., 2004).

La capa MAC también desempeña otras funciones clave, como la asignación de direcciones MAC únicas a cada dispositivo para su identificación dentro de la red local, la priorización de datos según su importancia, la gestión de tramas de datos dividiéndolos en fragmentos más pequeños, la detección y corrección de

errores en las tramas, la sincronización de dispositivos y la resolución de conflictos cuando varios dispositivos intentan transmitir simultáneamente. Estas funciones aseguran un flujo eficiente y ordenado de datos entre los dispositivos en la red. Los estándares de redes, como Ethernet y Wi-Fi, implementan protocolos específicos en la capa MAC para facilitar la comunicación en estas tecnologías particulares. La importancia de la capa MAC en las redes de telecomunicaciones radica en su capacidad para coordinar y regular el acceso al medio compartido, garantizando una transmisión de datos efectiva y sin conflictos entre dispositivos(Cagalj et al., 2005).

La capa de control de acceso al medio (MAC) definida en el modelo OSI se encarga de permitir la transmisión de tramas MAC a través del canal físico. También administra el envío de *beacons* de red y el acceso a los canales físicos. Además, puede encargarse de las asociaciones de nodos, garantizar intervalos de tiempo, controlar la validación de las tramas y proporcionar servicios de seguridad. Los dos modos operativos admitidos son los *beacon* habilitado y los *beacon* no habilitados(Zen et al., 2008)

#### 2.1.2.1 CSMA/CA No ranurado del protocolo 802.15.4

El protocolo CSMA/CA se basa en la postergación aleatoria de las transmisiones de paquetes para el uso eficiente de un canal inalámbrico compartido entre muchos nodos en una red(Cagalj et al., 2005).

En el algoritmo CSMA/CA no ranurado, los nodos antes de transmitir un paquete, primero escuchan el medio para detectar si hay actividad de transmisión. Si el medio está ocupado, esperan un tiempo aleatorio (*backoff*) antes de intentar nuevamente. Este mecanismo aleatorio de *backoff* ayuda a evitar colisiones en el canal y permite que los nodos compartan el medio de manera eficiente. Si varios nodos intentan transmitir al mismo tiempo y hay una colisión, el mecanismo de CSMA/CA realizará retransmisiones para intentar transmitir con éxito el paquete. Antes de iniciar la transmisión, cada nodo elige un tiempo aleatorio dentro de una ventana de contención, definida por el valor BE (*backoff exponent*). Esta ventana de tiempo aleatorio asegura que los nodos no intenten transmitir al mismo tiempo, reduciendo así las colisiones en el canal. Luego, el nodo realiza técnicas de censado del medio para determinar el estado actual del canal dentro de su rango de cobertura. Si el canal está ocupado debido a que otro nodo está transmitiendo, el nodo esperará durante un nuevo periodo de tiempo aleatorio antes de volver a censar el canal e intentar otra transmisión. Este proceso se repetirá hasta que el número de intentos de transmisión NB haya excedido el valor máximo de intentos  $macMaxCSMABackoffs$ .



Figura 3. Mecanismo de CSMA/CA no ranurado(Cardenas, 2019)

La Figura 3 muestra un diagrama de flujo que describe la versión sin ranuras del protocolo de acceso CSMA/CA. Se puede resumir en cinco pasos:

1. Inicialización de NB y BE: NB se inicia en 0 y BE se establece en macMinBE, que por defecto es igual a 3.
2. Retardo aleatorio para evitar colisiones: el algoritmo intenta evitar colisiones esperando durante un retardo generado aleatoriamente en el rango  $[0, 2^{BE} - 1]$  en períodos de unidades de espera. Un período de unidad de espera equivale a  $20 T_s$ , con  $T_s = 16 \mu s$ .
3. Evaluación del canal despejado (**Clear channel assessment** CCA por sus siglas en inglés): la evaluación del canal comienza inmediatamente después de que expire el retardo aleatorio. Si se detecta que el canal está ocupado, el algoritmo pasa al paso 4. De lo contrario, es decir, si el canal está libre, el algoritmo pasa al paso 5.

4. Canal ocupado: los valores de NB y BE aumentan en uno. Sin embargo, BE no puede exceder  $macMaxBE$ , que tiene un valor predeterminado de 5. Si el número de intentos es menor que  $macMaxCSMABackoffs$ , con un valor predeterminado de 4, el algoritmo vuelve al paso 2. De lo contrario, el algoritmo finaliza con un estado de falla en el acceso al canal.
5. Canal inactivo: la subcapa MAC comienza inmediatamente a transmitir su trama actual justo después de que el canal se evalúa como inactivo mediante el procedimiento de CCA.

### 2.1.3 Ciclo de trabajo

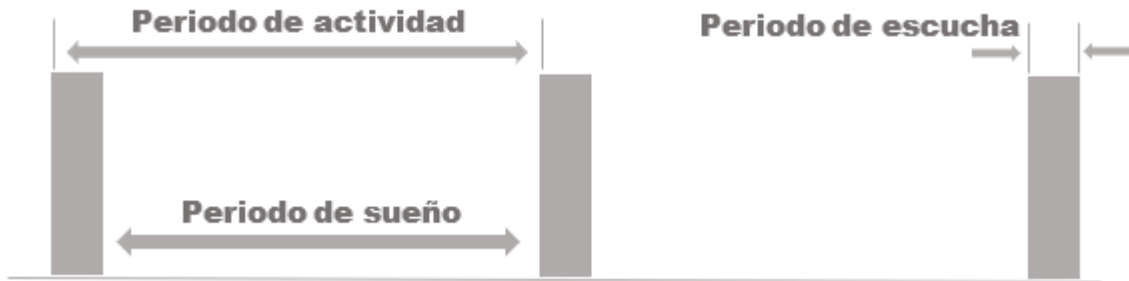
Los dispositivos inalámbricos de baja potencia deben mantener estrictos presupuestos de energía para lograr años de vida útil. De todos los componentes en un dispositivo inalámbrico de baja potencia, el transceptor inalámbrico a menudo tiene el mayor consumo de energía. El transceptor inalámbrico consume la misma cantidad de energía cuando está en modo de escucha pasiva (*listening*) para recibir transmisiones de otros dispositivos que cuando está transmitiendo activamente, por lo que el transceptor debe apagarse por completo para ahorrar energía. Dado que el dispositivo no puede recibir ningún dato cuando el transceptor está apagado, se debe utilizar un mecanismo de ciclo de trabajo para encender el transceptor de vez en cuando (Dunkels, 2011) (Ashraf et al., 2017). El ciclo de trabajo en las WSNs se utiliza comúnmente como una estrategia para reducir el consumo de energía en los nodos y prolongar su vida útil en la red. Sin embargo, esta estrategia puede tener implicaciones en otros aspectos del rendimiento de la red, como el caudal efectivo y el retraso en la transmisión (Carrano et al., 2013; Charfi & Bouyahi, 2012).

Las WSNs con ciclos de trabajo se pueden clasificar en dos categorías principales: las Redes Inalámbricas de Sensores con Ciclos Aleatorios (Hsin & Liu, 2006) y las Redes Inalámbricas de Sensores con Ciclos Coordinados (Wang et al., 2010). En el primer caso, los nodos sensores se encienden y apagan de manera independiente en un patrón aleatorio. En el segundo caso, los nodos sensores se coordinan entre sí a través de comunicación e intercambio de mensajes de control. Este enfoque tiene el potencial de mejorar la eficiencia de la comunicación, pero requiere un intercambio adicional de información para difundir el horario de activación/reposo de cada nodo. Las Redes de Sensores Inalámbricos con ciclos aleatorios son más sencillas de diseñar, ya que no requieren sobrecarga adicional (Rout & Ghosh, 2012).

La clave para una operación de bajo ciclo de trabajo es la siguiente:



- Dormir la mayor parte del tiempo (radio off).
- Despertar e iniciar lo más rápidamente el procesamiento de la información captada en el medio de comunicaciones (radio on).
- Minimizar el trabajo y volver a dormir.



**Figura 4.** Descripción general del mecanismo de los ciclos de trabajo en WSN.

En la Figura 4 se muestra una descripción general de cómo sería un ciclo de trabajo en una WSN donde los módulos de radios se prenden periódicamente para entrar en periodo de *listen* y así revisar la información si es que existe en el medio inalámbrico. En general se garantiza que el módulo de radio este la mayor parte del tiempo apagada ahorrando así energía y alargando la vida útil de los nodos de la red.

#### 2.1.4 6LoWPAN

6LoWPAN (IPv6 *over Low power Wireless Personal Area Networks*) es un estándar desarrollado por *Internet Engineering Task Force* (IETF) *6LoWPAN Working Group*. Fue diseñado desde el principio para ser utilizado en sensores pequeños. Es un protocolo de comunicación simple de bajo costo que permite la conectividad inalámbrica en aplicaciones con requisitos de potencia limitada y baja capacidad de procesamiento, ya que proporciona la capacidad de conexión IPv6 siendo este demasiado "caro" para usar. Caro, en este caso, siendo una medida de tamaño de código, complejidad de protocolo, infraestructura de configuración requerida o *header overhead* a través de redes IEEE 802.15.4 (Mulligan, 2007). 6LoWPAN está formado por dispositivos compatibles con el estándar IEEE 802.15.4 y caracterizados por corto alcance, baja tasa de bits, bajo consumo de energía, bajo uso de memoria y bajo costo (Mahmood et al., 2015).

6LoWPAN *Working Group* también ha podido eliminar la necesidad de servidores de configuración (DHCP y NAT) utilizando las capacidades *ZeroConf* y *Neighbor Discovery* de IPv6. Al basarse el protocolo en IPv6, estos también pudieron definir un mecanismo exclusivo de compresión de encabezados sin estado que permite transmisión de paquetes IPv6 en tan solo 4 bytes. El protocolo IPv6 utiliza direcciones de 128 bits a diferencia de 32 bits en IPv4, con 128 bits, IPv6 puede proporcionar  $56 \times 10^{27}$  direcciones por persona o  $667 \times 10^{21}$  por metro cuadrado en la Tierra (Mulligan, 2007).

## 2.2 Simuladores de redes de telecomunicaciones

La simulación es una tecnología importante en la época moderna. En la simulación por computadora, es posible modelar objetos hipotéticos y de la vida real en una computadora para su estudio. Esto también se aplica a las redes, donde se utilizan simuladores de red. Un simulador de red es una técnica para implementar una red de sensores o de computadora. A través de esto, se calcula el comportamiento de la red ya sea mediante la interconexión de entidades de red utilizando fórmulas matemáticas o mediante la captura y reproducción de observaciones de una red de producción (Siraj et al., 2012).

Los simuladores de red permiten a los investigadores probar escenarios que son difíciles o costosos de desplegar en el mundo real. Son particularmente útiles para probar nuevos protocolos de red o cambios en los protocolos existentes en un entorno controlado y reproducible. Se pueden diseñar diferentes topologías de red utilizando varios tipos de nodos (hosts, concentradores, puentes, enrutadores y unidades móviles, etc.). Los simuladores de red se dividen en diferentes tipos que se pueden comparar en función de varios criterios, como alcance, especificación de los nodos y las conexiones entre esos nodos y el tráfico entre ellos, especificación de todo lo relacionado con los protocolos utilizados para manejar el tráfico en una red, aplicaciones gráficas, entre otras.

Las aplicaciones de WSN deben ser probadas a gran escala y en condiciones complejas y variables para capturar una gama lo suficientemente amplia de interacciones, tanto entre nodos como con el entorno. Un simulador de WSN consta de varios módulos, eventos, medio, entorno, nodo, transceptor, protocolos y aplicaciones. Cada categoría está representada por una interfaz que define sus métodos y los eventos generados y consumidos (Sundani et al., 2011).

1. **Evento:** Este módulo es una clase base abstracta que proporciona funcionalidad básica para todos los eventos en el simulador. Contiene información sobre el momento en que un evento debe

ocurrir y proporciona métodos para comparar eventos según sus tiempos de activación, determinar si los eventos son iguales, imprimir información sobre sí mismos en una cadena y un método abstracto para activar el evento.

2. **Medio:** El módulo de medio modela el medio inalámbrico. Permite que los nodos transmitan señales y es responsable de informar a los nodos sobre las señales que les afectan. Para lograr esto, el Medio debe estar al tanto de la presencia de cada nodo y de cualquier cambio en la posición o propiedades de radio, como la potencia del transmisor o la sensibilidad del receptor. El Medio también tiene propiedades como el ancho de banda y la longitud de onda del medio modelado, así como una referencia a un modelo de propagación que se le proporciona en el momento de la construcción.
3. **Entorno:** El módulo de entorno es similar al módulo de Medio, pero se enfoca en las propiedades físicas del entorno modelado. Además, el Entorno tiene un modelo de propagación que representa la propagación de los fenómenos físicos modelados. Los fenómenos físicos de interés en las redes de sensores incluyen temperatura, luz, humedad, campo magnético, sonido, óptica y presencia química.
4. **Nodo:** Este módulo representa un nodo individual en una red de sensores inalámbricos. Sirve como contenedor para todos los componentes, tanto hardware como software, en un nodo. Estos componentes incluyen procesador, transceptor, sensores, actuadores, fuente de energía (como una batería), protocolos de red y aplicaciones. Además, cada nodo tiene propiedades de ubicación e identificación.
5. **Transceptor:** El Transceptor modela el hardware del transceptor en cada nodo sensor. Representa los estados del transceptor (por ejemplo, dormir, en espera, recibir y transmitir) y su comportamiento y consumo de energía asociado. El Transceptor consume eventos que le informan sobre el inicio y el final de cada señal que recibe y genera eventos para el inicio y el final de cada señal que transmite. Estos eventos se intercambian con una instancia del módulo Medio.
6. **Protocolo Físico:** El Protocolo Físico es la capa más baja en una pila de red. A menudo se implementa en el hardware del transceptor. La capa física proporciona servicios para cambiar el estado del transceptor, la detección de portadora, el envío y la recepción de paquetes, la detección

de energía recibida en los paquetes recibidos y el cambio de canales en capas físicas que admiten múltiples canales.

7. **Protocolo MAC:** El Protocolo MAC es la siguiente capa en una pila de red. Por lo general, se implementa en software que se ejecuta en el procesador del nodo. La capa MAC proporciona servicios para cambiar el estado de la capa MAC (es decir, modo de bajo consumo), establecer y obtener parámetros de protocolo, enviar y recibir paquetes, etc. Un simulador de WSN generalmente ofrece implementaciones de varios protocolos MAC de redes de sensores.
8. **Protocolo de Enrutamiento:** El protocolo de enrutamiento reside por encima del protocolo MAC y proporciona servicios para enrutar mensajes a través de múltiples saltos entre nodos que no pueden comunicarse directamente.
9. **Capa de Aplicación:** Esta capa representa las aplicaciones de usuario que se ejecutan en los nodos de la red. Las aplicaciones pueden incluir recopilación de datos, procesamiento de datos, control de actuadores y más.

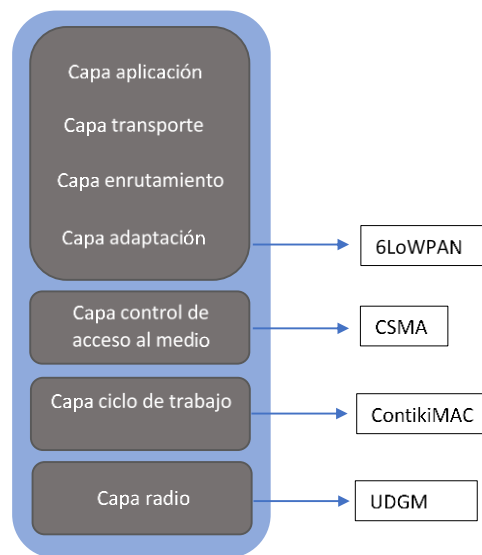
### 2.2.1 Contiki

Contiki OS es un sistema operativo de código abierto desarrollado desde principios de la década de 2000 por Adam Dunkels. Se basa en Ubuntu como sistema operativo base y utiliza Linux como kernel. Este sistema operativo está diseñado para su implementación en sistemas de pequeña escala, abarcando desde ordenadores de 8 bits hasta sistemas embebidos, incluyendo nodos de redes de sensores. Su principal enfoque se encuentra en aplicaciones relacionadas con la Internet de las Cosas (IoT). Este sistema operativo debe su nombre a la balsa que llevo el aventurero y etnógrafo noruego Thor Heyerdahl para demostrar que los primeros pobladores de la Polinesias provenían del continente Sudamericano y no de Asia como se creía en aquel entonces (Malik et al., 2018).

Una de las principales ventajas que tiene este sistema es que permite optimizar mucho el uso de memoria RAM y ROM del nodo, permitiendo por ejemplo operar utilizando el protocolo TCP/IP y aun así requerir solo de 10 kilobytes de memoria RAM y 30 kilobytes de ROM. Además las principales características que posee dicho sistema operativo son que el núcleo gestiona eventos los cuales pueden ser ordenados dinámicamente en el tiempo de ejecución, el código se estructura en *protothread* (protohilos), que sirven

para abastecer una idea secuencial a la raíz del evento, este a su vez puede soportar varios *protothreads*, permite varios protocolos de comunicación siendo para nosotros los más relevantes IPv6, debido a que hoy en día la aparición de IoT ha generado una mayor necesidad de conexión a internet por parte de dichos dispositivos sin la necesidad de usar redes privadas ni el uso de traducción de acceso a la red (NAT).

En la Figura 5 se muestra las 4 capas de la pila de protocolos usado en Contiki, y los protocolos usados para las simulaciones de WSN son mostrados a la parte derecha de la figura, las cuales fueron explicadas anteriormente exceptuando ContikiMAC a la cual se le dedicará una sección entera debido a la relevancia que tiene para este trabajo de investigación.



**Figura 5.** Stack de protocolos de Contiki.

### 2.2.1.1 COOJA

COOJA (abreviación de su nombre en inglés **CO**ntiki **O**s **JA**va) es una herramienta de simulación de redes proporcionada por el sistema operativo Contiki. Permite simular redes pequeñas y grandes de nodos de sensores de Contiki. COOJA también permite la emulación de nodos, tanto a nivel de hardware como a un nivel menos detallado, para inspeccionar con precisión el comportamiento del sistema y simular redes más grandes, respectivamente. Aunque COOJA ha sido desarrollado utilizando la plataforma Java, todos los nodos en el simulador COOJA se configuran utilizando códigos y *scripts* en lenguaje C para el análisis.

COOJA es una herramienta independiente que se instala previamente en el sistema operativo Contiki. Cada vez que se ejecuta utilizando la interfaz de línea de comandos de Contiki, es necesario compilarlo. COOJA permite el diseño completo de la red, junto con la creación de sistemas virtuales tanto en el mundo virtual como en el mundo real, conectando los nodos a los puertos USB del sistema. COOJA viene preinstalado con algunas extensiones como PowerTracker, Collect View, etc., que deben configurarse según las estructuras de Contiki, utilizando las especificaciones de las rutas de variables en COOJA(Thomson et al., 2016). Las simulaciones se almacenan en formato .csc y se pueden cargar fácilmente desde cualquier lugar, mientras que cargar los programas individuales de los nodos requiere una construcción adecuada de la arquitectura(Malik et al., 2018).

Esta herramienta tiene dos paquetes de software de emulador: Avrora y MSPSim. Cooja utiliza Avrora para la emulación de dispositivos basados en Atmel AVR, y MSPSim para la emulación de dispositivos basados en TI MSP430. La mayoría de las plataformas tienen microcontroladores MSP430(Roussel & Zendra, 2016), por lo que MSPSim es el paquete de software más utilizado para la simulación de WSN. Cooja puede emular múltiples plataformas como TelosB/SkyMote, Zolertia Z1 mote, Wismote, ESB, MicaZ mote. Es una herramienta muy útil para el desarrollo y depuración de aplicaciones de Contiki OS. Permite a los desarrolladores probar su código y sistemas antes de ejecutarlo en el hardware objetivo real, estimar el consumo de energía de los nodos en las simulaciones o mostrar transmisiones y recepciones de radio(Velinov & Mileva, 2016).

La ventana principal de Cooja muestra varias de estas herramientas, incluyendo:

1. **Salida de los nodos:** Permite obtener los registros de consola de cada nodo, lo que es valioso para depurar y entender el comportamiento de los nodos durante la simulación.
2. **Ventana de red:** Proporciona una visualización gráfica de la red, mostrando cada nodo con su ID, rango de transmisión e interferencia. Esto permite una representación visual de cómo están distribuidos los nodos en la red y cómo interactúan entre sí.
3. **Control de simulación:** Ofrece funciones para controlar la simulación, como iniciar, pausar, reiniciar o ejecutarla paso a paso según sea necesario. Esto permite tener un mayor control sobre el progreso de la simulación y observar eventos específicos en detalle.

4. **Línea de tiempo:** Muestra la actividad de cada nodo en la red, indicando cuándo están transmitiendo, recibiendo o experimentando interferencias con los paquetes. También muestra la actividad de los indicadores y los logs output de los nodos. Esta herramienta proporciona una visión detallada de lo que está sucediendo en la red en cada momento.
  
5. **Mote radio duty cycle:** Es una herramienta especialmente relevante para este trabajo de investigación, ya que brinda información sobre el uso del módulo de radio de cada nodo. Permite conocer cuánto tiempo permanece el módulo de radio encendido, cuánto de ese tiempo se utiliza en modo de transmisión y cuánto en modo de recepción. Esto es esencial para entender el consumo de energía de los nodos y optimizar su eficiencia.

En conjunto, estas herramientas de COOJA facilitan la observación y análisis de la simulación, lo que es fundamental para el desarrollo y la evaluación de protocolos y algoritmos en redes de sensores inalámbricos.

#### 2.2.1.2 UDGM (Unit Disk Graph Medium)

En COOJA, un medio de radio actúa como observador de la interfaz de radio de cada mote. Cuando la radio de un mote comienza a transmitir, se notifica al medio. Para decidir qué radios deben recibir el mensaje transmitido en la simulación, el medio de radio utiliza la ubicación de radio de cada nodo. En COOJA se utiliza el modelo de propagación de radio Unit Disk Graph Medium (UDGM), que representa el rango de transmisión idealmente como un disco, donde los motes que se encuentran fuera del disco no pueden recibir paquetes, mientras que los motes dentro de la región del disco reciben todos los paquetes.

El modelo de propagación UDGM considera la pérdida de distancia UDGM como una extensión de este modelo básico, que se utiliza de forma predeterminada y tiene en cuenta las interferencias en las transmisiones. En este modelo, los paquetes pueden ser recibidos y transmitidos con una probabilidad de éxito cada uno. Se utilizan dos parámetros de distancia: el rango de transmisión, que determina si los nodos están dentro del rango de transmisión del otro, y el rango de interferencia, que determina si habrá interferencia entre la transmisión de un nodo y la recepción de otro nodo. Un nodo que envía un paquete puede ser recibido por un nodo que se encuentra solo dentro del radio de transmisión definido de ese nodo. Fuera de este rango, pero dentro del rango de interferencia, el nodo no puede recibir el paquete del nodo emisor, pero enfrentará interferencia debido a la transmisión del nodo emisor. Sin embargo,

fuera del rango de interferencia, el nodo no tiene conocimiento de las transmisiones del nodo emisor (Malik et al., 2018).

El modelo de radio UDGM considera la antena como omnidireccional. Dentro del radio de transmisión de un nodo, la intensidad de la señal del receptor del mote está relacionada con la distancia desde el mote emisor por un factor único. Este factor de distancia se define como la relación entre la distancia y el rango de transmisión. La pérdida de trayecto se describe en términos sencillos mediante la ecuación 1 mostrada, donde se utilizan decibelios para medir todas las cantidades observables similares a la vida real. Esta es la ecuación de Friis. Sin embargo, el modelo UDGM existente no describe con precisión la pérdida de señal de intensidad de la vida real.

$$P_r = P_t + G_t + G_r - P_l \quad (1)$$

Donde:

$P_r$  : Potencia de la señal recibida en la antena receptora.

$P_t$  : Potencia de la señal que entrega la antena transmisora.

$G_t$  : Ganancia de la antena transmisora.

$G_r$  : Ganancia de la antena receptora

$P_l$  : Pérdida de trayectoria en el espacio libre (Rege, 2015).

### 2.3 ContikiMAC

Los nodos de sensor inalámbricos alimentados por batería son uno de los componentes fundamentales del tan anunciado "Internet de las cosas" (IoT). Para maximizar la vida útil de la batería, la radio que consume mucha energía se apaga periódicamente mediante un protocolo de "Ciclo de Trabajo de Radio" (RDC, **Radio Duty Cycling**, por sus siglas en inglés) incluido en la capa MAC de la pila de protocolos de comunicaciones (Uwase et al., 2017).

ContikiMAC utiliza únicamente mecanismos asíncronos, sin mensajes de señalización y sin encabezados de paquetes adicionales. Los paquetes de ContikiMAC son mensajes ordinarios de capa de enlace. ContikiMAC tiene un mecanismo de activación mucho más eficiente en términos de consumo de energía que los mecanismos de ciclo de trabajo anteriores. Esto se logra mediante una sincronización precisa a



través de un conjunto de restricciones de tiempo. Además, ContikiMAC utiliza una optimización de sueño rápido (*fast sleep*) para permitir que los receptores detecten rápidamente falsos positivos de activación, y una optimización de bloqueo de fase de transmisión para permitir la optimización en tiempo de ejecución de la eficiencia energética de las transmisiones.

ContikiMAC es un protocolo de ciclo de trabajo de radio que utiliza activaciones periódicas para escuchar transmisiones de paquetes de vecinos. Si se detecta una transmisión de paquete durante una activación, el receptor se mantiene encendido para poder recibir el paquete. Cuando el paquete se recibe correctamente, el receptor envía un acuse de recibo de capa de enlace. Para transmitir un paquete, el remitente envía repetidamente su paquete hasta recibir un acuse de recibo de capa de enlace por parte del receptor, como se observa en la Figura 6.

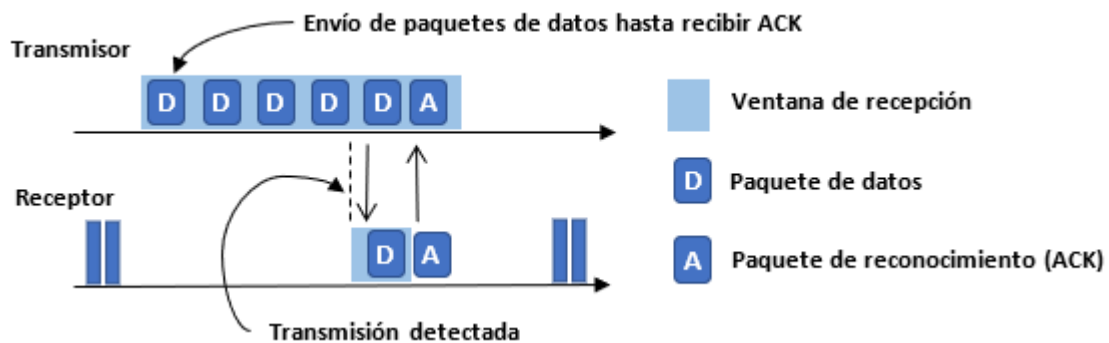


Figura 6. Diagrama de acción para el transmisor y el receptor usando ContikiMAC en transmisiones de tipo unicast

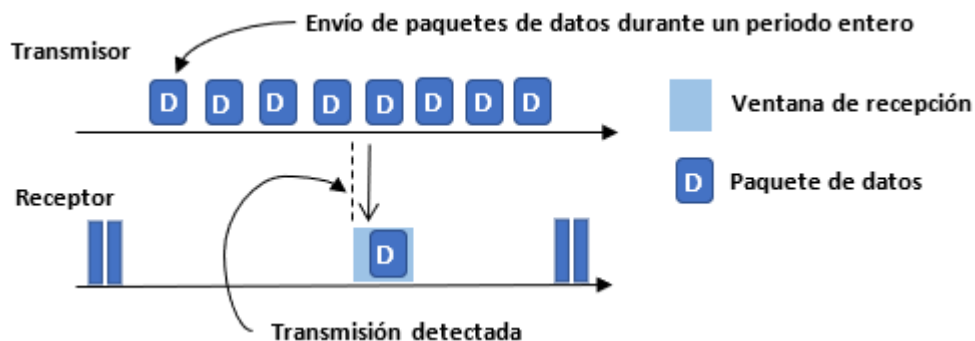


Figura 7. Diagrama de acción para el transmisor y el receptor usando ContikiMAC en transmisiones de tipo broadcast

Los paquetes que se envían de tipo broadcast no reciben acuses de recibo de capa de enlace. En su lugar, el remitente envía repetidamente el paquete durante todo el intervalo de activación para asegurarse de que todos los vecinos lo hayan recibido, como se muestra en la Figura 7.

### 2.3.1 ContikiMAC Timing

ContikiMAC cuenta con un mecanismo de *power-efficient wake-up* que se basa en una sincronización precisa entre transmisiones. Las activaciones de ContikiMAC utilizan un mecanismo económico de Evaluación de Canal Despejado (CCA, *Clear Channel Assessment*, por sus siglas en inglés) que utiliza el Indicador de Intensidad de Señal Recibida (RSSI, *Received Signal Strength Indicator*, por sus siglas en inglés) del transceptor de radio para dar una indicación de la actividad de radio en el canal. Si el RSSI está por debajo de un umbral dado, el CCA devuelve un resultado positivo, indicando que el canal está despejado. Si el RSSI está por encima del umbral, el CCA devuelve un resultado negativo, indicando que el canal está en uso. El *timing* de ContikiMAC se muestra en la Figura 8.

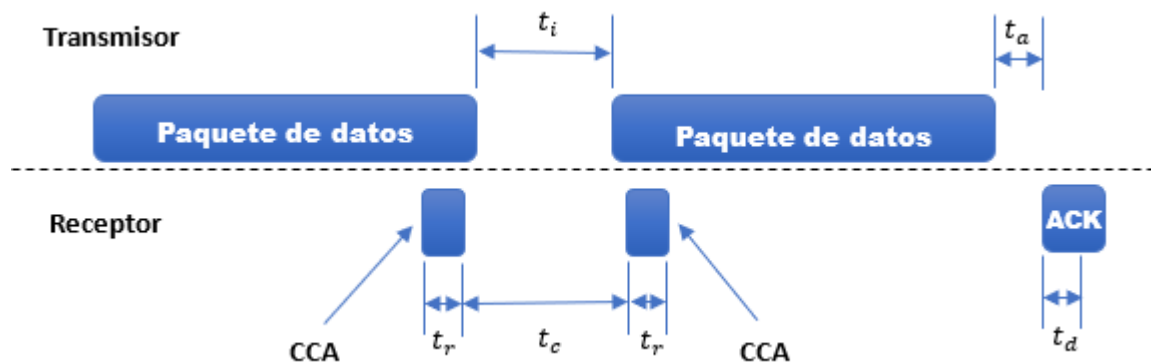


Figura 8. Distribución de los timing de ContikiMAC.

Donde:

$t_i$ : Tiempo entre cada paquete de transmisión.

$t_r$ : Tiempo requerido para un estable RSSI, necesario para un estable CCA.

$t_c$ : Intervalo entre cada CCA.

$t_a$ : Tiempo entre la recepción de un paquete y el envío de un paquete Ack.

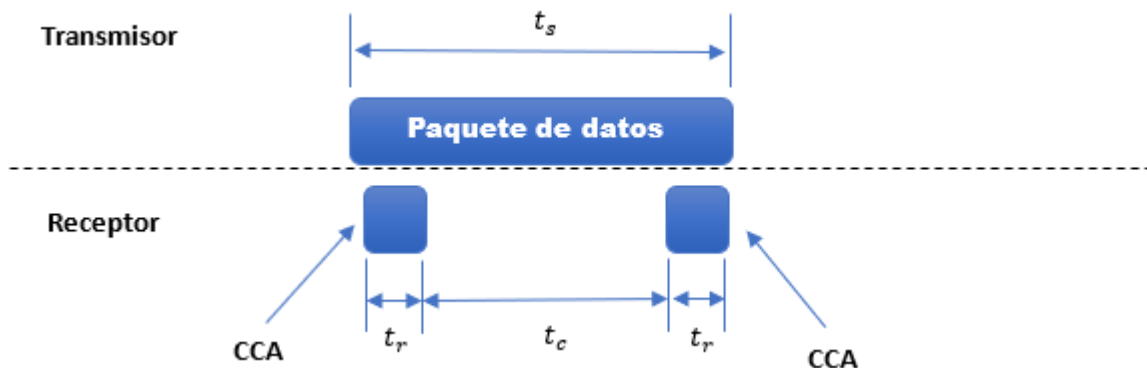
$t_d$ : Tiempo requerido para una detección satisfactoria de un paquete Ack.

Los **timing** deben satisfacer ciertas restricciones. Primero,  $t_i$  debe ser más pequeño que  $t_c$ . Esto se hace para garantizar que tanto la primera como la segunda CCA puedan detectar la transmisión del paquete. Si  $t_c$  fuera menor que  $t_i$ , dos CCAs no serían capaces de detectar de manera confiable una transmisión. El requisito de  $t_i$  y  $t_c$  también impone un requisito sobre el tamaño más corto de paquete que ContikiMAC puede admitir, como se muestra en la Figura 9. Para que las dos CCAs de ContikiMAC puedan detectar el paquete, una transmisión de paquete no puede ser tan corta como para caer entre las CCAs. Específicamente,  $t_s$ , el tiempo de transmisión del paquete más corto, debe ser mayor que  $t_r + t_c + t_r$ .

Cuando una CCA ha detectado una transmisión de paquete, ContikiMAC mantiene la radio encendida para poder recibir el paquete completo. Una vez que se ha recibido el paquete completo, se transmite un acuse de recibo de capa de enlace. El tiempo que lleva transmitir un paquete de acuse de recibo,  $t_a$ , y el tiempo que lleva detectar un paquete de acuse de recibo,  $t_d$ , establece el límite inferior para el intervalo de verificación  $t_c$  (Dunkels, 2011; Uwase et al., 2017).

Ahora podemos construir las restricciones de tiempo completas de ContikiMAC como:

$$t_a + t_d < t_i < t_c < t_c + 2t_r < t_s \quad (2)$$



**Figura 9.** El tiempo de transmisión de un paquete  $t_s$  debe ser lo suficientemente larga para que no se encuentre entre dos CCA.

### 2.3.2 Detección de paquetes y sueño rápido (fast sleep)

Las CCAs de ContikiMAC no detectan de manera confiable las transmisiones de paquetes: solo detectan que la intensidad de la señal de radio está por encima de un cierto umbral. La detección de una señal de

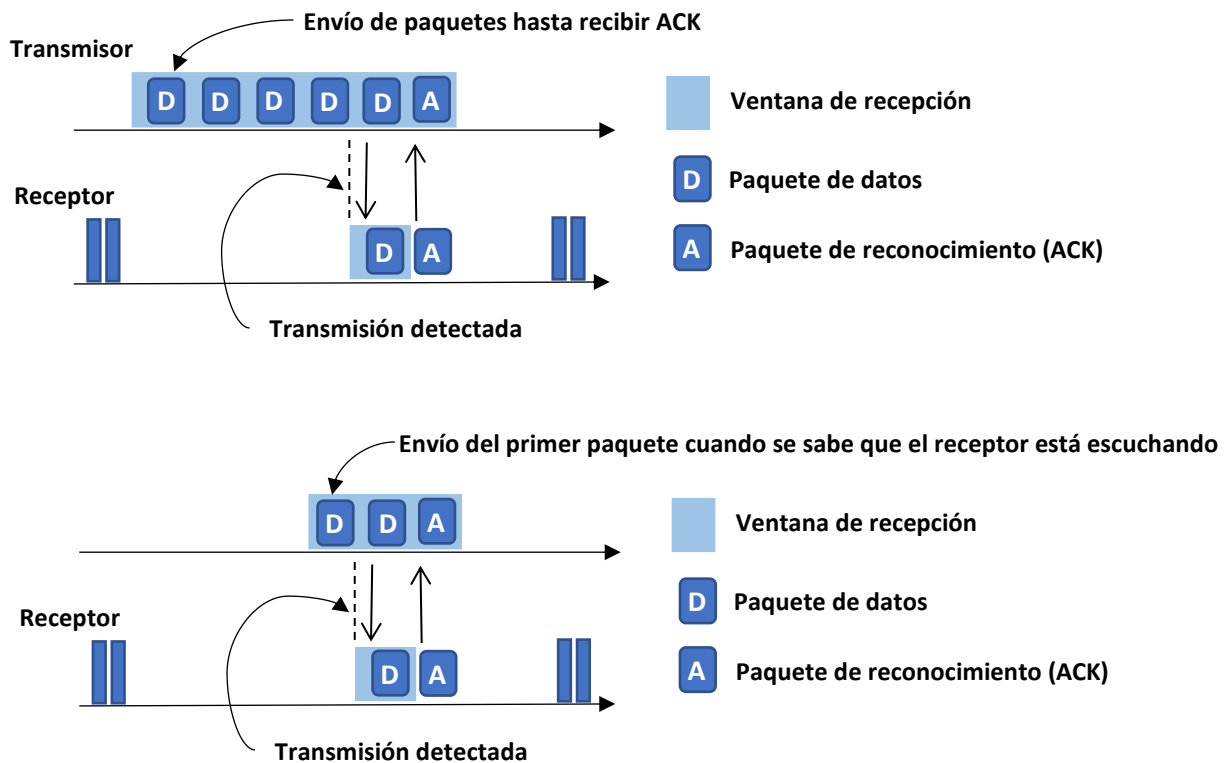
radio puede significar que un vecino está transmitiendo un paquete al receptor, que un vecino está transmitiendo a otro receptor o que algún otro dispositivo está emitiendo energía de radio que está siendo detectada por el mecanismo de CCA. ContikiMAC debe ser capaz de distinguir entre estos eventos y reaccionar adecuadamente. Si un vecino está transmitiendo un paquete al receptor, el receptor debe permanecer activo para recibir el paquete completo y transmitir un acuse de recibo de capa de enlace de ser necesario. Sin embargo, los receptores potenciales no pueden dormir rápidamente, ya que deben ser capaces de recibir el paquete completo.

La forma burda de determinar cuánto tiempo debe permanecer despierto un nodo cuando un CCA ha detectado actividad de radio es quedarse despierto por  $t_l + t_i + t_l$ , donde  $t_l$  es el tiempo de transmisión del paquete más grande posible. Esto asegura que, si la CCA se despertó al comienzo del paquete, el receptor recibirá el paquete completo.

La optimización de sueño rápido permite que los receptores potenciales se duerman antes si la CCA se activó debido a ruido de radio espurio. La optimización de sueño rápido aprovecha el conocimiento del patrón específico de las transmisiones de ContikiMAC de la siguiente manera. Primero, si la CCA detecta actividad de radio, pero la duración de la actividad de radio es mayor que la longitud máxima del paquete  $t_l$ , significa que la CCA ha detectado ruido y puede volver a dormir, es decir, si el período de actividad no es seguido por un período de silencio. Segundo, si la actividad de radio es seguida por un período de silencio que es mayor que el intervalo entre dos transmisiones sucesivas  $t_i$ , el receptor puede volver a dormir. Tercero, si el período de actividad es seguido por un período de silencio de la longitud correcta, seguido de actividad, pero no se pudo detectar el inicio del paquete, el receptor puede volver a dormir (Dunkels, 2011).

Otra funcionalidad que tiene ContikiMAC es el Bloque de Fase de Transmisión la cual no resulta relevante para este trabajo de investigación debido a que se usa para transmisiones de tipo unicast, pero vale la pena hacer mención de dicha funcionalidad. La optimización de bloqueo de fase de transmisión en ContikiMAC permite la optimización en tiempo de ejecución de la eficiencia energética de las transmisiones. Al sincronizar el tiempo de transmisión con la fase de activación del receptor, el remitente puede minimizar el consumo de energía y mejorar la confiabilidad de las transmisiones de paquetes. Una vez que el remitente ha aprendido la fase de activación de un receptor, puede ajustar el tiempo de transmisión para que coincida con la ventana de activación del receptor. Esta sincronización asegura que el paquete se transmita cuando el receptor esté despierto y listo para recibirlo. Al evitar intentos de transmisión innecesarios durante los períodos de sueño del receptor, la optimización de bloqueo de fase

de transmisión reduce el desperdicio de energía. Además, la optimización de bloqueo de fase de transmisión ayuda a mitigar la interferencia entre nodos vecinos. Al coordinar el tiempo de transmisión, los nodos adyacentes pueden minimizar las colisiones y mejorar el rendimiento general de la red. En resumen, la optimización de bloqueo de fase de transmisión en ContikiMAC permite una comunicación eficiente y confiable al optimizar el tiempo de transmisión de paquetes en función de la fase de activación del receptor. Una descripción grafica se puede observar en la Figura 10(Dunkels, 2011).



**Figura 10.** Bloque de transmisión de fase, el transmisor aprende sobre las fases donde se despierta el receptor y la acomoda a la siguiente transmisión, lo cual causa que el transmisor necesite enviar menos paquetes.

## 2.4 Algoritmos de diseminación broadcast

Es bien sabido que la redundancia excesiva en los mensajes **broadcast**, como resultado de una **broadcast storm**, conduce a una grave contención en la capa de enlace, colisiones de paquetes, un uso ineficiente del ancho de banda y la potencia de procesamiento, y, lo que es más importante, la interrupción del servicio debido a una alta contención (Wisitpongphan et al., 2007). Un **broadcast storm** es un fenómeno no deseado que ocurre en redes de comunicación, cuando un mensaje de **broadcast** se replica y se reenvía

de manera exponencial por múltiples nodos de la red. Esto puede generar una congestión severa y provocar un alto consumo de ancho de banda de red.

El **broadcast storm** se origina cuando un nodo de la red envía un paquete de difusión a todos los demás nodos, y luego algunos de esos nodos reenvían ese paquete nuevamente a todos los nodos, incluido el nodo original. Luego, otros nodos también replican y reenvían el paquete, creando así un ciclo de reenvío infinito que aumenta exponencialmente la cantidad de tráfico de difusión en la red.

Los algoritmos de difusión son ampliamente utilizada en redes ad-hoc, como las WSNs(Tilak et al., 2003), las Redes Ad Hoc Móviles (MANETs por sus siglas en inglés)(Hoebeke et al., 2004) y las Redes Ad Hoc Vehiculares (VANETs por sus siglas en inglés)(Wisitpongphan et al., 2007). En los protocolos de enrutamiento para redes ad-hoc, los paquetes de **broadcast** forman parte de la fase de descubrimiento, que se encarga de encontrar una ruta de comunicación para dirigir los datos de una aplicación desde un nodo fuente a uno o varios nodos de destino (enrutamiento unicast o multicast)(Williams & Camp, 2002).

Los paquetes de **broadcast** también se utilizan en el mantenimiento de rutas, ya que los nodos intercambian paquetes de saludo (**Hello**) para recopilar información sobre sus vecinos. Además, se emplea en la diseminación de mensajes de emergencia o advertencia en escenarios de desastre, que es una de las principales aplicaciones de las MANETs. En condiciones adversas como estas, un mecanismo de **broadcast** eficiente es vital y puede ser la única forma posible de transmitir información crucial. Además, en las VANETs, la diseminación de mensajes de advertencia también es importante para alertar a los conductores sobre congestiones debidas a accidentes de tráfico. En resumen, la técnica de **broadcast** desempeña un papel fundamental en la comunicación y el enrutamiento de datos en redes ad hoc, así como en la difusión de información importante en situaciones críticas y en entornos de movilidad, como el tráfico vehicular(Reina et al., 2015).

#### 2.4.1 Técnicas de mitigación del broadcast storm

La forma más simple de difusión es el **flooding**, en el cual cada nodo en la red retransmite un mensaje entrante una vez. El principal beneficio de usar el inundamiento es la facilidad de implementación. Sin embargo, esta técnica también presenta varios problemas, uno de ellos es su ineficiencia en términos de consumo de recursos como el ancho de banda y la energía. Mientras que lo primero afecta a la red en términos de eficiencia y capacidad, lo último es crucial para la vida útil de la red. El principal problema del

inundamiento en redes inalámbricas ad-hoc es que causa el ya antes mencionado problema de la **broadcast storm**(Ni et al., 1999) debido a colisiones y contención. Dado que todos los nodos comparten el mismo medio inalámbrico, el problema de la tormenta de difusión lleva a una dramática deterioración del rendimiento en redes inalámbricas ad-hoc.

Para aliviar el problema de la tormenta de difusión, se han propuesto varios protocolos de difusión en la última década(al-Humoud et al., 2008; Jung et al., 2018; Williams & Camp, 2002; Wisitpongphan et al., 2007). Una clasificación básica de los esquemas de difusión los divide en dos categorías: esquemas deterministas y esquemas probabilísticos. En las técnicas deterministas, solo un subconjunto de nodos puede participar en el proceso de difusión. **Multi Point Relay** (MPR)(Liang et al., 2006) y **Connected Dominating Set** (CDS) son algunos ejemplos de algoritmos de difusión determinista. Sin embargo, esto podría llevar a un uso repetido de los mismos nodos. Además, bajo condiciones de movilidad, este conjunto de nodos debería cambiar con mucha frecuencia debido a los cambios topológicos.

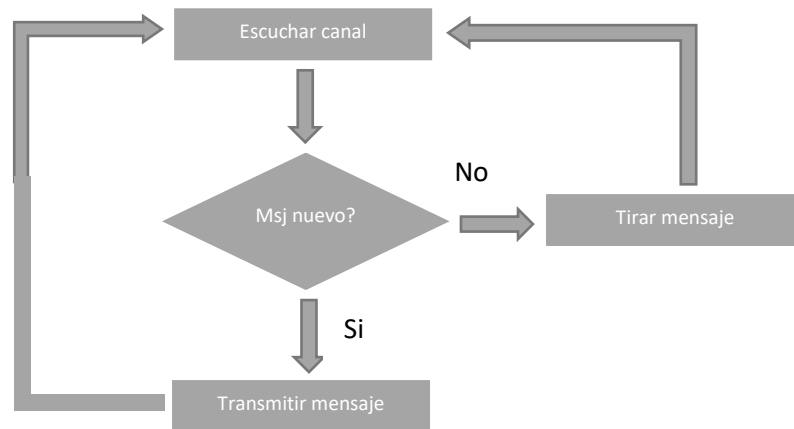
Los esquemas de difusión probabilística, en cambio, equilibran el consumo de energía entre todos los nodos de la red al seleccionar rutas bien equilibradas a lo largo de la vida útil de la red. En la difusión probabilística, los nodos reenvían los paquetes de difusión entrantes según un cierto valor de probabilidad, lo que permite que todos los nodos participen en el proceso de difusión. Además, los esquemas probabilísticos son más robustos contra fallos, ataques y no se ven afectados por la movilidad de los nodos, a diferencia de los esquemas deterministas(Reina et al., 2015).

#### 2.4.1.1 Esquemas de probabilidad de reenvío fija

Una probabilidad de reenvío en la cual los mensajes se reenvían con algún valor de probabilidad es el punto focal de todos los protocolos de enrutamiento basados en **gossip** (rumor). Reenviar mensajes con cierta probabilidad mejora significativamente el problema de la tormenta de difusión que ocurre en las técnicas de **flooding**. Por otro lado, la latencia y la terminación temprana del **gossip** son los principales problemas asociados con los protocolos basados en esta técnica(Shi & Shen, 2004; Zhang & Agrawal, 2005).

**Flooding:** Es una técnica de difusión en redes de comunicación en la cual un nodo o dispositivo reenvía un mensaje entrante a todos los demás nodos o dispositivos en la red. En otras palabras, un mensaje se difunde de manera indiscriminada a todos los destinos posibles en la red como se puede observar en el

diagrama de flujo de la Figura 11. Este método de difusión es simple de implementar, pero puede generar una alta redundancia y consumo de recursos, ya que cada nodo reenvía el mensaje a todos sus vecinos, lo que puede resultar en una congestión de la red y una utilización ineficiente del ancho de banda (Kifle et al., 2019). El **flooding** es útil en algunas situaciones, como cuando se necesita garantizar que un mensaje alcance a todos los nodos en una red sin importar la topología o la ubicación de los nodos. Sin embargo, también puede causar problemas como la tormenta de difusión en la que múltiples nodos reenvían el mismo mensaje repetidamente, lo que lleva a una saturación de la red (Haas et al., 2002).



**Figura 11.** Diagrama de flujo (**Flooding**).

**GOSSIP1(p):** Una fuente envía un mensaje de tipo **broadcast** con una probabilidad de 1. Cuando un nodo recibe por primera vez un mensaje de tipo **broadcast**, con una probabilidad  $p$ , retransmite el mensaje a sus vecinos, y con una probabilidad de  $1-p$ , descarta el mensaje. Si el nodo recibe mensaje nuevamente, la descarta. De esta manera, un nodo retransmite el mensaje de tipo **broadcast** como máximo una vez. A este protocolo simple se le llama GOSSIP1(p) (Haas et al., 2002, 2006) como se muestra en el diagrama de flujo de la Figura 12. El principal problema de GOSSIP1 es determinar la probabilidad óptima de retransmisión para un escenario dado. En (Haas et al., 2006), los autores demostraron que la probabilidad óptima se encuentra en el rango  $[0.6, 0.8]$  en escenarios donde el número de nodos es inferior a 1000.

**GOSSIP1(p, k):** GOSSIP1(p) tiene un pequeño problema con las condiciones iniciales. Si la fuente tiene relativamente pocos vecinos, existe una buena probabilidad de que ninguno de ellos realice la retransmisión y que la propagación de la información se detenga. Para evitar que esto suceda, el algoritmo de **broadcast** GOSSIP1(p, k) retransmite con una probabilidad de 1 durante los primeros  $k$  saltos antes de continuar con la probabilidad  $p$ . El rendimiento de GOSSIP1(p, k) claramente depende de la elección de  $p$ .



y  $k$ . Claramente, GOSSIP1(1,1) es equivalente a **flooding** y GOSSIP1( $p$ ,0) es equivalente al algoritmo previamente mencionado de GOSSIP1( $p$ ). Para obtener un mejor rendimiento, el valor de  $k$  debe elegirse cuidadosamente. Los autores establecieron  $k$  en 1 para redes densas y en 4 para redes dispersas. (Haas et al., 2006; Kifle et al., 2019; Reina et al., 2015).

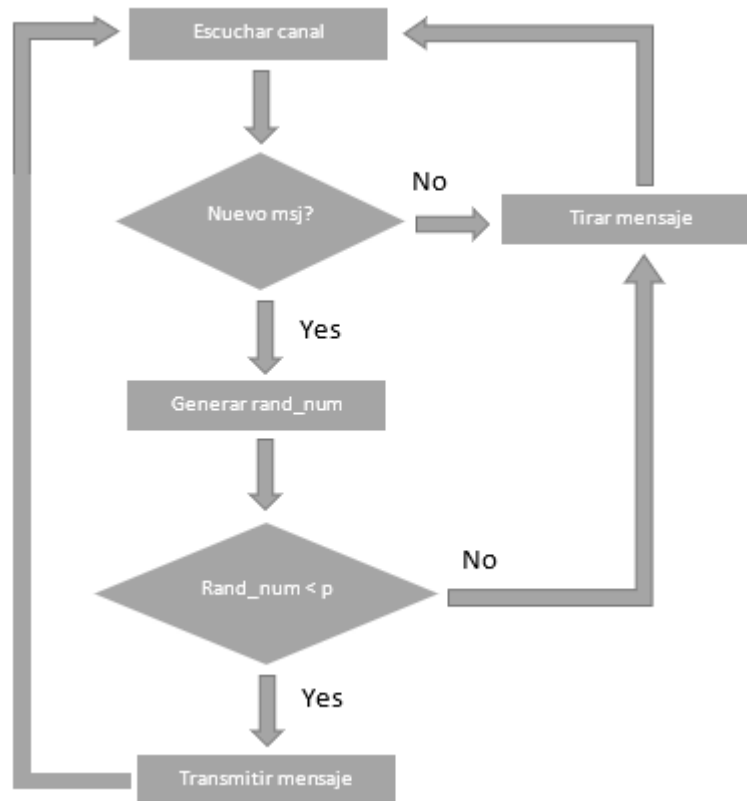


Figura 12. Diagrama de flujo GOSSIP1( $p$ ).

**GOSSIP2( $p_2, p_1, k, n$ ):** En una red aleatoria, a diferencia de una cuadrícula, un nodo puede tener muy pocos vecinos. En este caso, la probabilidad de que ninguno de los vecinos del nodo propague el **broadcast** es alta. En general, es posible que deseemos que la probabilidad de retransmitir el **broadcast** en un nodo sea una función de su grado, donde los nodos con menor grado difunden el broadcast con una probabilidad más alta. Para mostrar el efecto de esta mejora, (Haas et al., 2002, 2006) propuso un protocolo con cuatro parámetros,  $p_1$ ,  $k$ ,  $p_2$  y  $n$ . Como en GOSSIP1,  $p_1$  es la probabilidad de retransmitir típica, pero el rumor ocurre con probabilidad 1 para los primeros  $k$  saltos. Las nuevas características son  $p_2$  y  $n$ ; la idea es que los vecinos de un nodo con menos de  $n$  vecinos difundan el rumor con probabilidad  $p_2 > p_1$ . Es decir, si un nodo tiene menos de  $n$  vecinos, instruye a sus vecinos inmediatos a difundir con probabilidad  $p_2$  en

lugar de  $p_1$ . Los parámetros de este algoritmo al ser modificaciones de los anteriormente mencionados el autor en (Haas et al., 2002, 2006) propone los valores de  $k$  en 1 para redes densas, 4 para redes dispersas y los valores de las probabilidades entre 0.6 y 0.8.

**GOSSIP3( $p, k, m$ ):** Este algoritmo es una optimización de GOSSIP1 y GOSSIP2, consiste en lo siguiente: si un nodo con  $n$  vecinos recibe un mensaje y no lo difunde, pero luego no recibe el mensaje de al menos  $m$  vecinos dentro de un período de tiempo razonable, difunde el mensaje a todos sus vecinos. La pregunta obvia aquí es cuánto debería ser  $m$ . Si  $m$  se elige demasiado grande, podríamos terminar con demasiados mensajes. En (Haas et al., 2002, 2006; Kifle et al., 2019) muestran que obtenemos la mejora de rendimiento más significativa tomando  $m = 1$ . Dejemos que GOSSIP3( $p, k, m$ ) sea igual que GOSSIP1( $p, k$ ), excepto por la siguiente modificación. Un nodo que originalmente no retransmitió un mensaje recibido (porque su lanzamiento de moneda salió cruz), pero luego no recibió el mensaje de al menos  $m$  distintos nodos dentro de un período de tiempo determinado, retransmite el mensaje inmediatamente después del período de tiempo de espera. (La elección del período de tiempo de espera debe ser tal que no afecte los parámetros de latencia de la red de manera significativa).

#### 2.4.1.2 Esquemas probabilísticos adaptativos

Los "Esquemas probabilísticos adaptativos" se refieren a enfoques de difusión en redes que ajustan sus probabilidades de reenvío de mensajes en función de las condiciones de la red o el entorno. En lugar de tener una probabilidad fija para retransmitir un mensaje, estos esquemas adaptativos permiten a los nodos cambiar dinámicamente su probabilidad de reenvío en función de diversos factores, como la congestión de la red, la densidad de nodos, la energía disponible, entre otros (Reina et al., 2015).

Algunos esquemas basados en contadores son:

**Density Aware Probabilistic Flooding (DAPF):** Es un enfoque de difusión de mensajes utilizado en redes ad-hoc y otras redes inalámbricas. Este enfoque tiene como objetivo optimizar la difusión de mensajes a través de la red de manera eficiente, teniendo en cuenta la densidad de nodos en la red. Cada nodo calcula su probabilidad de retransmisión tal y como se muestra en la ecuación 3.

$$p = \frac{k}{n} \quad (3)$$

donde  $k \geq 1$ , es el parámetro de eficiencia y  $n > 0$  es el grado del nodo (número de vecinos). Un nodo con menos vecinos tendrá mayor probabilidad de retransmisión y viceversa (Kifle et al., 2019). En (Cartigny & Simplot, 2003) proponen varios valores para el parámetro  $k$  para la evaluación del desempeño de este algoritmo. DAPF utiliza información sobre la topología de vecindario en la red. Cada nodo envía periódicamente mensajes "HELLO" (mensajes cortos que informan a los vecinos de la presencia del nodo). Estos mensajes HELLO permiten que cada nodo conozca a sus vecinos y, por lo tanto, la densidad local de nodos en su entorno.

**Weighted  $p$ -Persistence Broadcasting:** Cuando un nodo  $j$  recibe un paquete de un nodo  $i$ , verifica el identificador del paquete y lo retransmite con una probabilidad  $p_{ij}$  si lo recibe por primera vez; de lo contrario, descarta el paquete. Esta probabilidad de reenvío,  $p_{ij}$ , se calcula en función de la distancia relativa entre los nodos  $i$  y  $j$  ( $D_{ij}$ ) y el rango promedio de transmisión ( $R$ ) utilizando la siguiente expresión:

$$p_{ij} = \frac{D_{ij}}{R} \quad (4)$$

Es importante notar que si el nodo  $j$  recibe paquetes duplicados de múltiples fuentes dentro del período de espera `WAIT_TIME` (por ejemplo, 2 ms) antes de la retransmisión, selecciona el valor más pequeño de  $p_{ij}$  como su probabilidad de reenvío. Es decir, cada nodo debe usar la distancia relativa al emisor más cercano para asegurarse de que los nodos que están más lejos transmitan con una probabilidad más alta. Si el nodo  $j$  decide no volver a transmitir, debe almacenar el mensaje durante un tiempo adicional de `WAIT_TIME +  $\delta$`  ms, donde  $\delta$  es el retardo de transmisión y propagación de un salto, que generalmente es menor que `WAIT_TIME`. Para evitar que el mensaje se pierda y garantizar un alcance del 100 por ciento, el nodo  $j$  debe volver a transmitir el mensaje con probabilidad 1 después de `WAIT_TIME +  $\delta$`  ms si no escucha la retransmisión de sus vecinos (Wisitpongphan et al., 2007).

**Slotted  $p$ -Persistence Broadcasting:** Al recibir un paquete, un nodo verifica el identificador del paquete y lo retransmite con una probabilidad predefinida  $p$  en el intervalo de tiempo asignado  $TS_{ij}$ , como se expresa en la ecuación 5, si recibe el paquete por primera vez y no ha recibido duplicados antes de su intervalo de tiempo asignado; de lo contrario, descarta el paquete. En este esquema, cada nodo debe almacenar en

búfer el mensaje durante un cierto período de tiempo y volver a transmitir con probabilidad 1 si nadie en el vecindario retransmite para evitar que el mensaje se pierda (Wisitpongphan et al., 2007).

$$Ts_{ij} = S_{ij} * \tau \quad (5)$$

donde  $\tau$  es el retardo estimado a un salto, en el cual se incluye el retardo de acceso al medio y el retardo de propagación. Y  $S_{ij}$  es el número de slot asignados y puede expresarse como:

$$S_{ij} = Ns \left( 1 - \left[ \frac{\min(D_{ij}, R)}{R} \right] \right) \quad (6)$$

**Received-signal-strength-based schemes:** La técnica de difusión propone utilizar la información de la RSS en lugar de la información de GPS. Es importante destacar que la medición instantánea de la RSS solo puede proporcionar una estimación aproximada de la distancia correspondiente entre los nodos transmisor/receptor debido a la interferencia *multipath* (desvanecimiento múltiple). Para eliminar el efecto de desvanecimiento a pequeña escala y obtener una estimación más precisa de la distancia relativa al transmisor, cada nodo debería sonar periódicamente a sus vecinos para hacer un seguimiento de la RSS promediada en el tiempo, lo que puede representar mejor la distancia real de un nodo al transmisor. Sin embargo, hacerlo puede aumentar la carga de tráfico en el sistema, lo que puede no ser deseable. Por lo tanto, en ausencia de una señal GPS y sondas periódicas de vecinos, cada nodo puede, como máximo, obtener la RSS del paquete de difusión recibido del nodo transmisor y determinar si retransmitir o no el paquete en función de la RSS instantánea medida y el conocimiento previo de la potencia de transmisión y la sensibilidad del receptor, como se puede observar en la siguiente formulación matemática.

$$RSS_{range} = RSS_{max} - RSS_{min} \quad (7)$$

donde  $RSS_{max}$  y  $RSS_{min}$  corresponden a los valores máximos y mínimos posibles de **RSS** medidos en el entorno considerado; estos valores pueden obtenerse experimentalmente o calcularse aplicando un modelo de propagación adecuado. Sabiendo esto podemos calcular la probabilidad de retransmisión de la siguiente manera:

$$P_{ij} = \frac{RSS_{ij} - RSS_{min}}{RSS_{range}} \quad (8)$$

donde  $RSS_{ij}$  es el RSS del mensaje de *broadcast* proveniente del nodo j(Wisitpongphan et al., 2007).

### 2.4.1.3 Esquemas basados en contadores

En los esquemas basados en contadores, se introducen retrasos de acceso aleatorio (RAD) para contar cuántas veces se recibe un mismo mensaje durante ese período de tiempo y luego realizar la retransmisión basándose en un umbral predefinido. La principal dificultad en estos enfoques radica en asignar correctamente los umbrales de conteo y los tiempos de retraso(Mohammed et al., 2007, 2008, 2009).

**Counter-Based Scheme:** Cuando un nodo recibe un paquete que no ha visto antes, inicia un contador con un valor de uno y establece un retraso de acceso aleatorio (RAD) que se elige al azar entre 0 y un valor máximo (Tmax) en segundos.

```

*****
Counter-based (RAD, Cth)
On receiving a message:
if(message is not in list)
then
    counter ← 1
    store message to list
    start new random timer at interval[0,RAD]
else
    counter ++
On timer stop:
if(counter<= Cth)
then
    message action ← broadcast
else
    message action ← drop
process message action
remove message from list
*****

```

Figura 13. Pseudo Código simple counter-based.

Durante el período de RAD, el contador se incrementa en uno por cada paquete redundante recibido. Si el contador es menor que un valor umbral (Cth) cuando expira el RAD, se retransmite el paquete. De lo contrario se descarta el paquete como se puede observar en la Figura 13 (de Melo Pires et al., 2016; Mohammed et al., 2009; Williams & Camp, 2002). Aunque varios autores en la literatura calculan el RAD para este algoritmo se plantea que el RAD debe ser tal que no afecte de manera significativa los parámetros de la red (Mohammed et al., 2008). El rendimiento de este esquema se ve fuertemente afectado por el valor umbral de conteo y el retraso de acceso aleatorio. Por ejemplo, cuando se establece un valor umbral de conteo demasiado bajo o se utiliza un retraso de acceso aleatorio demasiado corto, disminuye la posibilidad de que un nodo retransmita los mensajes de difusión (Jung et al., 2018).

**Distance-Aware Counter-Based Broadcast (DCB) Scheme:** Es un algoritmo en el cual un nodo que recibe un mensaje de difusión determina un RAD basado en la distancia entre él mismo y el nodo que realiza la difusión (C. Chen et al., 2005). Si  $T_{max}$  es el máximo retraso de acceso aleatorio,  $R$  es el rango máximo de transmisión del nodo y  $D$  es la distancia entre los nodos que realizan la difusión y los nodos receptores, entonces, para que un nodo que esté lejos del nodo que realiza la difusión use un RAD corto, se define el RAD de la siguiente manera:

$$RAD = rand[0, 1] * T_{max} * \frac{R^2 - D^2}{R^2} \quad (9)$$

donde  $rand[0, 1]$  es una variable aleatoria uniformemente distribuida entre 0 y 1. En DCB, Cth se define como un valor fijo. En DCB, la distancia entre los nodos que realizan la difusión y los nodos receptores se puede obtener de un dispositivo de sistema de posicionamiento global (GPS). Sin embargo, cuando la información del GPS no está disponible o no es confiable, se puede aplicar un método basado en la disimilitud, en el cual el número de vecinos compartidos por dos nodos se utiliza como métrica de rendimiento para estimar la distancia relativa entre ellos (Reina et al., 2016).

**Neighborhood-Aware Counter-Based Broadcast (NCB) Scheme:** En (al-Humoud et al., 2008) propuso un esquema NCB en el cual el RAD se determinaba en función del número de nodos vecinos. Se introdujo un factor aleatorio RF el cual se establece en uno de dos valores:  $RF_1$  o  $RF_2$ , donde  $RF_1 < RF_2$ . Cuando un nodo recibe un mensaje de difusión, verifica el número de sus nodos vecinos,  $n$ , en comparación con el número promedio de nodos vecinos,  $n_{avg}$ . Si  $n < n_{avg}$  ( $n \geq n_{avg}$ ), la red se considera dispersa (densa) y

RF se establece en  $RF_1$  ( $RF_2$ ). De esta manera, un nodo con un gran (pequeño) número de nodos vecinos determina un RAD corto (largo) de la siguiente manera:

$$RAD = \frac{rand[0, 1]}{RF} \quad (10)$$

**Battery-Aware Counter-Based Broadcast (BCB) Scheme:** Es un esquema de difusión basado en contadores que toma en cuenta el remanente de la batería, en el cual un nodo que recibe un mensaje de difusión determinaba tanto  $C_{th}$  como RAD en función del nivel de batería restante. Sea  $C_{max}$ ,  $B_R$  y  $B_{max}$  el valor umbral de conteo máximo, la batería restante de un nodo receptor y la carga máxima de la batería, respectivamente (Z. Chen et al., 2017; Jung et al., 2018; Yassein et al., 2011). Entonces, para que un nodo con un nivel alto de batería emplee un valor alto de  $C_{th}$ , el cual se define de la siguiente manera:

$$C_{th} = ceil\left(C_{max} * \frac{B_R}{B_{max}}\right) \quad (11)$$

donde  $ceil(x)$  es la función de techo, que proporciona el entero más pequeño mayor o igual a x. Por otro lado, RAD se define como:

$$RAD = rand[0, 1] * 2(C_{max} - C_{th}) \quad (12)$$

## 2.5 Métricas de desempeño

Las métricas de eficiencia de algoritmos de **broadcast** evalúan los esquemas de difusión como una técnica de difusión independiente. Estas métricas son especialmente útiles en casos en los que los algoritmos de **broadcast** se utilizan para difundir mensajes de advertencia. En el proceso de investigación el autor de esta tesis consideró 3 métricas de eficiencia de difusión para evaluar: 1) Alcance (**Reachability**), 2) Retransmisiones evitadas (**SRB**) y 3) Retardo (**Delay**) (Cartigny & Simplot, 2003; Jung et al., 2018; Reina et al., 2015; Wisitpongphan et al., 2007). A continuación, se explica cada una de ellas a detalle.

### 2.5.1 Reachability

La métrica de alcance (**Reachability**) se define como la proporción de nodos que recibieron los paquetes de **broadcast** con respecto al número total de nodos en la red, ver ecuación 13. Cuando la red no está completamente conectada, la métrica de alcance se define como la proporción de nodos que recibieron los paquetes de **broadcast** con respecto al número total de nodos a los que puede llegar el nodo fuente directamente o a través de un camino multi-salto. Esta métrica también se conoce como la fracción de nodos, Ratio de Entrega de Difusión (BDR), Ratio de Cobertura de Difusión (BCR) o simplemente Cobertura.

$$Re = \frac{1}{n} \sum_{i=0}^{99} \frac{N_{rbi}}{N_r} \quad (13)$$

Donde  $N_{rb}$  es el número de nodos que reciben el **broadcast**,  $N_r$  el número de nodos que pueden ser alcanzados directa o indirectamente por una fuente y  $n$  es el número de paquetes lanzados en la simulación, en este caso fueron 100. En otras palabras, la métrica de alcance evalúa qué tan efectivamente se han difundido los paquetes en la red y cuántos nodos los han recibido en relación con el tamaño total de la red. Esta métrica es esencial para comprender la eficacia de un algoritmo de **broadcast** en términos de llegar a los nodos destinatarios y determinar si se ha logrado una cobertura adecuada en la red, especialmente en entornos donde no todos los nodos pueden estar al alcance directo del nodo emisor.

### 2.5.2 Saved Rebroadcast (SRB)

**Saved ReBroadcast(SRB)** se refiere a una métrica que evalúa la eficiencia de una estrategia de difusión al determinar cuántas retransmisiones innecesarias se han evitado mediante la estrategia de difusión utilizada.

Para calcular las retransmisiones guardadas, primero se debe contar el número de nodos que recibieron el mensaje de difusión  $N_{rb}$  y el número de nodos que realmente transmitieron el mensaje  $N_{tb}$ . Luego, las retransmisiones guardadas se definen como la diferencia entre el número de nodos que recibieron el mensaje y el número de nodos que lo transmitieron dividido al número de nodos que retransmiten el **broadcast**, es decir:



$$SRB = \frac{1}{n} \sum_{i=0}^{99} 1 - \frac{N_{tbi}}{N_{rbi}} \quad (14)$$

Este valor representa cuántas retransmisiones innecesarias se evitaron gracias a la estrategia de difusión utilizada. Cuantas más retransmisiones guardadas haya, más eficiente será la estrategia de difusión, ya que se están utilizando menos recursos de red para alcanzar el mismo alcance en la difusión. Esto puede ser crucial en redes inalámbricas, donde la eficiencia energética y de ancho de banda es fundamental.

### 2.5.3 Delay

El retardo (*delay*) están relacionadas con el tiempo que un mensaje de *broadcast* tarda en cruzar la red. Por regla general, un retardo bajo siempre es una característica deseable de un esquema de difusión, especialmente cuando se utiliza para difundir mensajes de emergencia.

El retardo se define como el tiempo que transcurre desde que se inicia el *broadcast* de un mensaje hasta que todos los nodos destinatarios han recibido efectivamente ese mensaje. En el contexto de mensajes de emergencia, un retardo bajo es esencial, ya que significa que la información crítica se distribuye rápidamente a todos los nodos de la red. Cuanto menor sea el retardo, más rápida será la respuesta a la situación de emergencia.

Las métricas de retardo pueden medirse de diversas formas, como el tiempo promedio que tarda un mensaje en llegar a todos los nodos siendo esta la de mayor relevancia para este trabajo de investigación, el tiempo máximo que tarda un mensaje en llegar a cualquier nodo, o el tiempo necesario para que un cierto porcentaje de nodos reciba el mensaje.

$$delay = t_f - t_i \quad (15)$$

Donde:

$t_f$  : tiempo final.

$t_i$  : tiempo inicial.

## 2.6 Método de Montecarlo

El método Montecarlo es una técnica de simulación estadística que se utiliza para resolver problemas complejos y estimar resultados mediante la generación de muestras aleatorias. Su nombre proviene de la ciudad de Montecarlo en Mónaco, conocida por su actividad de juegos de azar, y fue desarrollado durante la Segunda Guerra Mundial como parte del Proyecto Manhattan para el diseño y análisis de armas nucleares. Fue ideado por Stanislaw Ulam, John von Neumann y Nicholas Metrópolis como un enfoque innovador para abordar problemas que involucraban cálculos matemáticos complicados y procesos estocásticos.

La aplicabilidad del método Montecarlo es amplia y diversa, encontrando aplicaciones en áreas como física, ingeniería, finanzas, ciencias de la computación, ciencias sociales, estadísticas, entre otras (Kroese et al., 2014). Se utiliza para estimar resultados que no pueden ser resueltos analíticamente o cuando se enfrentan problemas con múltiples incertidumbres y variables. Algunos ejemplos comunes de aplicaciones del método Montecarlo incluyen:

1. Integración numérica.
2. Simulación de sistemas complejos.
3. Optimización y toma de decisiones.
4. Estimación de probabilidades y estadísticas.
5. Modelado y simulación en finanzas.

La demostración matemática del método Montecarlo se basa en el teorema del límite central y en la ley de los grandes números (Sobol, 2018). El teorema del límite central establece que, a medida que aumentamos el tamaño de la muestra aleatoria, la distribución de las medias de las muestras se aproxima a una distribución normal, independientemente de la distribución original de los datos. La ley de los grandes números afirma que, a medida que aumentamos el número de muestras, el promedio de las muestras se aproxima al valor esperado de la variable.

Para demostrar el método Montecarlo, consideremos el cálculo de una integral definida. Supongamos que queremos calcular el valor de una integral  $\int f(x)dx$  en un intervalo  $[a, b]$ , pero la función  $f(x)$  no tiene una primitiva analítica. Utilizando el método Montecarlo, generamos un conjunto de puntos  $(x, y)$  aleatoriamente dentro del rectángulo  $[a, b] \times [0, M]$ , donde  $M$  es un número máximo conocido para  $f(x)$  en el intervalo  $[a, b]$ . Luego, contamos el número de puntos que caen debajo de la curva  $f(x)$  y calculamos el área total del rectángulo. La aproximación de la integral se obtiene mediante la fórmula:

$$\int f(x)dx \approx n/N * (M * (b - a)) \quad (16)$$

Siendo  $n$  el número de puntos debajo de la curva y  $N$  el número total de puntos. A medida que aumentamos el número de puntos, la aproximación se vuelve más precisa y se acerca al valor real de la integral.

La ventaja clave del método Montecarlo es su flexibilidad para abordar problemas complejos en diversas áreas. La simulación estadística basada en muestras aleatorias permite estimar resultados y analizar sistemas que, de otra manera, serían difíciles o imposibles de abordar mediante métodos analíticos tradicionales. A medida que la capacidad de cómputo aumenta, el método Montecarlo se ha vuelto aún más poderoso y se ha convertido en una herramienta esencial en la resolución de problemas en ciencia e ingeniería (Rubinstein & Kroese, 2016).

En conclusión, el método Montecarlo es una poderosa técnica de simulación estadística que utiliza la generación de muestras aleatorias para estimar resultados y resolver problemas complejos en diversas áreas.

## Capítulo 3. Validación y configuración del simulador

---

### 3.1 Escenario

Es importante destacar que no podemos basar nuestro análisis en un escenario donde los nodos sean desplegados de forma completamente aleatoria, ya que los resultados obtenidos en ese contexto podrían solo ser aplicables a una distribución no necesariamente realista, o muy específica. En este trabajo de tesis se utilizó un escenario tipo *Manhattan grid*, donde los nodos son desplegados de manera uniforme, el cual permite controlar variables como el número de vecinos de un nodo y la distancia entre nodos. Esto permite analizar los resultados de manera más general, teniendo en cuenta las similitudes y diferencias entre estos segmentos de topología.

Este enfoque nos permite obtener conclusiones en un escenario más controlado que se pueden extrapolar a escenarios similares a los encontrados en la realidad, ya que consideramos una variedad de escenarios topológicos que pueden encontrarse en redes inalámbricas ad-hoc. Es importante adaptar nuestros análisis a la complejidad y diversidad de estas redes para obtener resultados significativos y prácticos. Las figuras desde la 14 hasta la 23 muestran los despliegues de nodos que se utilizaron para la evaluación de los diferentes algoritmos.

Estos escenarios están compuestos por un transmisor (color verde)  $n$  nodos que tienen implementado los algoritmos de *broadcast* (color amarillo) y un nodo receptor final (color morado). Estos tienen un radio de transmisión-recepción de 50 metros y un radio de interferencia de 75 metros, en el caso de los escenarios dispersos tienen una distancia entre nodos de 35 metros hacia arriba, abajo y sus laterales y una distancia de  $35 * \sqrt{2}$  con respecto a las diagonales.

En el caso de los escenarios densos la diferencia es que en las diagonales tienen un nodo exactamente a la mitad, o sea, a los laterales, arriba y abajo se mantienen distancias de 35 metros y en las diagonales distancias de  $\frac{35*\sqrt{2}}{2}$  logrando así una mayor densidad de nodos por metro cuadrado.

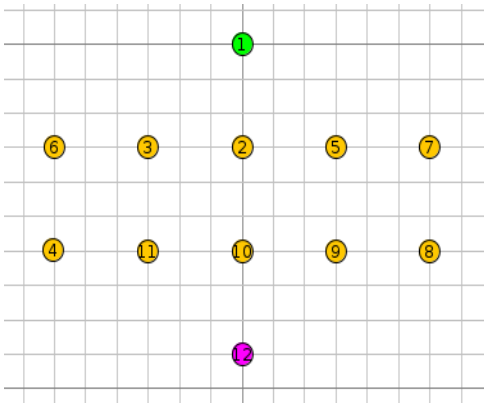


Figura 14. Escenario disperso 10 nodos

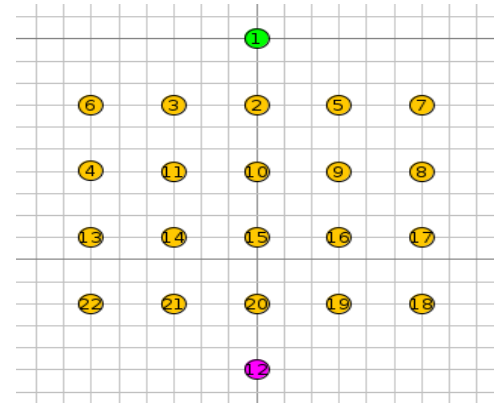


Figura 15. Escenario disperso 20 nodos

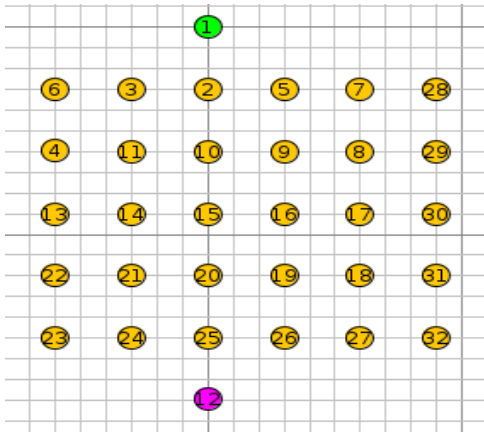


Figura 16. Escenario disperso 30 nodos

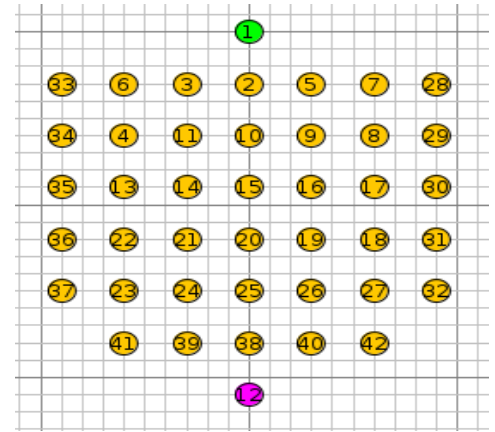


Figura 17. Escenario disperso 40 nodos

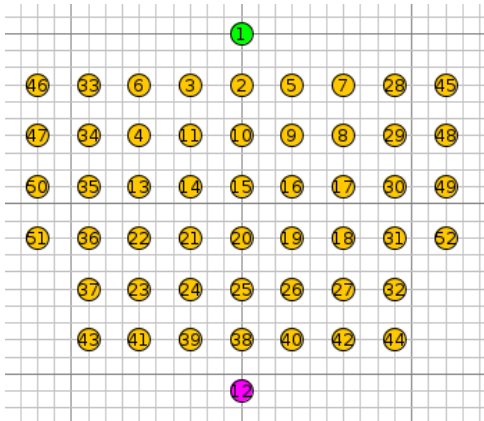


Figura 18. Escenario disperso 50 nodos

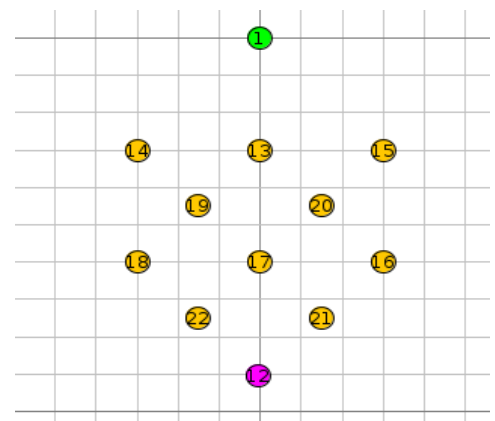


Figura 19. Escenario denso 10 nodos

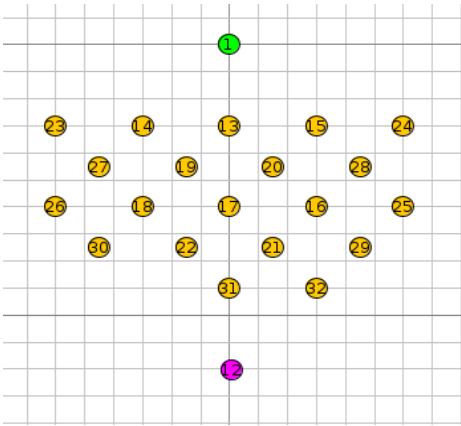


Figura 20. Escenario denso 20 nodos

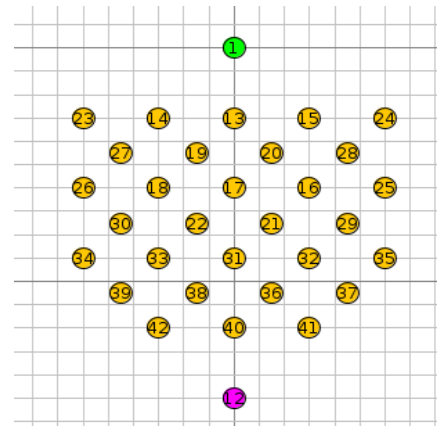


Figura 21. Escenario denso 30 nodos

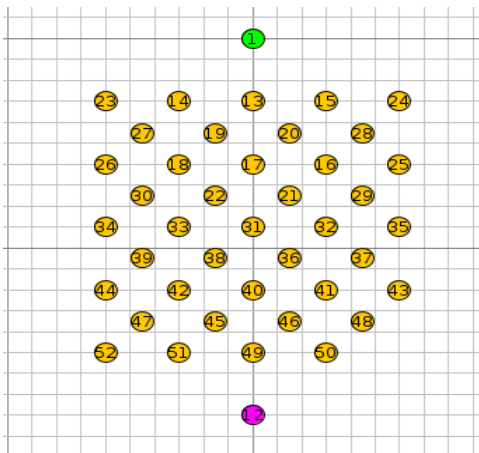


Figura 22. Escenario denso 40 nodos

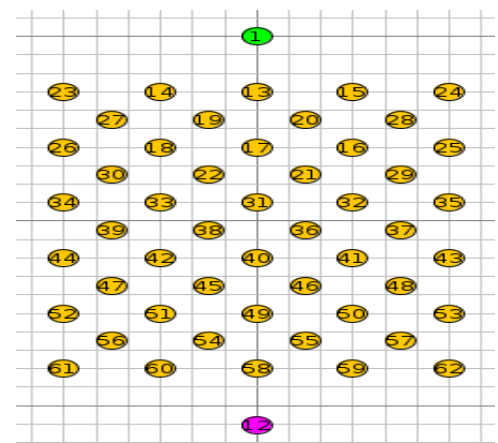


Figura 23. Escenario denso 50 nodos.

### 3.2 Validación de la cama de pruebas

La validación de un simulador, como COOJA, es esencial para garantizar que los resultados obtenidos en entornos virtuales sean coherentes y reflejen con precisión el comportamiento de sistemas en el mundo real. Por ello, en este trabajo de tesis el simulador COOJA se validó por medio de una implementación de software y comparada con los resultados de (Cruz, 2017) y de (Cardenas, 2019).

En (Cruz, 2017) se realizó la validación de la capa IEEE 802.15.4 del simulador OMNET++ en el *framework* enfocado a redes inalámbricas y móviles “MiXiM” (Kodali & Malothu, 2016). Esto mediante la comparativa del modelo de simulación de la capa MAC del protocolo IEEE 802.15.4 con el modelo analítico de (Gaxiola, 2012) y una cama de pruebas experimental.

En (Cardenas, 2019) se realizó la validación del simulador COOJA, replicando el mismo experimento que realizó (Cruz, 2017) en el simulador OMNET++. Esto aplicado a la comparativa del modelo de simulación de la capa MAC en el protocolo 802.15.4. En este trabajo de tesis se replicó el experimento realizado en (Cardenas, 2019) con el fin de corroborar nuevamente la fiabilidad de los datos arrojados por el simulador de redes COOJA.

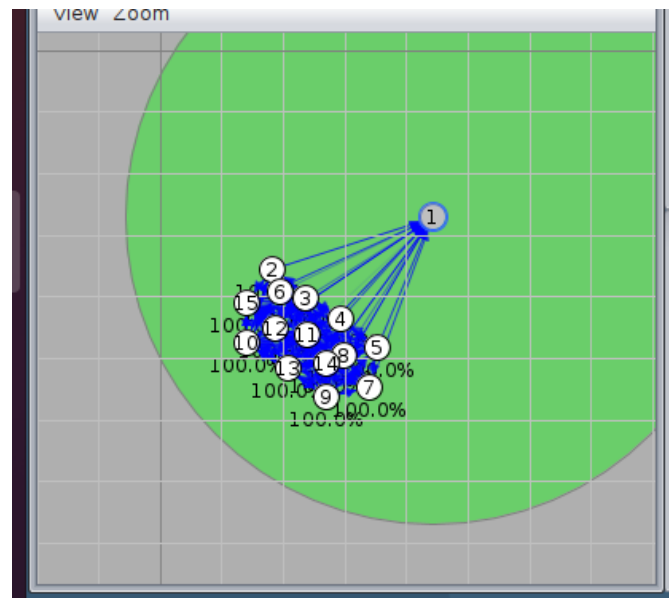
### 3.2.1 Escenario de validación

En la Tabla 1 se muestran las configuraciones hechas a los parámetros de CSMA a implementarse en el simulador COOJA con el fin de replicar los experimentos realizados para la validación del simulador en (Cardenas, 2019; Cruz, 2017).

**Tabla 1.** Parámetros de CSMA/CA no ranurado.

<i>Parámetro</i>	<i>Valor</i>	<i>Descripción</i>
<b><i>aUnitBackoffPeriod</i></b>	20 símbolos	El número de símbolos que forman el periodo de tiempo básico usado por el algoritmo de CSMA/CA.
<b><i>macMaxFrameRetries</i></b>	3	El número máximo de reintentos permitidos después de una transmisión fallida.
<b><i>macMaxCSMABackoffs</i></b>	4	El número máximo de valor desistimiento que el algoritmo de CSMA /CA intenta antes de declarar una transmisión fallida en el canal.
<b><i>macMinBE</i></b>	3	El valor mínimo del exponente de <b><i>back off</i></b> del algoritmo CSMA/CA.
<b><i>macMaxBE</i></b>	5	El valor máximo del exponente <b><i>de back off</i></b> del algoritmo CSMA/CA.
<b><i>aMaxPHYPacketSize</i></b>	127 bytes	Máximo tamaño de la trama que la capa física puede aceptar.

El escenario que se utilizó para la simulación de la capa MAC y validar los resultados obtenidos del simulador de redes COOJA se muestra en la Figura 24. La topología de red que se muestra es una topología de tipo estrella en donde el número de nodos en la red se van variando paulatinamente desde 1 hasta 14 nodos, todos dentro del rango de cobertura de los mismos, garantizando así que todos se escuchen entre ellos. La evaluación consiste en realizar repetidas veces la misma simulación para alcanzar la media estadística de los experimentos, más en concreto se realizarán 10 simulaciones por escenarios. Generando así un total de 140 simulaciones, cada una con un tiempo de duración de 600 segundos para garantizar que todas las simulaciones estén en igualdad de condiciones.



**Figura 24.** Topología de validación de la cama de pruebas.

Los escenarios tienen las siguientes consideraciones:

1. Todos los nodos estarán en el rango de cobertura de los otros nodos, garantizando así que cuando uno de los nodos decida transmitir, los otros nodos detectarán el canal ocupado y tendrán que esperarse para transmitir, evitando así que ocurra el fenómeno del nodo oculto.
2. No existen pérdidas por propagación debido a que los nodos están lo suficientemente cerca del nodo resumidero, tampoco existen interferencias de alguna red externa. Se toma en consideración el modelo de propagación UDGM explicado en el epígrafe 2.2.1.2, porque este considera el modelo de propagación de Friis.
3. La movilidad de los nodos es cero, cada nodo se encuentra en una posición geográfica desde su inicio hasta el final, esta nunca variará ni sufrirá ninguna alteración.

Los parámetros para el entorno de simulación del experimento están definidos en la Tabla 2.

La métrica considerada para la evaluación es la tasa de paquetes perdidos, como se observa en la ecuación 17, definida como el complemento del número de paquetes recibidos en el resumidero entre el número total de paquetes generados por los nodos transmisores.



**Tabla 2.** Parámetros de simulación para la validación del simulador.

<i>Parámetro</i>	<i>Valor</i>
<b><i>Tiempo de simulación</i></b>	600 seg
<b><i>Número de simulaciones por nodo</i></b>	10
<b><i>Número total de simulaciones</i></b>	140
<b><i>Mote utilizado</i></b>	Z1 Zolertia
<b><i>Modelo de propagación</i></b>	UDGM
<b><i>Capa de red</i></b>	Rime Protocol
<b><i>Capa MAC</i></b>	CSMA/CA not Beacon able
<b><i>Capa RDC</i></b>	Nullrdc
<b><i>Capa PHY</i></b>	IEEE 802.15.4
<b><i>Tamaño de Paquete</i></b>	31 y 115 Bytes
<b><i>Tasa de transmisión</i></b>	12 kbps

$$P_l = 1 - \frac{\sum_{j=1}^Q P_{rj}}{\sum_{i=1}^N P_{ti}} \quad (17)$$

Donde:

$P_l$  : Tasa de paquetes perdidos.

$P_{rj}$  : Paquetes recibidos por el j-esimo resumidero.

$P_{ti}$  : Paquetes transmitidos por el i-esimo nodo.

$Q$  : Número de resumideros en la red.

$N$  – Número de nodos transmisores en la red.

El perfil de tráfico utilizado en la comparativa de los resultados de (Cruz, 2017) en OMNET++ y (Cardenas, 2019) en COOJA se muestran en la Tabla 3.

**Tabla 3.** Análisis del empaquetamiento para validación del simulador COOJA.

<i>Tasa de 12kbps</i>	<i>Tamaño de paquete 31 bytes</i>	<i>Tamaño de paquetes de 115 bytes</i>
<b><i>Payload</i></b>	18 bytes	102 bytes
<b><i>Tasa de envío de paquetes</i></b>	83.33 pqte/seg	14.7 pqte/seg
<b><i>Encabezado MAC</i></b>	13 bytes	13 bytes
<b><i>Sobre encabezado MAC</i></b>	8.66 kbps ~ 72%	1.52 kbps ~12%
<b><i>Tráfico total por nodo</i></b>	20.66 kbps	13.524 kbps

Los datos presentados en la Tabla 3 y los resultados ilustrados en la Figura 25 permiten observar claramente que el método de empaquetado grande arroja resultados superiores. En situaciones con una distribución amplia de nodos, como en el caso de una red con 14 usuarios, este método muestra una pérdida de paquetes de aproximadamente el 14%. En contraste, el método de empaquetado pequeño experimenta una pérdida de alrededor del 40% en la misma situación.

El método de empaquetado pequeño muestra un mayor grado de contención del canal debido a la cantidad de paquetes necesarios para satisfacer la tasa de demanda. A medida que el número de usuarios en la red aumenta, esta contención se intensifica, lo que resulta en una probabilidad más alta de que el medio esté ocupado cuando se intenta acceder a él. En consecuencia, se produce un mayor número de paquetes descartados al superar el límite de intentos establecido por la capa MAC, como se ilustra en la Figura 25, específicamente para el escenario de empaquetamiento de 31 Bytes.

Este fenómeno revela que el método de empaquetado pequeño es menos eficiente en situaciones de alta densidad de usuarios, ya que la cantidad de paquetes intercambiados para satisfacer la demanda de datos resulta en una mayor congestión del canal y, como resultado, un aumento en la pérdida de paquetes debido a la superación del límite de intentos establecido por la capa MAC.

La comparación de resultados entre los simuladores muestra una tendencia consistente. A medida que el número de nodos en la red aumenta, la tasa de pérdida de paquetes también se incrementa debido a la mayor contención del canal generada por la congestión. En el caso de empaquetamiento grande (115 Bytes), la tasa de pérdida de paquetes permanece bastante constante con pequeñas variaciones a medida que se incrementa el número de usuarios. Las fluctuaciones que existe entre las gráficas de (Cardenas, 2019) y las llevadas a cabo en este trabajo de investigación son mínimas (alrededor de un 2%) las cuales

corroboran la validación de el simulador COOJA. Para eliminar esta pequeña variación bastara con realizar una mayor cantidad de simulaciones, hasta lograr que los resultados se igualen.

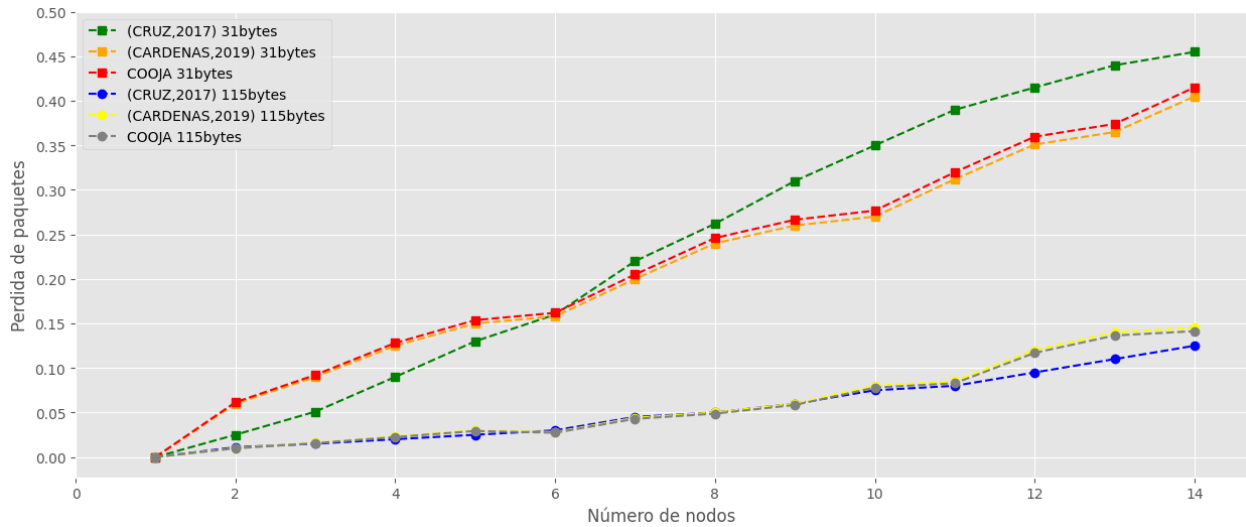


Figura 25. Comparativa de pérdida de paquetes promedio entre los simuladores OMNET++ y COOJA

### 3.3 Simulación

En esta sección se describirá a detalle todas las configuraciones que se tienen en el sistema operativo Contiki tanto en la capa física, la capa MAC, la capa de RDC y como se establecieron las simulaciones para validar los resultados. Todo con el fin de que este trabajo pueda ser replicado en futuras investigaciones.

#### 3.3.1 Configuración de la capa Física

En la Tabla 4 se describen los parámetros más relevantes de la configuración de la capa física en Contiki, tales como los módulos de radio que usarán los nodos, el protocolo de comunicación que se usará en las simulaciones, así como el mínimo y el máximo tamaño que podrán tener los paquetes.

Tabla 4. Parámetros de la Capa física que se utilizarán en la simulación.

Parámetro	Valor	Descripción
<b>NETSTACK_CONF_RADIO</b>	cc2420_driver	Módulo de radio usado por los nodos
<b>NETSTACK_CONF_FRAME</b>	framer_802154	Protocolo de capa física
<b>SHORTEST_PACKET_SIZE</b>	43	Mínimo tamaño de los paquetes
<b>SICSLWPAN_CONF_MAC_MAX_PAYLOAD</b>	127	Tamaño máximo del Payload

### 3.3.2 Configuración de la capa MAC

En la Tabla 5 se describen los parámetros principales de la configuración de la capa MAC en el sistema operativo Contiki. Estos parámetros son el exponente inicial y el exponente máximo que tendrá el algoritmo de CSMA/CA el cual se describió con más detalles en la subsección 2.1.2.1, otro parámetro que se describe son el máximo número de reintentos antes de considerar el canal ocupado y el número máximo de intentos de retransmisiones.

**Tabla 5.** Parámetros de la capa MAC que se utilizarán en la simulación.

<i>Parámetro</i>	<i>Valor</i>	<i>Descripción</i>
<b>CSMA_MIN_BE</b>	0	Inicial exponente de backoff
<b>CSMA_MAX_BE</b>	4	Máximo exponente de backoff
<b>CSMA_MAX_BACKOFF</b>	5	Máximo número de reintentos antes de considerar el canal ocupado
<b>CSMA_MAX_MAX_FRAME_RETRIES</b>	7	Máximo número de intentos de retransmisiones

### 3.3.3 Configuración de la capa RDC

En la Tabla 6 se da una descripción de los principales parámetros para la capa encargada de controlar el ciclo de trabajo en Contiki. Todos estos parámetros son necesario para que el simulador use la capa de RDC tal como se describió en el epígrafe 2.3.

**Tabla 6.** Parámetros de la capa RDC que se utilizarán en la simulación.

<i>Parámetro</i>	<i>Valor</i>	<i>Descripción</i>
<b>WITH_FAST_SLEEP</b>	1	Activar el <i>fast sleep</i> después de detectar el canal ocupado con otro tráfico
<b>RDC_CONF_HARDWARE_ACK</b>	0	La radio devuelve TX_OK o TX_NOACK después de la espera de confirmación automática
<b>RDC_CONF_MCU_SLEEP</b>	0	MCU puede dormir durante el periodo que el radio esta off.
<b>CCA_COUNT_MAX</b>	2	Número de CCA a realizar para cada verificación periódica del canal.
<b>CCA_COUNT_MAX_TX</b>	6	Número de veces consecutivas que se lanzan CCA para checar el canal antes de iniciar una TX.
$T_{rdc}$	$1/f_{rdc}$	Periodo de activación de los módulos de radio

El parámetro de  $f_{rdc}$  se varia desde 8hz hasta 64hz con el fin de evaluar y comparar los resultados que se encuentran para varios ciclos de trabajo.

### 3.3.4 Proceso de simulación

Con el propósito de obtener un análisis estadísticamente robusto y evitar asumir la definitividad de los resultados de las simulaciones, se llevaron a cabo múltiples repeticiones de los experimentos, variando aleatoriamente la semilla. La variación de esta semilla es esencial debido a que la generación de números aleatorios en los procesos computacionales no es completamente aleatoria, sino que sigue un patrón pseudoaleatorio determinado por una serie con parámetros de entrada, siendo la semilla uno de estos parámetros. Al modificar la semilla, se altera la generación de números pseudoaleatorios, introduciendo variabilidad en los resultados de las simulaciones. Es por ello que, para un mismo escenario, con una misma frecuencia de ciclo de trabajo se realizan treinta simulaciones variando la semilla de generación de números aleatorios. Luego de obtener estas simulaciones, se hallaba las métricas anteriormente mencionadas en el epígrafe 2.5 para cada simulación y luego se promediaban para así obtener un resultado definitivo como se muestra en la ecuación 18.

$$Metrica_{ave} = \frac{\sum_{i=1}^{sim} Metrica_i}{sim} \quad (18)$$

Donde:

**$Metrica_{ave}$**  : Es el promedio de la métrica a analizar (*Reachability, Delay, SRB*).

**$Metrica_i$**  : Es la métrica por simulación (*Reachability, Delay, SRB*).

**$sim$**  : Es el total de simulaciones realizadas, en este caso 30.

Por cada simulación se tiene un escenario como los mostrados en el epígrafe 3.1 donde se cuenta con un nodo que es el transmisor, en cualquier de los casos siempre es el nodo 1 donde este lanza cien paquetes, todos de igual tamaño, solamente se diferenciaban entre ellos por un identificador que se va incrementando según se lanzan a la red. Estos paquetes se lanzan a la red por parte del nodo transmisor a razón de cuatro paquetes por segundo, después de que el nodo transmisor lance todos sus paquetes se considera la simulación terminada, dejando siempre un tiempo de guarda para que los otros nodos de la

red tengan tiempo de procesar la información y actuar según sus algoritmos de broadcast, y se guarda la información de los nodos en formato **.txt** proveniente de la herramienta Mote Output de COOJA.

En la Figura 26 podemos observar la estructura de la información extraída del simulador COOJA donde se encuentra los tiempos, tanto del transmisor (ID:1) como el resto de los nodos, así como los nodos que reciben los paquetes emitidos por el transmisor. Luego la simulación es reiniciada con una nueva semilla para volver a realizar exactamente el mismo experimento durante treinta veces, variando siempre la semilla de generación de números pseudoaleatorios. Después de terminada las treinta simulaciones se procede a cambiar la frecuencia del ciclo de trabajo y se vuelve a realizar el mismo proceso anteriormente descrito. Luego de iterar sobre los ciclos de trabajo se procede a iterar esta vez sobre los distintos escenarios para así completar todas las simulaciones realizadas para un parámetro de uno de los algoritmos de **broadcast** seleccionados para su implementación.

```
00:01.711 ID:1 Sending broadcast: 0.CUBA.0
00:01.754 ID:2 Data received 0.CUBA.0
00:01.754 ID:3 Data received 0.CUBA.0
00:01.754 ID:5 Data received 0.CUBA.0
00:01.814 ID:2 Data received 1.CUBA.0
00:01.865 ID:10 Data received 1.CUBA.0
00:01.865 ID:8 Data received 1.CUBA.0
00:01.868 ID:9 Data received 1.CUBA.0
00:01.899 ID:16 Data received 2.CUBA.0
00:01.899 ID:7 Data received 2.CUBA.0
00:01.928 ID:8 Data received 2.CUBA.0
00:01.942 ID:5 Data received 2.CUBA.0
00:01.949 ID:17 Data received 2.CUBA.0
00:01.961 ID:1 Sending broadcast: 0.CUBA.1
00:01.992 ID:8 Data received 3.CUBA.0
00:02.003 ID:3 Data received 0.CUBA.1
```

**Figura 26.** Ejemplo de datos mostrados por la herramienta **Mote Output** de COOJA

**Tabla 7.** Estructura de una línea de la herramienta **Mote Output**.

<i>Tiempo</i>	<i>Identificador</i>	<i>Paquete</i>
00:00.000	ID: #	0.CUBA.0
Min:seg.mseg	Nodo	#_salto . información . identificador

En la Tabla 7 se describe más a detalle la estructura de una de las líneas procedentes del Mote Output de COOJA, tales como la estructura que tiene el tiempo mostrado, el ID de cada nodo que corresponde con su ID en la red de la simulación y la estructura que tiene los paquetes lanzados a la red.

## Capítulo 4. Resultados y discusión

La metodología seguida para el experimento en este trabajo de investigación fue que el nodo transmisor envía 100 paquetes por simulación, el cual se repitió 30 veces cambiando la semilla de simulación, dando un total de 3000 paquetes por cada tupla (parámetro\_algoritmo; Frecuencia de ciclo de trabajo; Densidad de vecindario) dando un total de 15,000 simulaciones con el fin de encontrar la media estadística para así garantizar la fiabilidad de los resultados. En la Tabla 8 se describe con mejor detalle los parámetros variados para cada algoritmo junto con la cantidad de veces que se realizaron dichas simulaciones variándoles sus semillas para así garantizar que no se repitiera exactamente el mismo experimento.

En la siguiente **¡Error! No se encuentra el origen de la referencia.** reflejará de mejor manera el trabajo realizado durante el desarrollo de esta tesis donde se muestran los algoritmos seleccionados para su evaluación, los ciclos de trabajo empleados, los escenarios en los cuales se evaluaron y los parámetros variados de cada algoritmo en particular, dando como resultado un total de 15000 simulaciones realizadas con las cuales se mostrará más adelante los resultados.

**Tabla 8.** Valores utilizados para los parámetros a variar dentro del escenario de simulación.

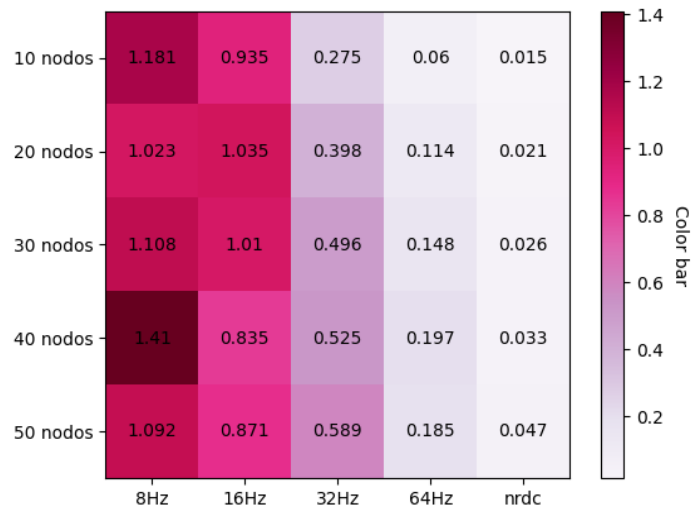
Algoritmos	Parámetros	Ciclo Trabajo (Hz)	Densidades (nodos)	Tx/sim	Simulaciones
<b>GOSSIP1</b>	P = [0.6;0.8;1]	8-16-32-64-nrdc	10-20-30- 40-50	100	30
<b>DAPF</b>	K = [2;3;5]	8-16-32-64-nrdc	10-20-30- 40-50	100	30
<b>Counter- Based Scheme</b>	RAD = [0.001;0.01] Cth = [2;3]	8-16-32-64-nrdc	10-20-30- 40-50	100	30
				Total:	15,000

### 4.1 Análisis de resultados para algoritmo GOSSIP1(p).

En esta sección, realizamos un análisis de los resultados obtenidos para el algoritmo GOSSIP1(p), el cual se analizó su estructura y funcionamiento en la sección 2.4.1.1. Para presentar de manera más concisa la gran cantidad de información generada por nuestras simulaciones, nos centraremos en los resultados



obtenidos para un valor específico de  $p$ , en concreto,  $p=0.8$ , y en el contexto de un escenario disperso. Si desea explorar los resultados de todas las demás configuraciones evaluadas durante este estudio, le invitamos a consultar el Anexo A.



**Figura 27.** Resultados obtenidos del algoritmo GOSSIP( $p$ ), en donde el **Delay**  $p = 0.8$ , el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.

En la Figura 27 se presentan los resultados correspondientes al algoritmo GOSSIP1( $p$ ) en la métrica de desempeño del **Delay**. Para representar estos datos, utilizamos el eje X para mostrar la variación de los ciclos de trabajo desde 8Hz hasta nrhc (sin ciclo de trabajo), y en el eje Y, representamos las densidades de nodos, que varían desde 10 hasta 50 nodos en los escenarios. Observamos una tendencia clara en la métrica que a medida que aumenta el ciclo de trabajo el **Delay** tiende a disminuir, lo que indica tiempos de transmisión más cortos. A medida que la frecuencia del ciclo de trabajo aumenta, los nodos tienen más oportunidades de transmitir sin interferencias. Esto conduce a una disminución del retardo, ya que los paquetes se retransmiten con mayor rapidez cuando el canal está libre. Este comportamiento se puede explicar considerando los conceptos analizados en el epígrafe 2.5, que profundizaremos a continuación:

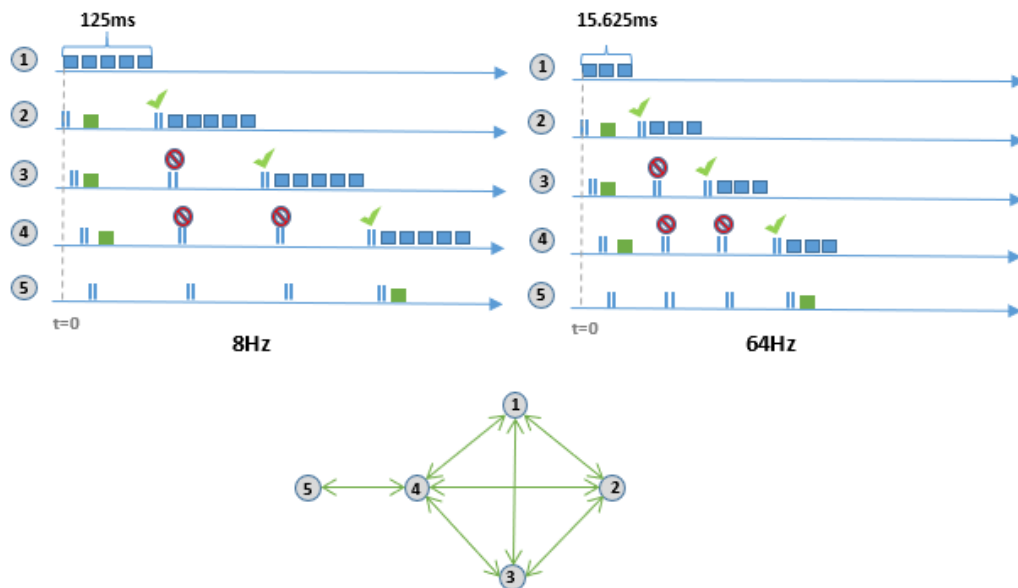
**Delay:** Como podemos observar en la Figura 27 a medida que aumenta la frecuencia del ciclo de trabajo el **Delay** disminuye. Esto se debe a que cuando un nodo opta por retransmitir un paquete basándose en su algoritmo de **broadcast**, debe realizar el sensado del medio de transmisión. Si el medio está ocupado, el nodo deberá regresar a un estado de inactividad hasta que, de acuerdo con su configuración de la frecuencia del ciclo de trabajo, se le permita efectuar la transmisión. Es en este punto donde observamos

un aumento en las demoras, que están directamente relacionadas con la frecuencia de los ciclos de trabajo. Estos intervalos de actividad de los nodos varían según la frecuencia de los ciclos de trabajo y se describen más detalladamente en la Tabla 9.

**Tabla 9.** Descripción de los tiempos de acuerdo a la frecuencia del ciclo de trabajo.

Ciclo de trabajo	Tiempo
8Hz	125 ms
16Hz	62.5 ms
32Hz	31.25 ms
64Hz	15.625 ms

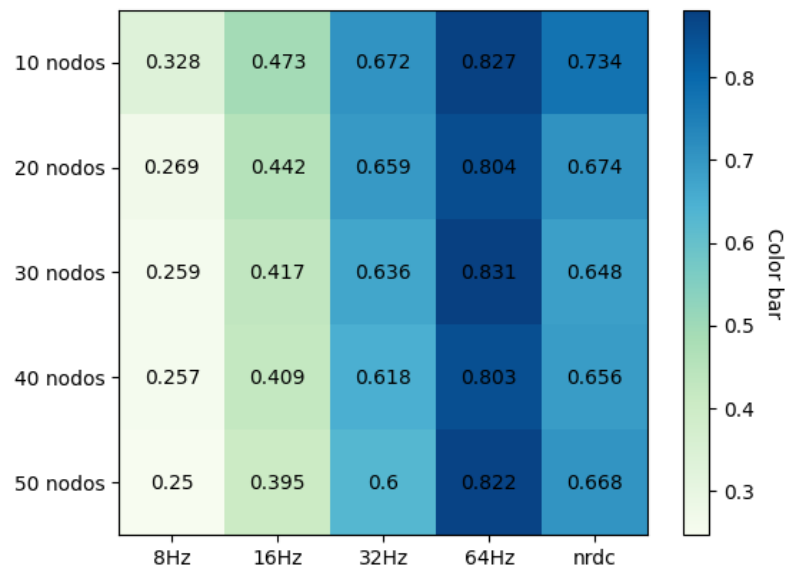
En la Figura 28, podemos observar una representación gráfica más detallada de lo que sucede con los retardos en nuestras simulaciones, respaldando los datos presentados en la Tabla 9. En la Figura 28, presentamos dos ejemplos de representaciones de líneas de tiempo para dos frecuencias de ciclos de trabajo diferentes (8Hz y 64Hz), con cinco nodos en total. En esta configuración, los nodos 1, 2, 3 y 4 están dentro del rango de transmisión directa entre sí, lo que significa que pueden comunicarse directamente, mientras que el nodo 5 solo puede ser alcanzado a través del nodo 4. Los paquetes azules representan una transmisión continua durante un periodo completo (125ms para 8Hz y 15.625ms para 64Hz), como se explicó en la sección 2.3. Los paquetes verdes representan una recepción exitosa por parte de los nodos receptores.



**Figura 28.** Ejemplo *Time Delay*

En la Figura 28, se ilustra la situación en la que el nodo 1 inicia la transmisión de un paquete en  $t = 0$  y continúa transmitiendo el mismo paquete durante 125ms (o 15.625ms, dependiendo de la frecuencia del ciclo de trabajo). Durante este tiempo, los nodos 2, 3 y 4 monitorean el canal utilizando los CCA y reciben el paquete de manera satisfactoria. Los tres nodos, según la configuración del algoritmo de **broadcast**, deciden retransmitir el paquete. El nodo 2 inicia la retransmisión durante el mismo período de 125ms (o 15.625ms), lo que hace que los nodos 3 y 4 encuentren el canal ocupado y, por lo tanto, esperen para realizar su retransmisión cuando sea posible. El nodo 3 retransmitirá en el próximo período de tiempo, y el nodo 4 lo hará después de que el nodo 3 haya terminado su retransmisión, ya que este detectó el canal ocupado por el nodo 3. Como resultado, el nodo 5 solo recibirá el paquete después de que el nodo 4 haya iniciado su retransmisión, lo que genera un retardo mínimo adicional de 375ms (o 46.875ms, dependiendo de la frecuencia del ciclo de trabajo).

Este ejemplo demuestra claramente el impacto de la frecuencia del ciclo de trabajo en el **Delay** y cómo los ciclos de trabajo con más baja frecuencia pueden generar mayores demoras debido al tiempo de las retransmisiones y la ocupación del canal por dicho nodo. Este análisis respalda la tendencia que existe entre la frecuencia del ciclo de trabajo y la disminución del **Delay** según esta frecuencia vaya aumentando.



**Figura 29.** Resultados obtenidos del algoritmo GOSSIP( $p$ ), en donde la **Reachability**  $p = 0.8$ , el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado.

**Reachability:** Así como observamos en la métrica de **Delay**, podemos notar en la Figura 29 que a medida que la frecuencia del ciclo de trabajo aumenta, mejora el desempeño de la métrica de **Reachability**. Esta

tendencia se debe a un caso de ocurrencia interesante. Cuando un nodo decide retransmitir un paquete y el canal de comunicación está libre, ese nodo transmitirá exactamente el mismo paquete durante un período completo entre los períodos de tiempo en los que se realizan los CCA, como se ilustra en la Figura 7 y se detalla en la Tabla 9. En otras palabras, cuanto menor sea la frecuencia del ciclo de trabajo (por ejemplo, 8Hz con una transmisión continua de 125ms), más tiempo ocupará el canal un nodo que decida retransmitir.

Este comportamiento puede afectar negativamente la métrica de **Reachability** debido a cómo funciona el mecanismo de ciclo de trabajo, como se explicó en la sección 2.3.2. Cuando el mecanismo de ciclo de trabajo detecta un CCA positivo, no puede diferenciar si ese CCA se debe a una señal de radio válida o a interferencia. En este caso, entra en acción el mecanismo de detección de paquetes y el modo de ahorro de energía conocido como **fast sleep**. Una de las condiciones para activar el **fast sleep** es si la CCA detecta actividad de radio, pero la duración de esa actividad es mayor que la longitud máxima de un paquete, denotada como  $t_l$ . Esto indica que la CCA ha detectado ruido en el canal y permite que el nodo vuelva a dormirse solo si el período de actividad no se sigue de un período de silencio. Este caso es particularmente común cuando, como se muestra en la Figura 30 tenemos una sección de red compuesta por 5 nodos, donde el nodo 1,2,4,5 no pueden comunicarse entre ellos, pero si se pueden comunicar con el nodo 3.

Esto provoca que cuando cualquiera de los nodos 1,2,4,5 quieran transmitir, estos checaran el medio y lo encontraran disponibles ya que no se interfieren entre ellos, produciéndose así dicha transmisión. Pero para el nodo 3 será algo más complejo debido a que como esta en el rango de escucha de los nodos 1,2,4,5 a la hora de checar el canal CCA, mientras más bajo sea la frecuencia del ciclo de trabajo más probabilidades hay de toparse con más de una transmisión de nodos distintos que se solapen en tiempo, generando así una de las condiciones que se trataron en el epígrafe 2.3.2 para el **fast sleep**. Al ocurrir esta situación donde se activó el mecanismo de **fast sleep** el nodo receptor descarta este paquete debido a la imposibilidad de recibirlo, degradando así la métrica de la **Reachability**. Como se puede observar en la Figura 30 mientras más baja sea la frecuencia del ciclo de trabajo es más propenso a que ocurra este fenómeno, al contrario del otro caso que se observa, donde mientras más alta sea la frecuencia del ciclo de trabajo, el tiempo de ocupación del canal de un nodo para transmisión es más corto, como consecuencia las probabilidades de solapamiento de transmisiones de los nodos es más pequeña y por lo tanto las probabilidades de recepción satisfactoria de un paquete aumentan. Para una revisión completa de los resultados obtenidos, refiérase el lector al Anexo B.

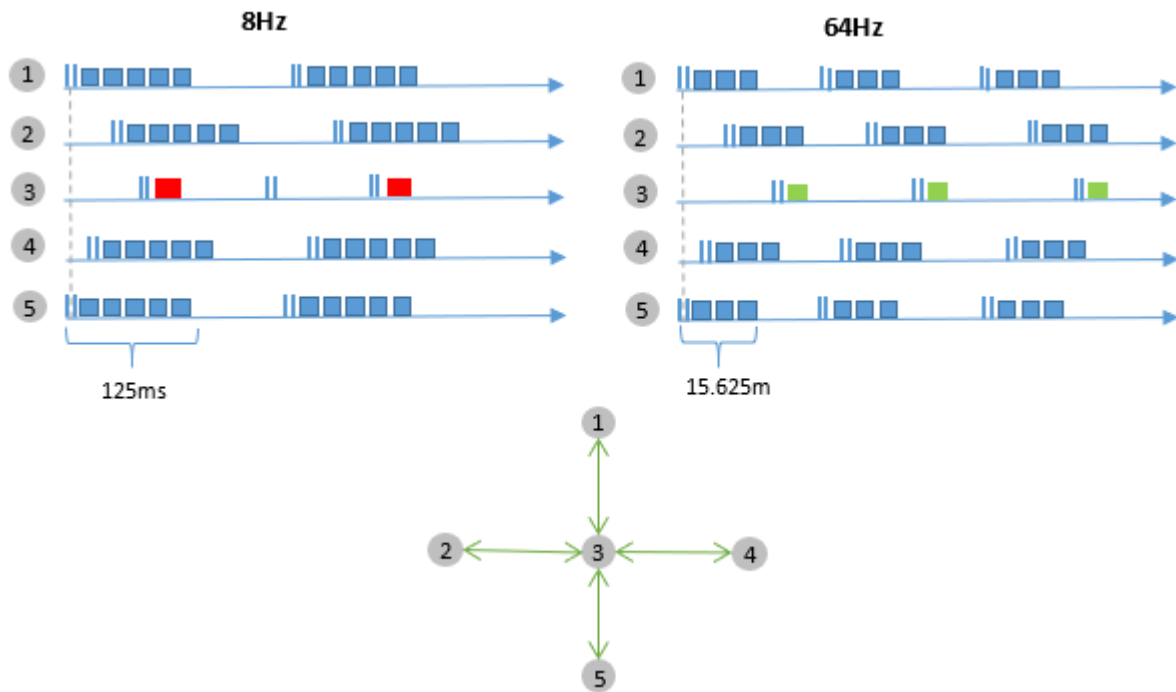


Figura 30. Ejemplo *Timeline Reachability*

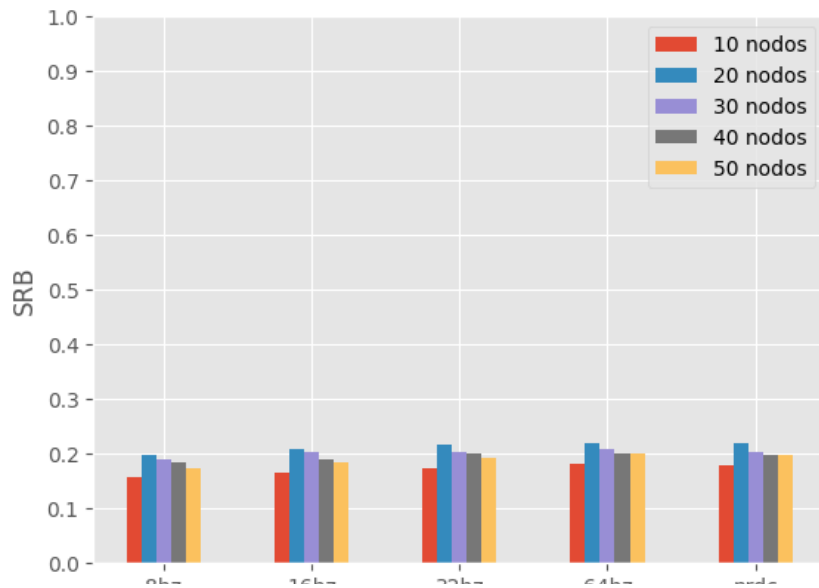


Figura 31. Resultado de la métrica *Saved Rebroadcast* para el esquema GOSSIP1(p).

**Saved Rebroadcast (SRB):** Como se observa en la Figura 31 la métrica de *Saved Rebroadcast* no se ve afectada con los cambios de la frecuencia de los ciclos de trabajo. Esto se debe a que este algoritmo de

**broadcast** basa su esquema en probabilidad, por lo tanto, las retransmisiones se rigen por el algoritmo seleccionado y no por cómo se comportan las retransmisiones en redes con ciclos de trabajo, contrario a lo que pasa en las otras métricas analizadas (**Delay** y **Reachability**).

En este caso la probabilidad de retransmisión como se mencionó en el inicio de este epígrafe 4.1 para la muestra de los resultados obtenidos se tomó  $p=0.8$ , es por eso que se observa en la Figura 31 como independientemente de la frecuencia del ciclo de trabajo y la densidad de la red, los resultados muestran que la SRB se acerca siempre a 0.2 debido a que la probabilidad de retransmisión es de 0.8 y la SRB es el complemento de las retransmisiones de cada nodo. Para una revisión completa de los resultados obtenidos, refiérase el lector al Anexo C.

## 4.2 Análisis de resultados para algoritmo DAPF

En esta sección, se muestra un análisis detallado de los resultados obtenidos con los parámetros de la Tabla 8 relacionadas con el algoritmo DAPF.

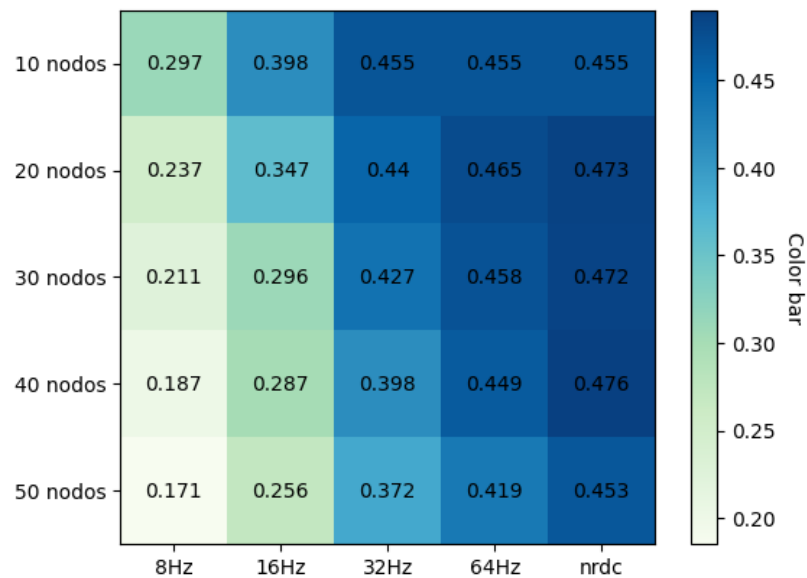


**Figura 32.** Resultados obtenidos del algoritmo DAPF, en donde **Delay**  $k=3$ , el eje X se refiere a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.

En la Figura 32 correspondiente al **Delay** muestra las variaciones obtenidas con DAPF con  $k=3$ , conforme aumenta la frecuencia del ciclo de trabajo y la densidad de nodos. Como se muestra en la Tabla 8, la densidad de nodos se varía desde 10 hasta 50 nodos, y el ciclo de trabajo desde 8Hz hasta 64Hz y se incluye la evaluación sin la capa de ciclo de trabajo. Podemos observar que a medida que la frecuencia del ciclo

de trabajo aumenta tenemos mejor desempeño de dicha métrica, esto se debe en gran medida por lo que anteriormente se analizó en el epígrafe 2.5 como analizaremos con mayor detalle a continuación:

**Delay:** Cuando un nodo decide por el algoritmo de **broadcast** retransmitir un paquete este debe analizar el medio de transmisión, si el medio se encuentra ocupado, el nodo debe volver a un estado de inactividad hasta que, según su configuración, se le permita reanudar la transmisión y es aquí cuando se incrementan las demoras en función del ciclo de trabajo, siendo estos intervalos de actividad distintos para cada ciclo de trabajo y se describieron en la Tabla 9. Visualmente puede ver los intervalos de análisis de CCA, que es el tiempo que debe transcurrir para volver a checar el medio de transmisión, en la Figura 7. Para una revisión completa de los resultados obtenidos, refiérase el lector al Anexo D.

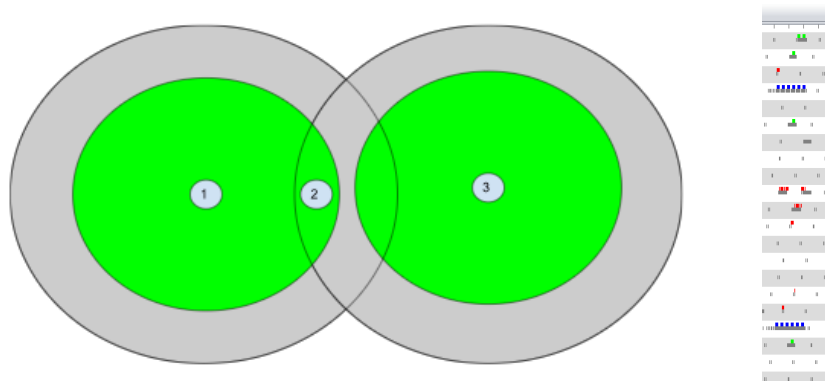


**Figura 33.** Resultados obtenidos del algoritmo DAPF, en donde la **Reachability**  $k = 3$ , el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado.

**Reachability:** Al igual que para el **Delay** podemos observar en la Figura 33 que a medida que aumenta el ciclo de trabajo mejora el desempeño de esta métrica. Esto se debe a que cuando un nodo decide retransmitir un paquete y el canal esta libre para hacerlo, este transmitirá exactamente el mismo paquete durante un periodo completo entre los CCA como se muestra en la Figura 7 y en la Tabla 9. En otras palabras, mientras más bajo el ciclo de trabajo (8Hz-125ms de transmisión continua), más tiempo estará ocupando el canal un nodo que decida retransmitir. Como consecuencia esto degradará la **Reachability** debido a que como se explicó en el epígrafe 2.3.2, el mecanismo del ciclo de trabajo cuando detecta un

CCA positivo, este no es capaz de diferenciar si es una señal de radio o interferencia, entrando a actuar el mecanismo de detección de paquetes y el **fast sleep** para la optimización de consumo de energía. Siendo una de las condiciones para que se active el **fast sleep** si la CCA detecta actividad de radio, pero la duración de la actividad de radio es mayor que la longitud máxima del paquete  $t_l$ , significa que la CCA ha detectado ruido y puede volver a dormir, es decir, si el período de actividad no es seguido por un período de silencio.

Este caso tiene una mayor probabilidad de ocurrencia cuando, como se muestra en la Figura 34a un nodo (en este caso el dos) se encuentra en el rango de transmisión del nodo 1 y en el rango de interferencia del nodo 3, si estos dos nodos (1 y 3) deciden retransmitir un paquete en el mismo intervalo de tiempo o en intervalos que sean capaces de solaparse uno con el otro, el nodo 2 al despertarse a checar el canal se encontrará con el caso que las transmisiones serán más largas que la duración de un paquete, como consecuencia activará el **fast sleep** y por ende esa información no llegará a él, degradando así la métrica de la **Reachability** como se ve más claro en la Figura 34b. Donde se muestra un fragmento de la captura del timeline de una simulación en COOJA, donde el azul representa una transmisión de paquetes de un nodo, el verde la recepción de un paquete por parte de un nodo y el rojo es cuando un nodo es incapaz de recibir un paquete porque lo considera interferencia (Thomson et al., 2016) y se activa el **fast sleep**. Para una revisión completa de los resultados obtenidos, refiérase el lector al Anexo E.

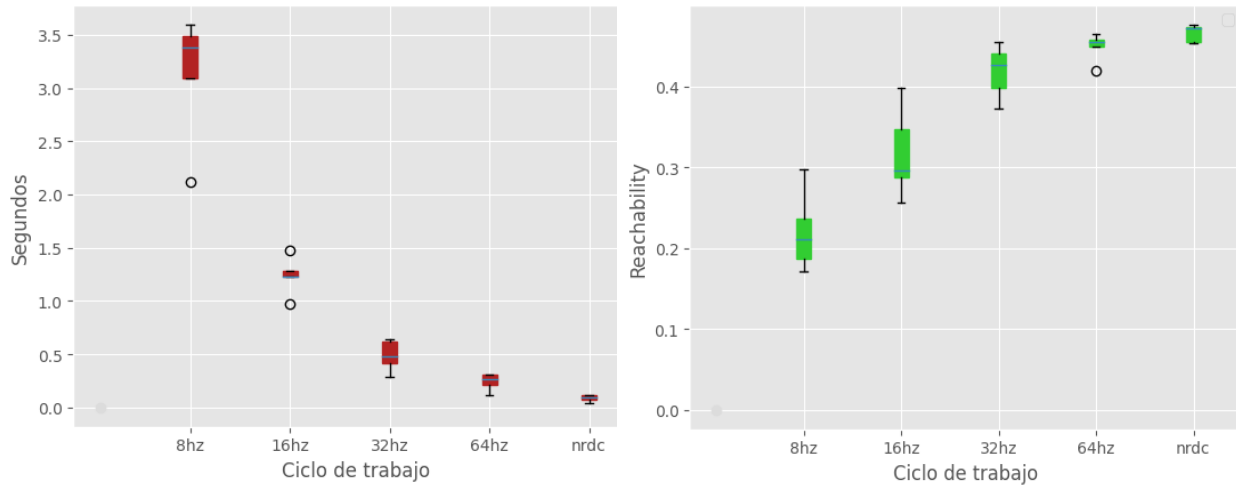


**Figura 34.** a-) Rango de Tx-Rx de los nodos. b-) Captura Timeline COOJA.

Otro aspecto a tener en cuenta en el análisis de las Figura 32 y Figura 33 es que a medida que aumenta el ciclo de trabajo prácticamente se vuelve irrelevante la densidad de los nodos en la red, esto podemos observarlo en dichas figuras donde notamos que a medida que aumenta el ciclo de trabajo la variación de colores para las distintas densidades se hace muy uniforme lo cual nos indica que las variaciones son



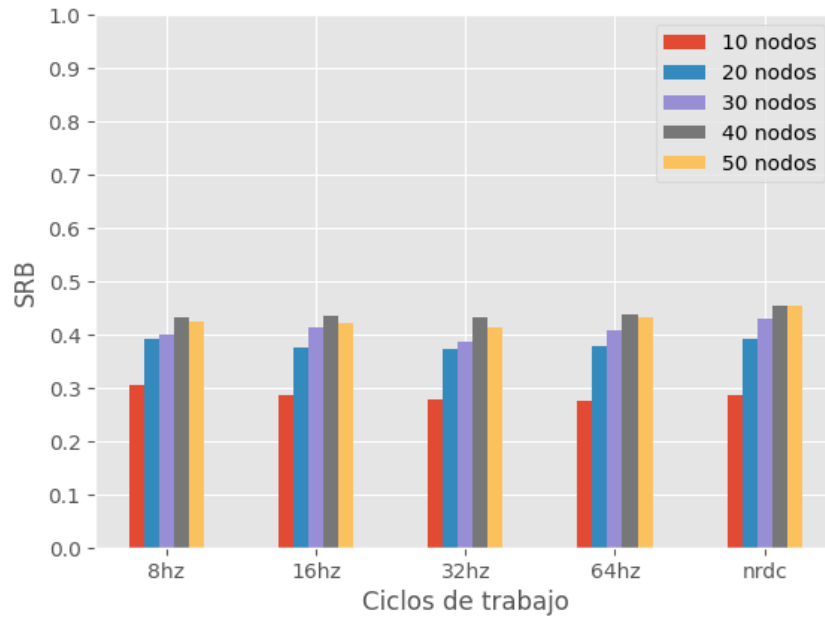
mínimas. Para lograr una mejor visualización puede observar como en la Figura 35 a medida que aumenta el ciclo de trabajo las cajas del grafico tienden a ser más chicas y a no tener **outlayers** lo cual se puede expresar como que los cambios entre redes con distinta densidad son más pequeños y por ende prácticamente despreciables a medida que aumenta la frecuencia del ciclo de trabajo.



**Figura 35.** a-) Gráfico de bigotes **Delay**. b-) Gráfico de bigotes **Reachability**.

**Saved Rebroadcast (SRB):** En la Figura 36 se muestran los resultados de las retransmisiones evitadas por los nodos. Recordando siempre que tanto para esta métrica como para las anteriores se promediaron los resultados de las distintas simulaciones. En el caso de la SRB podemos observar que no hay impacto significativo por la variación del ciclo de trabajo. Esto se puede explicar porque al ser este algoritmo un esquema de probabilidad adaptativa, su probabilidad de retransmisión se basa en la cantidad de vecinos con la que cuente cada nodo, esta cantidad de vecinos no varía con la variación del ciclo de trabajo y eso se ve reflejado en la Figura 36 en cuestión. Contrario con lo que paso en las métricas anteriormente analizadas (**Reachability-Delay**), acá si podemos observar como la densidad de los nodos en la red influye en el comportamiento de la SRB.

Esto pasa porque como se plasmó en la ecuación 3 la probabilidad disminuye a medida que aumenta el número de vecinos de un nodo, el factor  $k$  es el mismo para todos los nodos y pudiera considerarse una constante, si la probabilidad de retransmisión disminuye aumenta la SRB como se ve reflejado en la Figura 36.



**Figura 36.** Resultado de la métrica *Saved Rebroadcast* para el esquema DAPF.

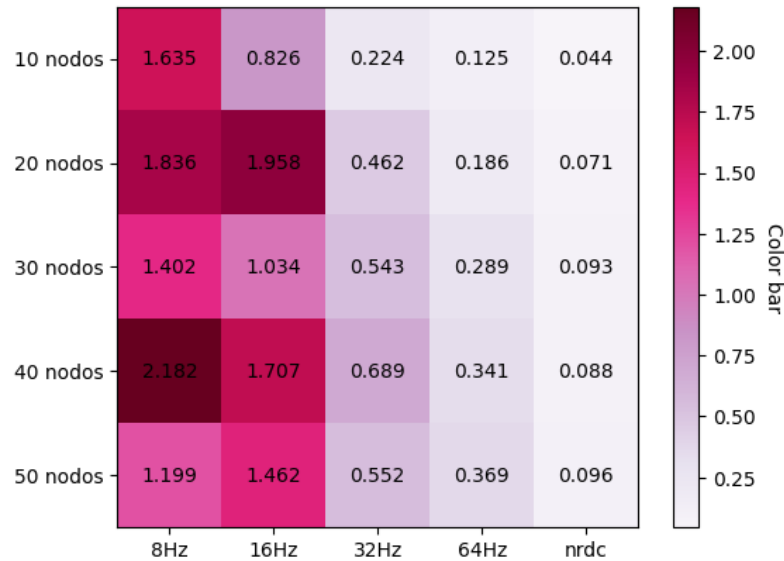
Nota: Las figuras que se mostraron anteriormente son el resultado de la evaluación del algoritmo de broadcast DAPF según la Tabla 8. Solamente se mostraron los resultados obtenidos para los escenarios dispersos como los de las Figura 14, Figura 15, Figura 16, Figura 17, Figura 18 y para  $k = 3$ . Para una revisión completa de los resultados obtenidos para otros valores de  $k$  y para otros escenarios, refiérase el lector al Anexo F.

### 4.3 Análisis de resultados para algoritmo Counter-Based Scheme.

En esta sección, se mostrarán y analizarán los resultados obtenidos para el algoritmo de *broadcast* donde al igual que para los otros algoritmos analizados se variaron los parámetros tal como se muestra en la Tabla 8. En esta sección solo se mostrarán los resultados para  $C_{th}=2$  y  $RAD = 0.001$ . Para una revisión completa de los resultados obtenidos para otros valores de  $k$  y para otros escenarios, refiérase el lector a los Anexo F Anexos G, H, I.

**Delay:** En la Figura 37 podemos observar los resultados obtenidos de la métrica de *Delay* para el algoritmo de *broadcast* CBS, donde al igual que en los algoritmos previamente analizados, se mantuvo la misma tendencia. El *Delay* disminuye conforme la frecuencia de los ciclos de trabajo aumenta, esto se debe en gran medida a como ya se explicó anteriormente en los epígrafes 4.1 y 0, cuando se detalló el *Delay* en

dichos algoritmos, las demoras mínimas a las que un nodo accede al medio de transmisión está regida por la frecuencia de ciclo de trabajo que tiene configurado el mismo. Mientras más baja es la frecuencia del ciclo de trabajo mayor será el tiempo que este nodo se demora para checar el canal, si al checar el canal este nodo lo detecta ocupado, el nodo regresará a un estado inactivo hasta su próximo periodo de actividad, incrementando así la demora que se tiene en las retransmisiones.

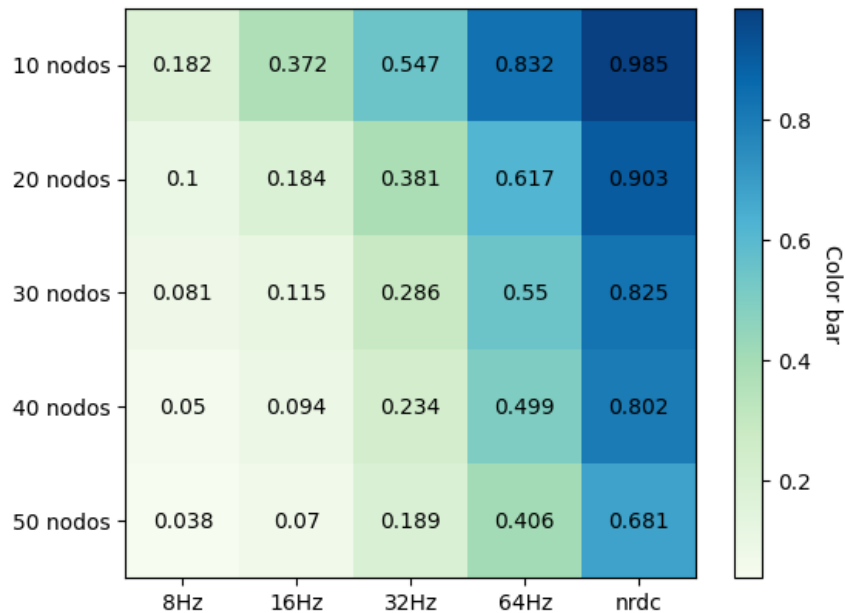


**Figura 37** Resultados obtenidos del algoritmo CBS, en donde  $\text{Delay}_{Cth}=2 \text{ RAD}=0.001$ , el eje X se refiere a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.

Como se explicó anteriormente en los epígrafes 4.1 y 0 y tomando como referencia la Figura 28 y la Tabla 9, los nodos de una WSN con ciclos de trabajo implementado habilitarán sus módulos de radio con el objetivo de recibir o transmitir un paquete. Para llevar a cabo esto previamente deberán checar los CCA que como se detalló en el epígrafe 2.3 este se basará en la RSSI para retornar un verdadero (señal de radio detectada) o falso (canal desocupado). En el caso de que el canal este ocupado y este nodo tenga la tarea de transmitir este simplemente apagará su módulo de radio hasta que sea su próxima oportunidad de despertarse y buscar nuevamente transmitir su paquete hasta que esta tarea sea exitosa. Hasta que el nodo no logre transmitir el tiempo de demora seguirá aumentando en relación a la frecuencia del ciclo de trabajo.

**Reachability:** Como ha sido tendencia en los anteriores algoritmos de broadcast analizados la **Reachability** ha aumentado conforme aumenta la frecuencia del ciclo de trabajo y este algoritmo en particular no ha

sido la excepción. En la Figura 38 podemos observar el comportamiento de este algoritmo respecto a la **Reachability**, teniendo como comparación por un eje X la frecuencia del ciclo de trabajo y por el eje Y las densidades de los nodos evaluadas en el trabajo de tesis. La tendencia mostró que conforme aumenta la frecuencia del ciclo de trabajo la **Reachability** aumenta. Esto se da en gran medida al comportamiento propio del ciclo de trabajo y su método para optimizar energía, tal como se explicó en el análisis de los resultados para los algoritmos del epígrafe 4.1 y 0.

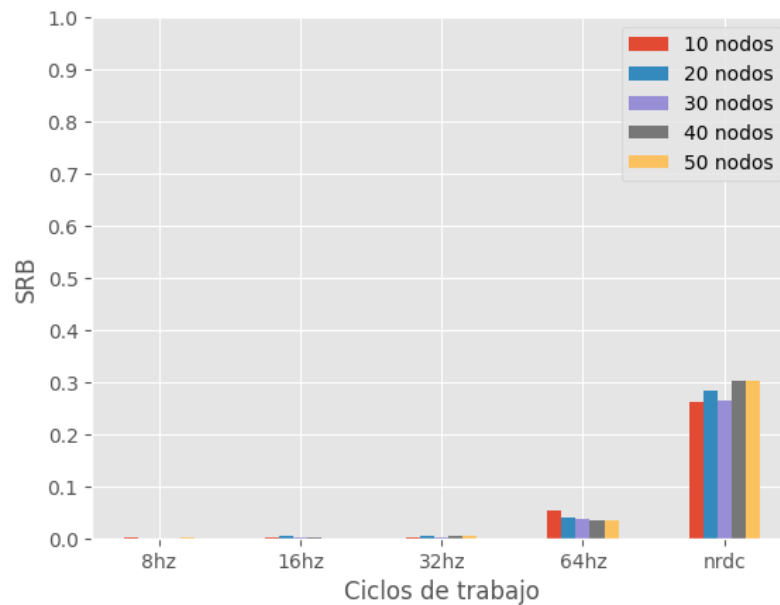


**Figura 38.** Resultados obtenidos del algoritmo CBS, en donde la **Reachability**  $C_{th}=2 \text{ RAD}=0.001$ , el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado.

En la Figura 30 y la Figura 34 donde se puede observar de manera detallada a que se debe la mejora de la **Reachability** conforme aumenta la frecuencia del ciclo de trabajo. Esto se debe a que mientras más alta es la frecuencia del ciclo de trabajo menos tiempo retransmitirá un mismo paquete un nodo que ocupe el medio de comunicación, o sea, menos tiempo ocupa el medio inalámbrico. Al ocupar menos tiempo el medio de comunicación las probabilidades que un nodo despierte y se encuentre el canal ocupado en dos o más transmisiones simultáneas disminuye, reduciendo así la pérdida de paquetes por activación del mecanismo de detección de paquetes y **fast sleep** que se detalló en la sección 2.3.2. Contrario a lo que pasaría si la frecuencia del ciclo de trabajo fuera baja, donde los nodos ocuparían por mayor tiempo el canal y esto traería como consecuencia una mayor probabilidad de solapar transmisiones de manera simultánea.

**Saved Rebroadcast:** Contrario a lo que paso en el análisis de esta métrica en los algoritmos de broadcast anteriores, esta no depende de una probabilidad de reenvío (ni fija ni adaptativa), sino más bien de la cantidad de información que se tenga en el medio de comunicación ya que este algoritmo de broadcast se basa en contadores. En este caso la tendencia a aumentar la SRB conforme aumenta la frecuencia del ciclo de trabajo se puede explicar de la mano con la métrica de la **Reachability**. Como se observó anteriormente en la Figura 38 conforme aumenta la frecuencia del ciclo de trabajo más información llega a todos los nodos de la red, al basarse este algoritmo de broadcast en conteo de un paquete repetido para decidir si se retransmite o no, mientras menos alcance haya en la red, menos información repetida recibirá un mismo nodo y menos retransmisiones evitará como vemos en la Figura 39.

En resumen, mientras aumenta la frecuencia del ciclo de trabajo aumenta la **Reachability**, lo que a su vez propicia la información duplicada recibida por un mismo nodo, que a su vez incrementa su contador interno para superar umbral y así evitar retransmitir dicho paquete repetido.



**Figura 39.** Resultado de la métrica **Saved Rebroadcast** Cth=2 RAD=0.001.

## Capítulo 5. Conclusiones, contribuciones y trabajo futuro

---

La principal aportación del presente trabajo de investigación es la evaluación del impacto que tiene el ciclo de trabajo sobre los algoritmos de broadcast sobre redes ad-hoc. Para estas conclusiones usamos el simulador de redes COOJA perteneciente al sistema operativo Contiki.

### 5.1 Conclusiones

Las principales conclusiones derivadas del presente trabajo son las siguientes:

A medida que aumenta la frecuencia del ciclo de trabajo, se observa una disminución en los *Delays* de propagación de paquetes, independientemente del algoritmo de *broadcast* y los parámetros específicos utilizados. Aunque estos parámetros afectan el valor puntual, la tendencia general persiste. El aumento de la frecuencia del ciclo de trabajo permite una transmisión más rápida de mensajes, reduciendo la latencia y mejorando la eficiencia en la entrega de paquetes en toda la red. Esto sugiere que ajustar adecuadamente la frecuencia del ciclo de trabajo es crucial para optimizar la eficiencia y el rendimiento en términos de retraso en la transmisión de paquetes.

En este trabajo de tesis podemos concluir que al aumentar la frecuencia del ciclo de trabajo la mejora de la *Reachability* es significativa alrededor de un 40% en promedio. Esto debido a que se reducen los tiempos de transmisiones conforme aumenta la frecuencia de los ciclos de trabajo se reduce la probabilidad de superposición de transmisiones y disminuye las colisiones, se logra una mayor capacidad para que los nodos receptores reciban paquetes sin problemas. Este aumento en la *Reachability* implica una mejora en la capacidad general de la red para alcanzar y entregar datos de manera más efectiva, lo que puede ser vital para una transmisión de datos confiable y oportuna. Esta tendencia se mantiene independientemente del algoritmo de *broadcast* que se implemente o los parámetros que se le cambien, estos solo afectarán su valor puntual, no así su tendencia a aumentar.

Respecto de la SRB, esta está más influenciada por factores relacionados con los algoritmos de *broadcast*, sus parámetros y las características de la topología de la red. Dicho comportamiento no exhibe una variación significativa (alrededor de un 5% como promedio) en respuesta al aumento de la frecuencia de los ciclos de trabajo. La gestión de la SRB sigue estando dominada por la interacción entre los nodos, la forma en que los algoritmos de *broadcast* están configurados y la topología de la red responden y

gestionan las retransmisiones de paquetes. En el caso del algoritmo de **broadcast** CBS si se observa una variación significativa cuando la frecuencia del ciclo de trabajo alcanza los 64Hz. Como se explicó en el epígrafe 4.3, esto se debe a que el algoritmo al basarse en contadores para retransmitir o no, debe recibir el mismo paquete repetido. Esto irá de la mano con la **Reachability**, mientras más alcance de un paquete a la red mayor probabilidad de recibir dicho paquete repetido en un nodo y por lo tanto aumentará la SRB en la red.

Una de las conclusiones cruciales derivadas de este trabajo de tesis es que implementar un algoritmo de **broadcast** diseñado originalmente para WSN sin ciclo de trabajo en una WSN con ciclo de trabajo no es una tarea sencilla. A lo largo del estudio se demostró que, en términos de **Reachability**, estos algoritmos operan de manera ineficiente en comparación con su diseño original. Esto se debe principalmente a cómo el ciclo de trabajo aborda las comunicaciones de tipo **broadcast** y cómo garantiza que los paquetes sean escuchados por todos los nodos vecinos. Este enfoque tiende a generar un mayor tráfico en la red y, por ende, un aumento significativo en las colisiones de paquetes, lo que puede resultar en el descarte de paquetes que los nodos receptores no logren detectar. Es esencial considerar estos desafíos al adaptar algoritmos de broadcast a entornos de WSN con ciclo de trabajo, lo que destaca la importancia de desarrollar enfoques específicos que aborden eficazmente las complejidades únicas de este tipo de redes.

## 5.2 Aportaciones del trabajo de investigación.

Las aportaciones de este trabajo de tesis son las siguientes:

Se llevó a cabo un análisis exhaustivo sobre cómo las diversas frecuencias para los ciclos de trabajo influyen en los algoritmos de **broadcast** en las WSNs. Se abordaron específicamente parámetros como el **Delay**, la **Reachability** y la **SRB**. En la investigación, se evidenció que a medida que estas frecuencias de trabajo aumentan, las métricas de desempeño experimentan mejoras significativas. Se explicó en detalle la interacción entre el ciclo de trabajo y la red, proporcionando una comprensión clara de este fenómeno.

Se validó una cama de pruebas implementada que podría ser utilizadas para futuros trabajos donde se requiera evaluar el impacto que pueden llegar a tener estos ciclos de trabajo en los algoritmos de **broadcast** o en WSN en general basándose en otras métricas, o usando otros algoritmos de **broadcast** existentes en la literatura.

Se creó una referencia a la hora de buscar implementar una WSN con ciclos de trabajo, donde se pueda elegir que algoritmo de **broadcast** usar, teniendo como premisas las métricas de desempeño evaluadas en este trabajo.

Se diseñó una metodología que brinda una estructura para la evaluación del rendimiento de los algoritmos de broadcast en WSN que emplean ciclos de trabajo. Su estructura flexible permite su adaptación para la evaluación de futuras propuestas de algoritmos de broadcast. Esta herramienta puede servir como un marco de referencia para investigaciones futuras, facilitando la comparación sistemática y precisa de los diversos enfoques algorítmicos y sus efectos en la eficiencia general de las redes de sensores. Además, su implementación podría suponer un avance significativo en el diseño y la optimización de algoritmos de **broadcast** más eficientes y adaptables que mejoren la capacidad de respuesta y la fiabilidad de las redes en diversas aplicaciones prácticas.

### 5.3 Trabajo futuro

Implementar otros algoritmos de **broadcast**, siguiendo la misma metodología para recabar más información acerca del impacto de los ciclos de trabajo sobre estos.

Realizar estudios adicionales que exploren más a fondo las interacciones entre los algoritmos de broadcast y la variación de la frecuencia de los ciclos de trabajo en entornos de WSN más complejas y que su topología tenga una distribución práctica de una aplicación y no genérica como en este trabajo de investigación.

Considerando la degradación de los algoritmos de broadcast existentes en WSN, especialmente en las métricas de **Delay** y **Reachability**, se abre una ventana de oportunidades para el diseño de nuevos algoritmos de broadcast. Estos podrían enfocarse en mejorar estos aspectos específicos en redes que emplean ciclos de trabajo.

Dado que se requiere realizar múltiples simulaciones para obtener una media estadística y resultados más confiables, este proceso se vuelve considerablemente lento. En mi opinión, una amplia área de investigación sería el desarrollo de un modelado matemático para comprender el comportamiento de estos procesos. Esto podría agilizar significativamente el tiempo empleado en las simulaciones durante el proceso de investigación.



Con base en las observaciones realizadas en esta investigación, se evidencia la importancia de tener en cuenta los mecanismos de transmisión de paquetes broadcast en las WSN con ciclos de trabajo para los algoritmos de **broadcast**. Esto ha sentado las bases para diseños futuros de algoritmos de **broadcast** más eficiente tomando en cuenta los mecanismos de transmisión de paquetes en redes con ciclos de trabajo, evitando la generación de mecanismos de redundancia distribuida que, como se demostró en el estudio, pueden resultar contraproducentes en términos de alcanzar una mayor **Reachability**. En su lugar, se propone una distribución estratégica de nodos transmisores en la red, de manera que abarquen un área más amplia de cobertura sin generar solapamientos ni el problema del nodo expuesto. La alternancia de estos nodos transmisores entre otros nodos de la red puede ayudar a evitar la sobrecarga de baterías en unos pocos nodos, contribuyendo así a una gestión energética más equitativa y eficaz dentro de la red. Este enfoque no solo optimizaría la eficiencia de la transmisión de paquetes, sino también garantizaría una mejor conservación de la energía, promoviendo así una mayor durabilidad y vida útil de la red en su conjunto.

## Literatura citada

---

- al-Humoud, S. O., Mackenzie, L. M., & Abdulai, J. (2008). Neighbourhood-aware counter-based broadcast scheme for wireless ad hoc networks. *2008 IEEE Globecom Workshops*, 1-6. <https://doi.org/10.1109/GLOCOMW.2008.ECP.45>
- Al-Obaisat, Y., & Braun, R. (2007). *On wireless sensor networks: Architectures, protocols, applications, and management*. <http://hdl.handle.net/10453/19594>
- Angueira, P., Val, I., Montalban, J., Seijo, Ó., Iradier, E., Fontaneda, P. S., Fanari, L., & Arriola, A. (2022). A survey of physical layer techniques for secure wireless communications in industry. *IEEE Communications Surveys & Tutorials*, 24(2), 810-838. <https://doi.org/10.1109/COMST.2022.3148857>
- Ashraf, S., Gao, M., Chen, Z., Haider, S. K., & Raza, Z. (2017). Efficient node monitoring mechanism in WSN using contikimac protocol. *International Journal of Advanced Computer Science and Applications*, 8(11). [https://www.researchgate.net/profile/Shahzad-Ashraf/publication/321496192\\_Efficient\\_Node\\_Monitoring\\_Mechanism\\_in\\_WSN\\_using\\_Contikimac\\_Protocol/links/5a683f75aca2720266b6a633/Efficient-Node-Monitoring-Mechanism-in-WSN-using-Contikimac-Protocol.pdf](https://www.researchgate.net/profile/Shahzad-Ashraf/publication/321496192_Efficient_Node_Monitoring_Mechanism_in_WSN_using_Contikimac_Protocol/links/5a683f75aca2720266b6a633/Efficient-Node-Monitoring-Mechanism-in-WSN-using-Contikimac-Protocol.pdf)
- Baggio, A. (2005). Wireless sensor networks in precision agriculture. *ACM workshop on real-world wireless sensor networks (REALWSN 2005), Stockholm, Sweden, 20*, 1567-1576. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=cdbc4cf0271a1049c8240ea70fd8579bdcdf0c97>
- Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A., & Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. *Computer communications*, 30(7), 1655-1695. <https://doi.org/10.1016/j.comcom.2006.12.020>
- Beck, M. (2005). *Ethernet in the First Mile: The IEEE 802.3 ah EFM standard*. McGraw-Hill Education. [https://scholar.google.com/scholar?hl=es&as\\_sdt=0%2C5&q=Ethernet+in+the+First+Mile%3A+The+IEEE+802.3+ah+EFM+standard&btnG=](https://scholar.google.com/scholar?hl=es&as_sdt=0%2C5&q=Ethernet+in+the+First+Mile%3A+The+IEEE+802.3+ah+EFM+standard&btnG=)
- Benrhaïem, W., & Hafid, A. S. (2020). Bayesian networks based reliable broadcast in vehicular networks. *Vehicular Communications*, 21, 100181. <https://doi.org/10.1016/j.vehcom.2019.100181>
- Cagalj, M., Ganeriwal, S., Aad, I., & Hubaux, J.-P. (2005). On selfish behavior in CSMA/CA networks. *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, 4, 2513-2524. <https://doi.org/10.1109/INFCOM.2005.1498536>
- Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5), 483-502. <https://doi.org/10.1002/wcm.72>
- Cardenas, R. A. H. (2019). *Evaluación del impacto del ciclo de trabajo del transceptor de radio al proceso de formación de redes inalámbricas de sensores multialto en el contexto de la Internet de las Cosas (IoT)* [Tesis de maestría en ciencias, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California]. <http://cicese.repositorioinstitucional.mx/jspui/handle/1007/3023>

- Carrano, R. C., Passos, D., Magalhaes, L. C., & Albuquerque, C. V. (2013). Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 16(1), 181-194. <https://doi.org/10.1109/SURV.2013.052213.00116>
- Cartigny, J., & Simplot, D. (2003). Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, 10 pp. <https://doi.org/10.1109/HICSS.2003.1174853>
- Catenazzo, D., O'Flynn, B., & Walsh, M. (2018). On the use of wireless sensor networks in preventative maintenance for industry 4.0. *2018 12th International Conference on Sensing Technology (ICST)*, 256-262. <https://doi.org/10.1109/ICSensT.2018.8603669>
- Charfi, F., & Bouyahi, M. (2012). Performance evaluation of beacon enabled IEEE 802.15. 4 under NS2. *arXiv preprint arXiv:1204.1495*. <https://doi.org/10.48550/arXiv.1204.1495>
- Chen, C., Hsu, C.-K., & Wang, H.-K. (2005). A distance-aware counter-based broadcast scheme for wireless ad hoc networks. *MILCOM 2005-2005 IEEE Military Communications Conference*, 1052-1058. <https://doi.org/10.1109/MILCOM.2005.1605818>
- Chen, Z., Liu, A., Li, Z., Choi, Y.-J., Sekiya, H., & Li, J. (2017). Energy-efficient broadcasting scheme for smart industrial wireless sensor networks. *Mobile Information Systems, 2017*. <https://doi.org/10.1155/2017/7538190>
- Cruz, L. A. R. (2017). *Análisis del impacto del número de usuarios y tasa de datos ofrecida en el traspaso entre resumideros de una WBAN/WPAN enfocada a aplicaciones de sistemas del cuidado de la salud* [Tesis de maestría en ciencias, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California]. <http://cicese.repositorioinstitucional.mx/jspui/handle/1007/1606>
- Czyz, J., Allman, M., Zhang, J., Iekel-Johnson, S., Osterweil, E., & Bailey, M. (2014). Measuring ipv6 adoption. *Proceedings of the 2014 ACM conference on SIGCOMM*, 87-98. <https://doi.org/10.1145/2619239.2626295>
- de Melo Pires, R., Arnosti, S. Z., Pinto, A. S. R., & Branco, K. R. C. (2016). Experimenting broadcast storm mitigation techniques in FANETs. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 5868-5877. <https://doi.org/10.1109/HICSS.2016.726>
- Dludla, A. G., Abu-Mahfouz, A. M., Kruger, C. P., & Isaac, J. S. (2013). Wireless sensor networks testbed: ASNTbed. *2013 IST-Africa Conference & Exhibition*, 1-10. <https://ieeexplore.ieee.org/abstract/document/6701766>
- Dunkels, A. (2011). *The contikimac radio duty cycling protocol*. Swedish Institute of Computer Science. <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1042289&dswid=-3684>
- Gaxiola, M. O. C. (2012). *Diseño de una red híbrida de dos saltos para aplicaciones de redes de sensores en telemonitoreo y telemedicina* [Tesis de maestría en ciencias, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California]. <http://cicese.repositorioinstitucional.mx/jspui/handle/1007/235>
- Haas, Z. J., Halpern, J. Y., & Li, L. (2002). Gossip-based ad hoc routing. *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 3, 1707-1716. <https://doi.org/10.1109/INFCOM.2002.1019424>

- Haas, Z. J., Halpern, J. Y., & Li, L. (2006). Gossip-based ad hoc routing. *IEEE/ACM Transactions on networking*, 14(3), 479-491. <https://doi.org/10.1109/TNET.2006.876186>
- Hoebeker, J., Moerman, I., Dhoedt, B., & Demeester, P. (2004). An overview of mobile ad hoc networks: Applications and challenges. *Journal-Communications Network*, 3(3), 60-66. [https://its-wiki.no/images/7/7a/Manet\\_Overview.pdf](https://its-wiki.no/images/7/7a/Manet_Overview.pdf)
- Hsin, C., & Liu, M. (2006). Randomly duty-cycled wireless sensor networks: Dynamics of coverage. *IEEE Transactions on Wireless Communications*, 5(11), 3182-3192. <https://doi.org/10.1109/TWC.2006.04861>
- Ibrahim, B. F., Toycan, M., & Mawlood, H. A. (2020). A comprehensive survey on VANET broadcast protocols. *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, 298-302. <https://doi.org/10.1109/ICCAKM46823.2020.9051462>
- Jung, J.-Y., Seo, D.-Y., & Lee, J.-R. (2018). Counter-based broadcast scheme considering reachability, network density, and energy efficiency for wireless sensor networks. *Sensors*, 18(1), 120. <https://doi.org/10.3390/s18010120>
- Jurdak, R., Baldi, P., & Lopes, C. V. (2007). Adaptive low power listening for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 6(8), 988-1004. <https://doi.org/10.1109/TMC.2007.1037>
- Kandris, D., Nakas, C., Vomvas, D., & Koulouras, G. (2020). Applications of wireless sensor networks: An up-to-date survey. *Applied system innovation*, 3(1), 14. <https://doi.org/10.3390/asi3010014>
- Khanafer, M., Guennoun, M., & Mouftah, H. T. (2013). A survey of beacon-enabled IEEE 802.15. 4 MAC protocols in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 16(2), 856-876. <https://doi.org/10.1109/SURV.2013.112613.00094>
- Kifle, D., Gianini, G., & Libsie, M. (2019). Improving probabilistic flooding using topological indexes. *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 376-382. <https://doi.org/10.1109/SITIS.2019.00067>
- Kodali, R. K., & Malothu, V. K. (2016). MIXIM framework simulation of WSN with QoS. *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 128-131. <https://doi.org/10.1109/ICACCCT.2016.7831614>
- Kroese, D. P., Brereton, T., Taimre, T., & Botev, Z. I. (2014). Why the Monte Carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6), 386-392. <https://doi.org/10.1002/wics.1314>
- Kumar, S., Dalal, S., & Dixit, V. (2014). The OSI model: Overview on the seven layers of computer networks. *International Journal of Computer Science and Information Technology Research*, 2(3), 461-466. <https://shorturl.at/gFP25>
- Lauwens, B., Scheers, B., & Van de Capelle, A. (2010). Performance analysis of unslotted CSMA/CA in wireless networks. *Telecommunication Systems*, 44, 109-123. <https://doi.org/10.1007/s11235-009-9220-7>

- Li, Y., Li, D., Cui, W., & Zhang, R. (2011). Research based on OSI model. *2011 IEEE 3rd International Conference on Communication Software and Networks*, 554-557. <https://doi.org/10.1109/ICCSN.2011.6014631>
- Liang, O., Sekercioglu, Y. A., & Mani, N. (2006). A survey of multipoint relay based broadcast schemes in wireless ad hoc networks. *IEEE Communications Surveys & Tutorials*, 8(4), 30-46. <https://doi.org/10.1109/COMST.2006.283820>
- Mahmood, A., Javaid, N., & Razzaq, S. (2015). A review of wireless communications for smart grid. *Renewable and sustainable energy reviews*, 41, 248-260. <https://doi.org/10.1016/j.rser.2014.08.036>
- Majid, M., Habib, S., Javed, A. R., Rizwan, M., Srivastava, G., Gadekallu, T. R., & Lin, J. C.-W. (2022). Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review. *Sensors*, 22(6), 2087. <https://doi.org/10.3390/s22062087>
- Malik, N., Ragiri, P. R., & Jain, A. (2018). Study of Security Mechanisms and Simulation Analysis of Packet Drop in RPL in Low Power Lossy Networks. *International Journal of Future Generation Communication and Networking*, 11(5), 45-62. <https://api.semanticscholar.org/CorpusID:69618843>
- Mohammed, A., Ould-Khaoua, M., & Mackenzie, L. (2007). An efficient counter-based broadcast scheme for mobile ad hoc networks. *Formal Methods and Stochastic Models for Performance Evaluation: Fourth European Performance Engineering Workshop, EPEW 2007, Berlin, Germany, September 27-28, 2007. Proceedings 4*, 275-283. [https://doi.org/10.1007/978-3-540-75211-0\\_20](https://doi.org/10.1007/978-3-540-75211-0_20)
- Mohammed, A., Ould-Khaoua, M., & Mackenzie, L. (2009). An improved rebroadcast probability function for an efficient counter-based broadcast scheme in MANETs. *25th Annual UK Performance Engineering Workshop (UKPEW'09)*. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=27c8f810799472c4e6bc4f9785f238db543391af>
- Mohammed, A., Ould-Khaoua, M., & Mackenzie, L. M. (2008). Improvement to efficient counter-based broadcast scheme through random assessment delay adaptation for MANETs. *2008 Second UKSIM European Symposium on Computer Modeling and Simulation*, 536-541. <https://doi.org/10.1109/EMS.2008.69>
- Mulligan, G. (2007). The 6LoWPAN architecture. *Proceedings of the 4th workshop on Embedded networked sensors*, 78-82. <https://doi.org/10.1145/1278972.1278992>
- Nasipuri, A., Zhuang, J., & Das, S. R. (1999). A multichannel CSMA MAC protocol for multihop wireless networks. *WCNC. 1999 IEEE Wireless Communications and Networking Conference (Cat. No. 99TH8466)*, 3, 1402-1406. <https://doi.org/10.1109/WCNC.1999.796968>
- Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., & Sheu, J.-P. (1999). The broadcast storm problem in a mobile ad hoc network. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 151-162. <https://dl.acm.org/doi/pdf/10.1145/313451.313525>

- Ojha, T., Misra, S., & Raghuwanshi, N. S. (2015). Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Computers and electronics in agriculture*, *118*, 66-84. <https://doi.org/10.1016/j.compag.2015.08.011>
- Ramson, S. J., & Moni, D. J. (2017). Applications of wireless sensor networks—A survey. *2017 international conference on innovations in electrical, electronics, instrumentation and media technology (ICEEIMT)*, 325-329. <https://doi.org/10.1109/ICEEIMT.2017.8116858>
- Rege, V. (2015). Design and implementation of an antenna model for the cooja simulator. *arXiv preprint arXiv:1610.06129*. <https://doi.org/10.48550/arXiv.1610.06129>
- Reina, D. G., Günes, M., & Toral, S. L. (2016). Real experimentation of probabilistic broadcasting algorithms based on dissimilarity metrics for multi-hop ad hoc networks. *Ad Hoc Networks*, *47*, 1-15. <https://doi.org/10.1016/j.adhoc.2016.04.002>
- Reina, D. G., Toral, S. L., Johnson, P., & Barrero, F. (2015). A survey on probabilistic broadcast schemes for wireless ad hoc networks. *Ad Hoc Networks*, *25*, 263-292. <https://doi.org/10.1016/j.adhoc.2014.10.001>
- Roussel, K., & Zendra, O. (2016). Using Cooja for WSN simulations: Some new uses and limits. *EWSN 2016—NextMote workshop*, 319–324. <https://inria.hal.science/hal-01240986/document>
- Rout, R. R., & Ghosh, S. K. (2012). Enhancement of lifetime using duty cycle and network coding in wireless sensor networks. *IEEE Transactions on Wireless Communications*, *12*(2), 656-667. <https://doi.org/10.1109/TWC.2012.111412.112124>
- Rubinstein, R. Y., & Kroese, D. P. (2016). *Simulation and the Monte Carlo method*. John Wiley & Sons. [https://books.google.com.mx/books?hl=es&lr=&id=r2VODQAAQBAJ&oi=fnd&pg=PR1&dq=Simulation+and+the+Monte+Carlo+method&ots=16PZhf3c2b&sig=gw3LTZ4PpJt4OibMNfFcuJLXdvg&redir\\_esc=y#v=onepage&q=Simulation%20and%20the%20Monte%20Carlo%20method&f=false](https://books.google.com.mx/books?hl=es&lr=&id=r2VODQAAQBAJ&oi=fnd&pg=PR1&dq=Simulation+and+the+Monte+Carlo+method&ots=16PZhf3c2b&sig=gw3LTZ4PpJt4OibMNfFcuJLXdvg&redir_esc=y#v=onepage&q=Simulation%20and%20the%20Monte%20Carlo%20method&f=false)
- Sasson, Y., Cavin, D., & Schiper, A. (2003). Probabilistic broadcast for flooding in wireless mobile ad hoc networks. *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, *2*, 1124-1130. <https://doi.org/10.1109/WCNC.2003.1200529>
- Sharma, S., Kumar, D., & Kishore, K. (2013). Wireless sensor networks-A review on topologies and node architecture. *International Journal of Computer Sciences and Engineering*, *1*(2), 19-25. [https://www.researchgate.net/profile/Shamneesh-Sharma/publication/309180675\\_Wireless\\_Sensor\\_Networks-A\\_Review\\_on\\_Topologies\\_and\\_Node\\_Architecture/links/580f019608ae7525273d45cf/Wireless-Sensor-Networks-A-Review-on-Topologies-and-Node-Architecture.pdf](https://www.researchgate.net/profile/Shamneesh-Sharma/publication/309180675_Wireless_Sensor_Networks-A_Review_on_Topologies_and_Node_Architecture/links/580f019608ae7525273d45cf/Wireless-Sensor-Networks-A-Review-on-Topologies-and-Node-Architecture.pdf)
- Shen, C.-C., Huang, Z., & Jaikao, C. (2006). Directional broadcast for mobile ad hoc networks with percolation theory. *IEEE Transactions on Mobile Computing*, *5*(4), 317-332. <https://doi.org/10.1109/TMC.2006.1599402>
- Shi, Z., & Shen, H. (2004). Adaptive gossip-based routing algorithm. *IPCCC*, 323-324. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=19b6a2788a1059cccc0f5953333257453ee46598>

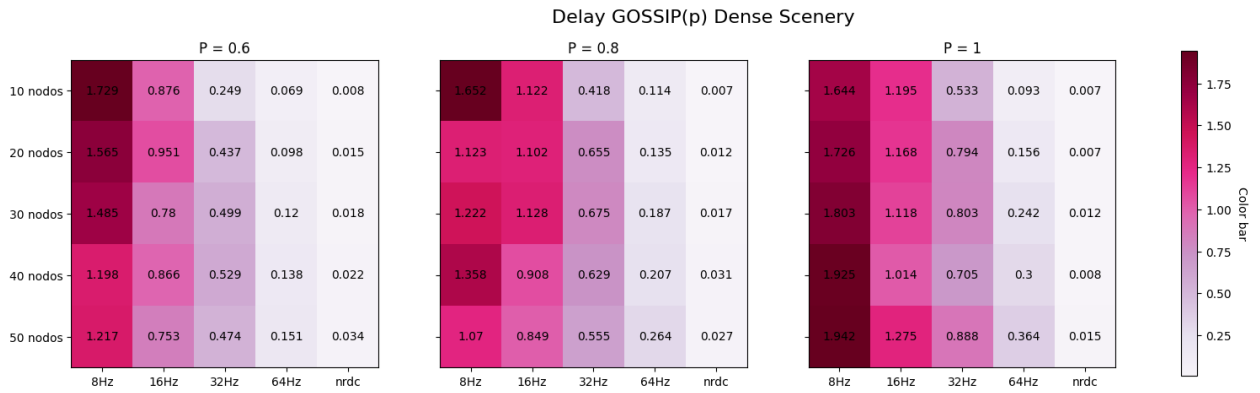
- Shnayder, V., Hempstead, M., Chen, B., Allen, G. W., & Welsh, M. (2004). Simulating the power consumption of large-scale sensor network applications. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 188-200. <https://doi.org/10.1145/1031495.1031518>
- Shu, T., Liu, W., Wang, T., Deng, Q., Zhao, M., Xiong, N. N., Li, X., & Liu, A. (2019). Broadcast based code dissemination scheme for duty cycle based wireless sensor networks. *IEEE Access*, 7, 105258-105286. <https://doi.org/10.1109/ACCESS.2019.2932006>
- Siraj, S., Gupta, A., & Badgajar, R. (2012). Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(4), 199-206. <https://www.ijarcce.com/upload/june/3-Network%20Simulation%20Tools%20Survey.pdf>
- Sobol, I. M. (2018). *A primer for the Monte Carlo method*. CRC press. <https://doi.org/10.1201/9781315136448>
- Sundani, H., Li, H., Devabhaktuni, V., Alam, M., & Bhattacharya, P. (2011). Wireless sensor network simulators a survey and comparisons. *International Journal of Computer Networks*, 2(5), 249-265. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=310fc34dc89736594e2d0fa9ca6bf489d7d275c7>
- Tay, Y. C., Jamieson, K., & Balakrishnan, H. (2004). Collision-minimizing CSMA and its applications to wireless sensor networks. *IEEE Journal on selected areas in Communications*, 22(6), 1048-1057. <https://doi.org/10.1109/JSAC.2004.830898>
- Thomson, C., Romdhani, I., Al-Dubai, A., Qasem, M., Ghaleb, B., & Wadhaj, I. (2016). *Cooja simulator manual*. <https://doi.org/10.13140/RG.2.1.4274.8408>
- Tilak, S., Murphy, A., & Heinzelman, W. (2003). Non-uniform information dissemination for sensor networks. *11th IEEE International Conference on Network Protocols, 2003. Proceedings.*, 295-304. <https://doi.org/10.1109/ICNP.2003.1249779>
- Uwase, M.-P., Bezunartea, M., Tiberghien, J., Dricot, J.-M., & Steenhaut, K. (2017). Experimental comparison of radio duty cycling protocols for wireless sensor networks. *IEEE sensors journal*, 17(19), 6474-6482. <https://doi.org/10.1109/JSEN.2017.2738700>
- Velinov, A., & Mileva, A. (2016). *Running and testing applications for Contiki OS using Cooja simulator*. <http://www.tfzr.rs/itro/Zbornik%20ITRO%202016.pdf>
- Wang, X. Y., Dokania, R. K., & Apsel, A. (2010). PCO-based synchronization for cognitive duty-cycled impulse radio sensor networks. *IEEE Sensors Journal*, 11(3), 555-564. <https://doi.org/10.1109/JSEN.2010.2051326>
- Williams, B., & Camp, T. (2002). Comparison of broadcasting techniques for mobile ad hoc networks. *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 194-205. <https://doi.org/10.1145/513800.513825>
- Wisitpongphan, N., Tonguz, O. K., Parikh, J. S., Mudalige, P., Bai, F., & Sadekar, V. (2007). Broadcast storm mitigation techniques in vehicular ad hoc networks. *IEEE Wireless Communications*, 14(6), 84-94. <https://doi.org/10.1109/MWC.2007.4407231>

- Yassein, M. B., Nimer, S. F., & Al-Dubai, A. Y. (2011). A new dynamic counter-based broadcasting scheme for mobile ad hoc networks. *Simulation Modelling Practice and Theory*, 19(1), 553-563. <https://doi.org/10.1016/j.simpat.2010.08.011>
- Ye, W., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on networking*, 12(3), 493-506. <https://doi.org/10.1109/TNET.2004.828953>
- Zakon, R. (1997). *Hobbes' Internet Timeline*. <https://www.rfc-editor.org/rfc/rfc2235.html>
- Zen, K., Habibi, D., Rassau, A., & Ahmad, I. (2008). Performance evaluation of IEEE 802.15. 4 for mobile sensor networks. *2008 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN'08)*, 1-5. <https://doi.org/10.1109/WOCN.2008.4542536>
- Zhang, Q., & Agrawal, D. P. (2005). Performance evaluation of leveled probabilistic broadcasting in MANETs and wireless sensor networks. *Simulation*, 81(8), 533-546. <https://doi.org/10.1177/0037549705060234>
- Zhou, H., & Guan, X. (2009). Idle-listening reduction for data aggregation in distributed sensor networks. *IEEE Transaction on Parallel and Distributed Systems*, 1-13. <https://www.paradise.caltech.edu/~hzhou/papers/TPDS09/Reed.pdf>

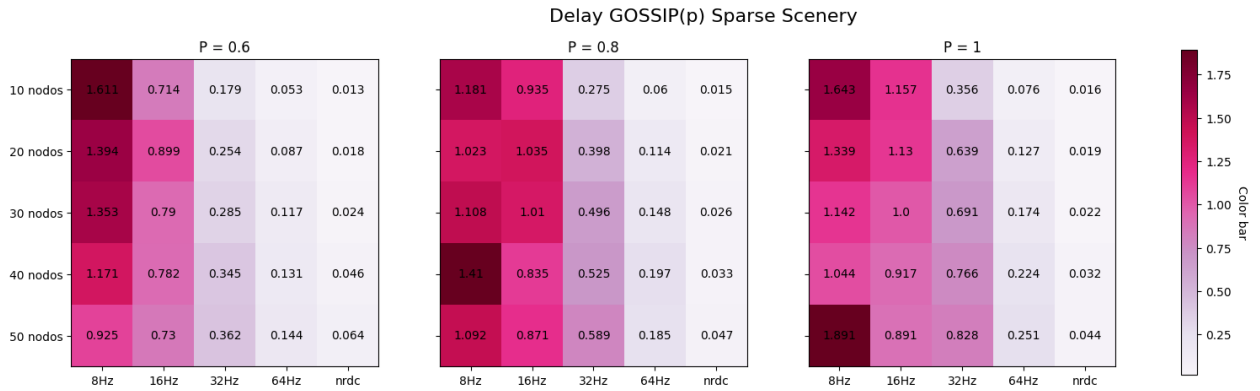


## Anexos

### Anexo A

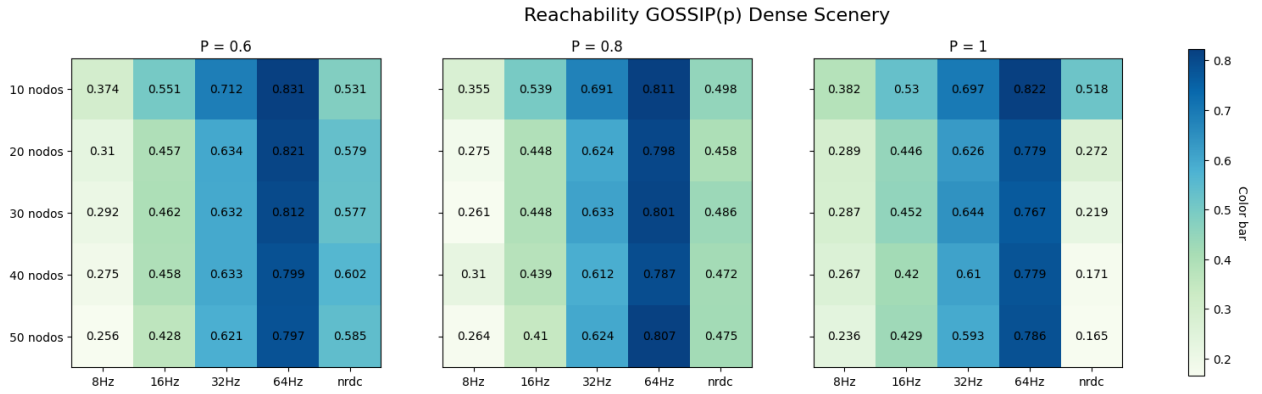


**Figura 40.** Resultados obtenidos del algoritmo GOSSIP(p) en escenarios densos, en donde el **Delay**  $p = 0.6$ ,  $p=0.8$ ,  $p=1$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.

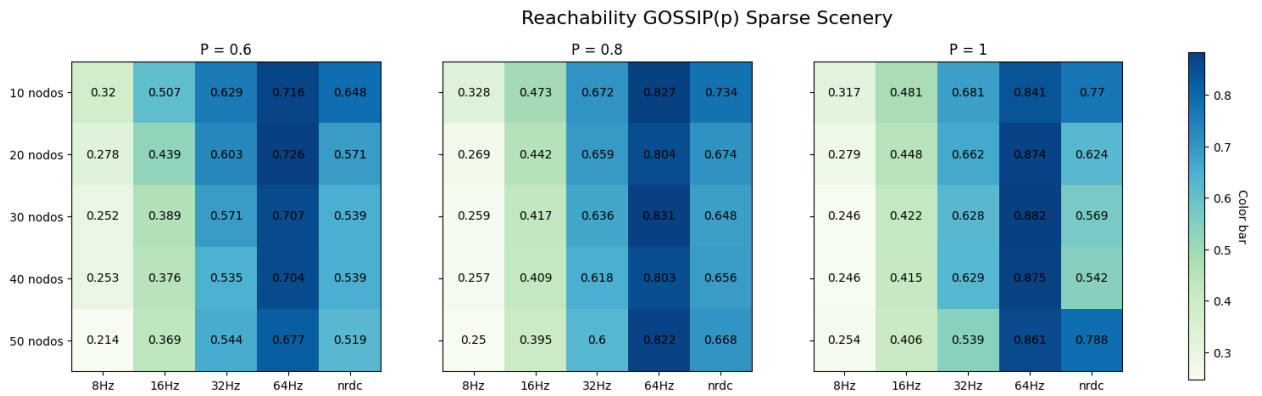


**Figura 41.** Resultados obtenidos del algoritmo GOSSIP(p) en escenarios dispersos, en donde el **Delay**  $p = 0.6$ ,  $p=0.8$ ,  $p=1$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.

## Anexo B

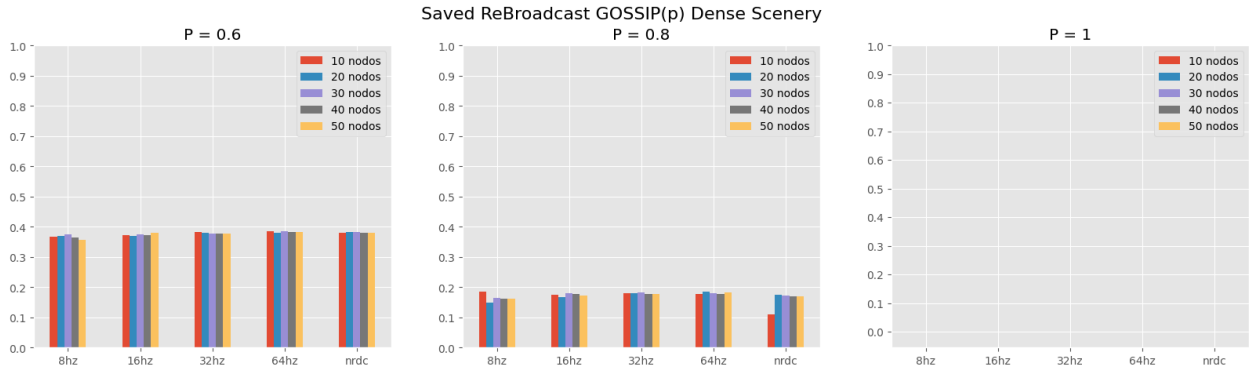


**Figura 42.** Resultados obtenidos del algoritmo GOSSIP(p) en escenarios densos, en donde la *Reachability*  $p = 0.6$ ,  $p=0.8$ ,  $p=1$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una *Reachability* más grande siendo un resultado deseado.

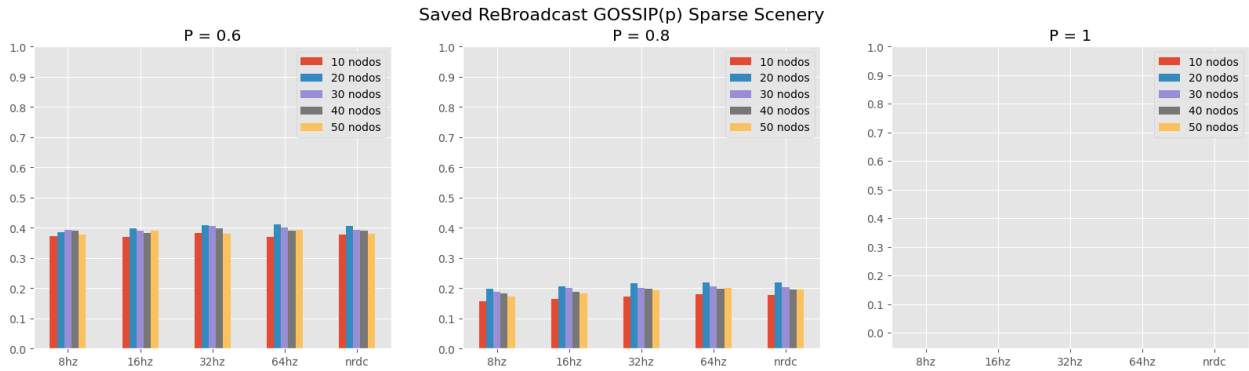


**Figura 43.** Resultados obtenidos del algoritmo GOSSIP(p) en escenarios dispersos, en donde la *Reachability*  $p = 0.6$ ,  $p=0.8$ ,  $p=1$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una *Reachability* más grande siendo un resultado deseado.

## Anexo C

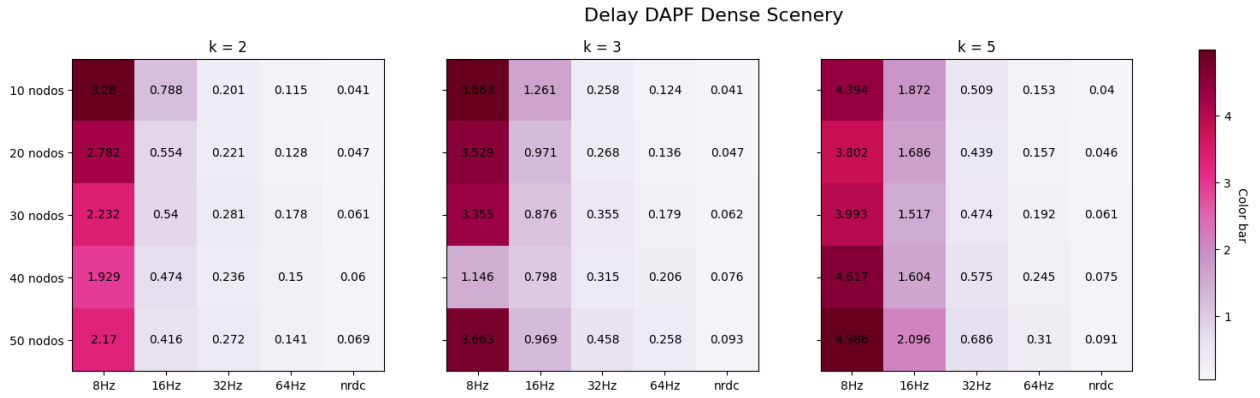


**Figura 44.** Resultado de la métrica *Saved Rebroadcast* para el esquema GOSSIP1(p) para p=0.6, p=0.8, p=1 respectivamente para escenarios densos.

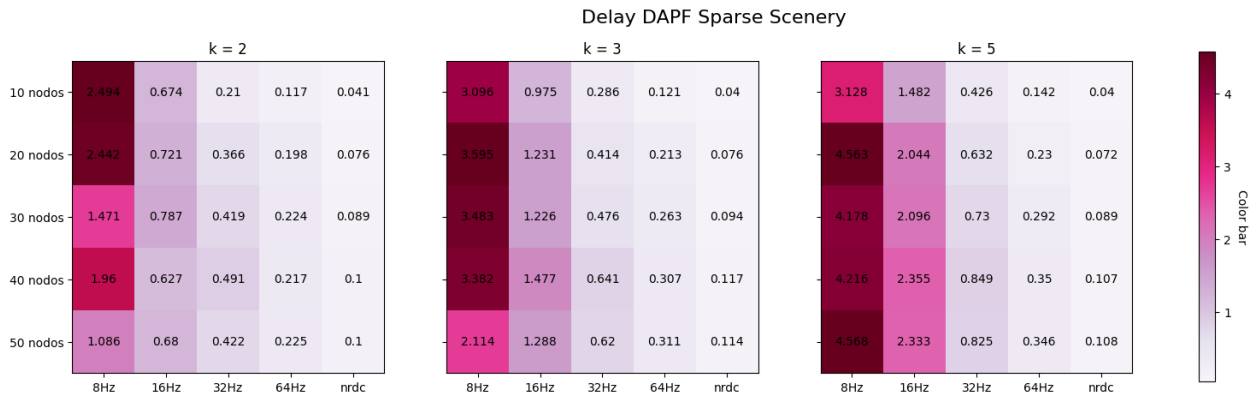


**Figura 45.** Resultado de la métrica *Saved Rebroadcast* para el esquema GOSSIP1(p) para p=0.6, p=0.8, p=1 respectivamente para escenarios dispersos.

## Anexo D

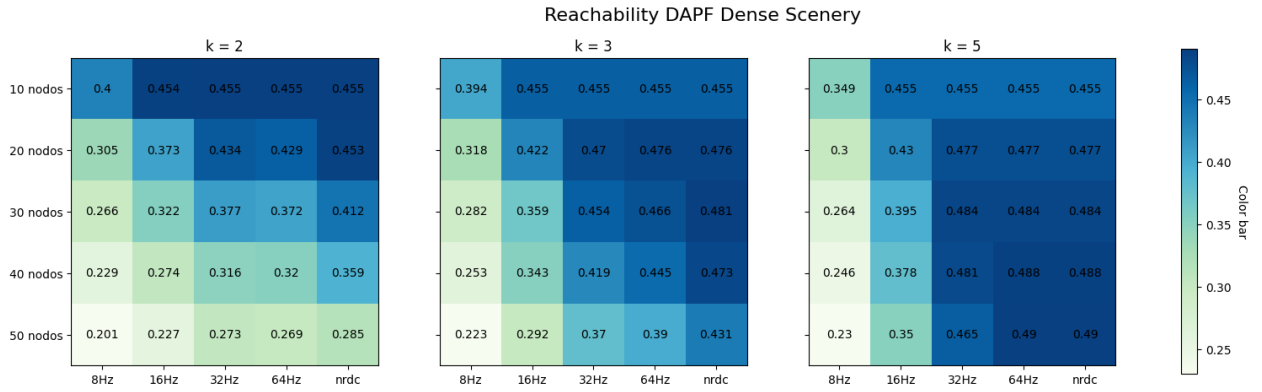


**Figura 46.** Resultados obtenidos del algoritmo DAPF en escenarios densos, en donde el **Delay**  $k=2, k=3, k=5$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.

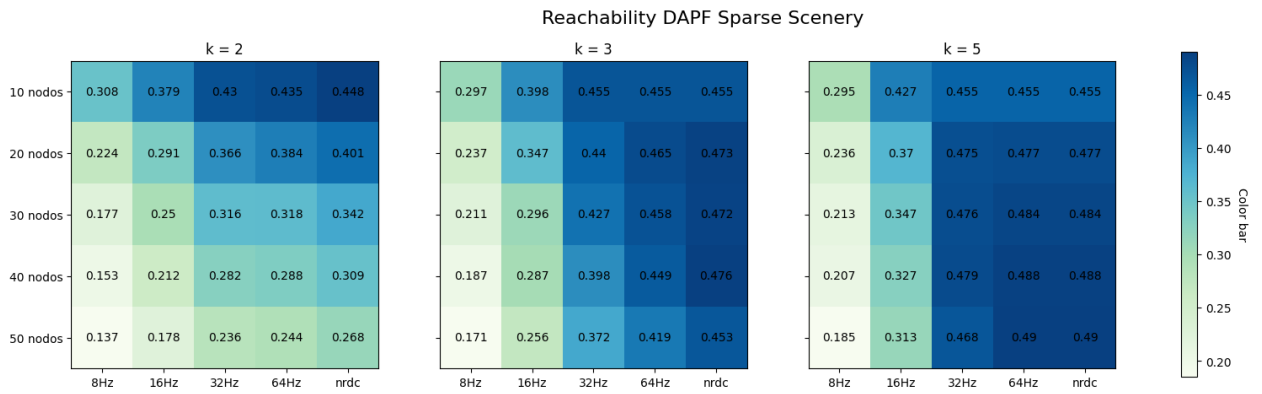


**Figura 47.** Resultados obtenidos del algoritmo DAPF en escenarios disperso, en donde el **Delay**  $k=2, k=3, k=5$  respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.

## Anexo E

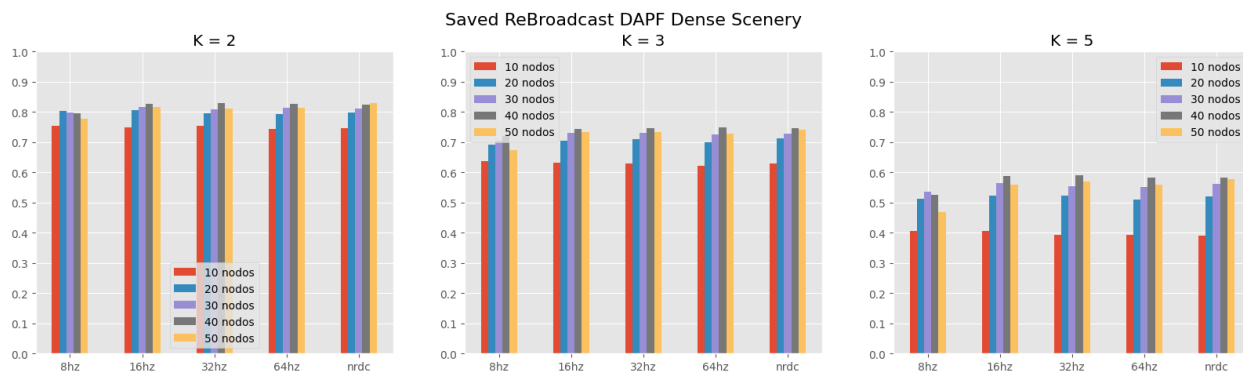


**Figura 48.** Resultados obtenidos del algoritmo DAPF en escenarios densos, en donde la *Reachability* k=2, k=3, k=5 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una *Reachability* más grande siendo un resultado deseado.

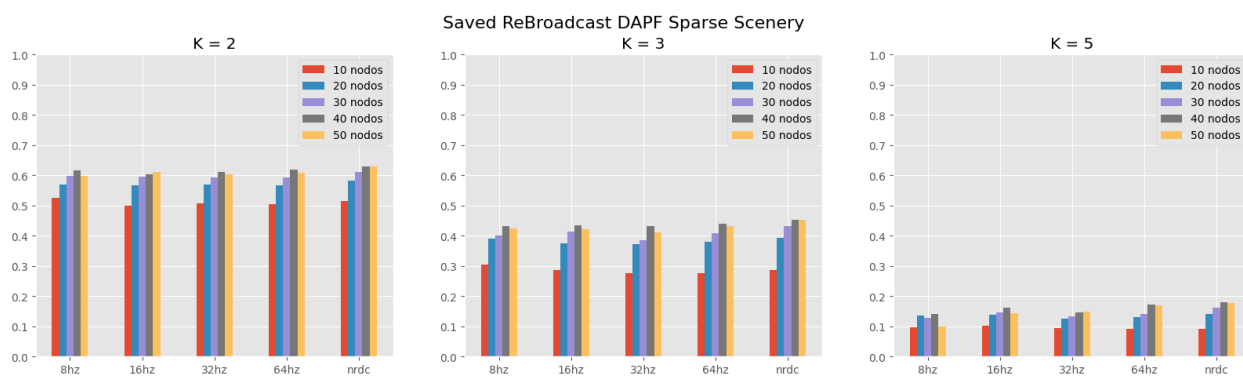


**Figura 49.** Resultados obtenidos del algoritmo DAPF en escenarios dispersos, en donde la *Reachability* k=2, k=3, k=5 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una *Reachability* más grande siendo un resultado deseado.

## Anexo F

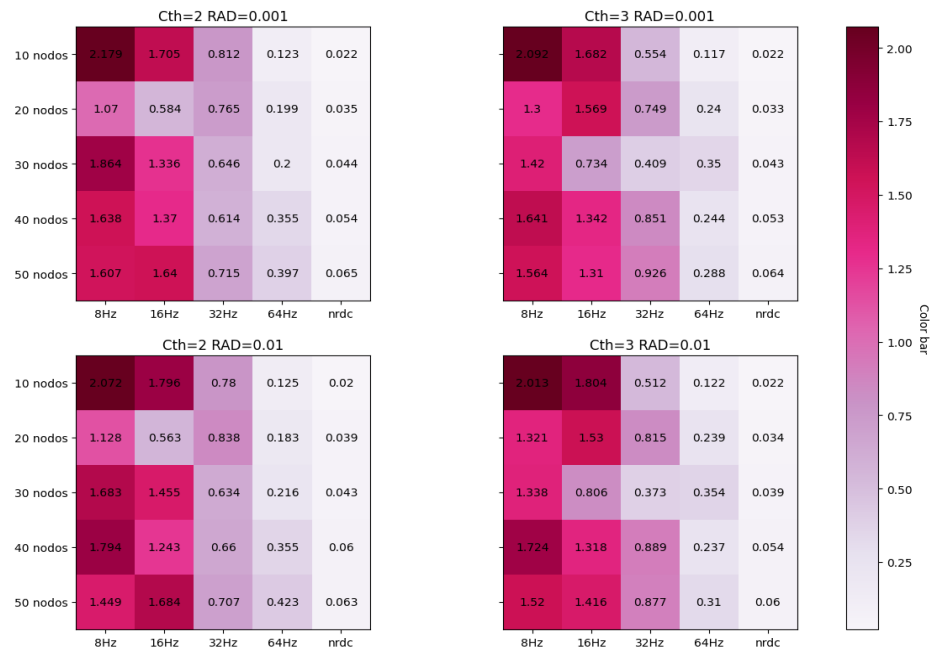


**Figura 50.** Resultado de la métrica *Saved ReBroadcast* para el esquema DAPF para K=2, K=3, K=5 respectivamente para escenarios densos.

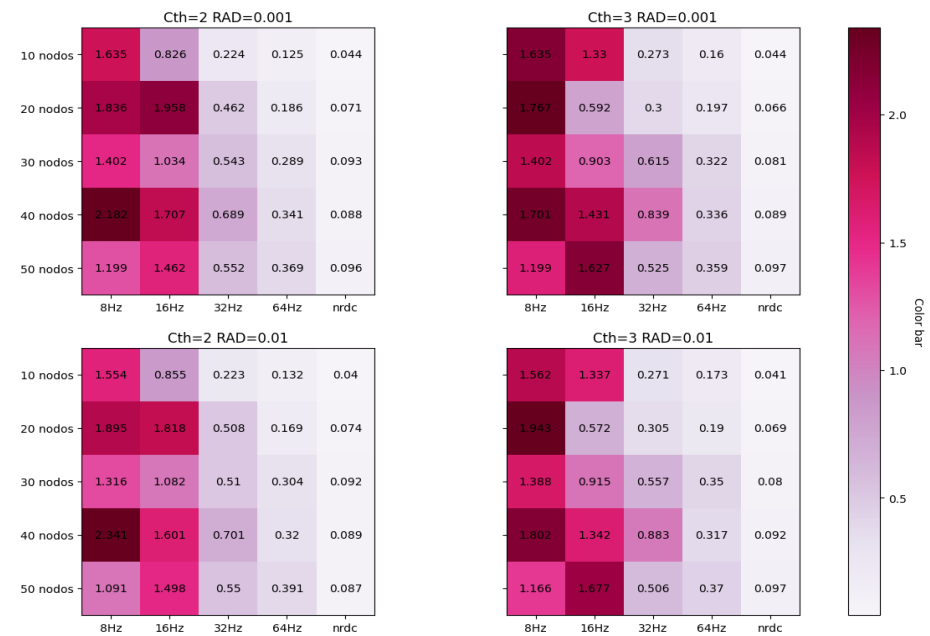


**Figura 51.** Resultado de la métrica *Saved ReBroadcast* para el esquema DAPF para K=2, K=3, K=5 respectivamente para escenarios dispersos.

## Anexo G



**Figura 52.** Resultados obtenidos del algoritmo CBS en escenarios densos, en donde el **Delay** Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.



**Figura 53.** Resultados obtenidos del algoritmo CBS en escenarios dispersos, en donde el **Delay** Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la densidad de nodos. El color más claro es significado de un **Delay** más pequeño siendo un resultado deseado.

# Anexo H

Reachability CBS Dense Scenery

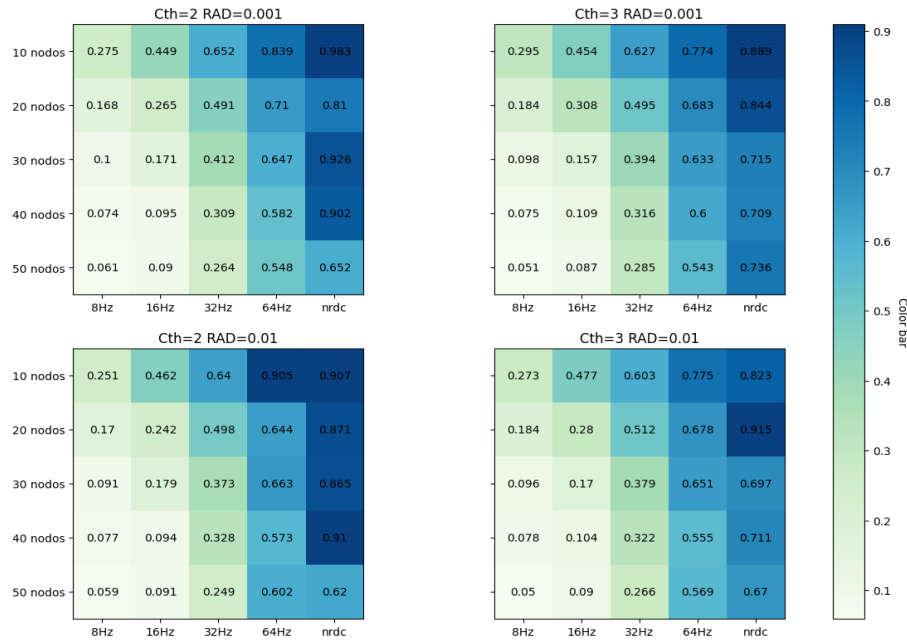


Figura 54. Resultados obtenidos del algoritmo CBS en escenarios densos, en donde la **Reachability** Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado.

Reachability CBS Sparse Scenery

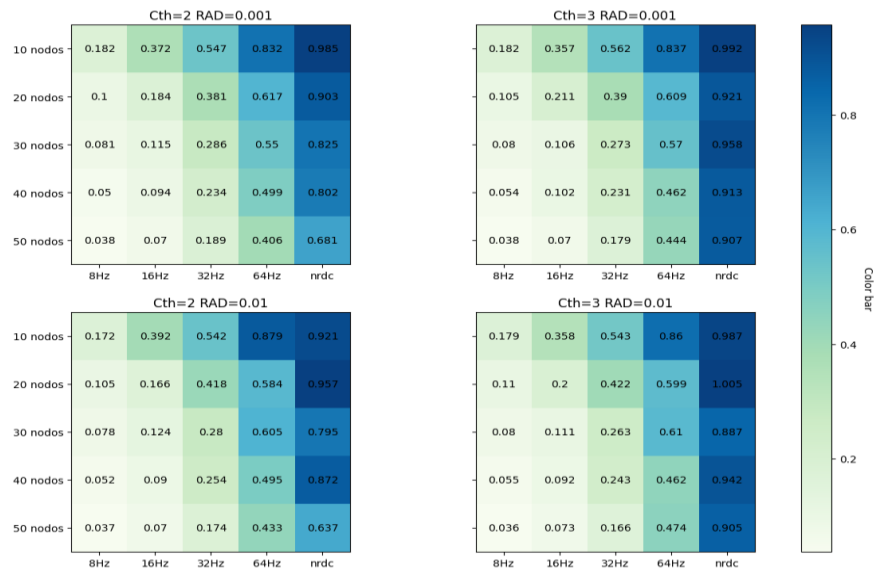
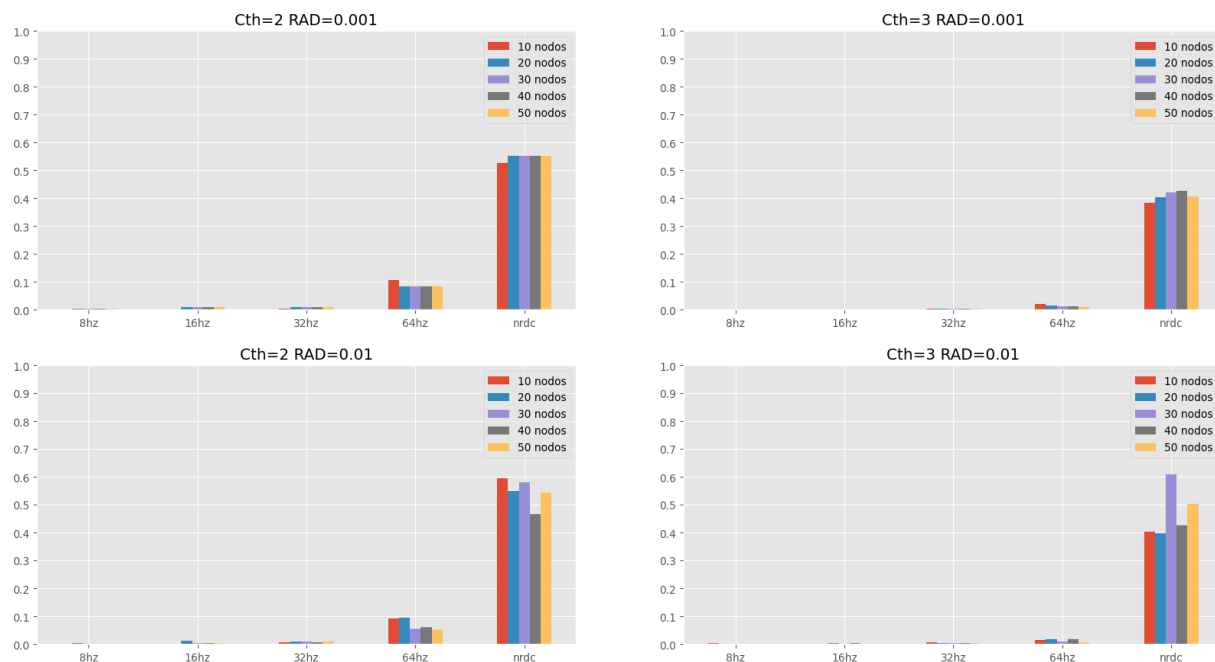


Figura 55. Resultados obtenidos del algoritmo CBS en escenarios dispersos, en donde la **Reachability** Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente, el eje X corresponde a la frecuencia de ciclo de trabajo, el eje Y a la Densidad de nodos. El color más intenso es significado de una **Reachability** más grande siendo un resultado deseado.



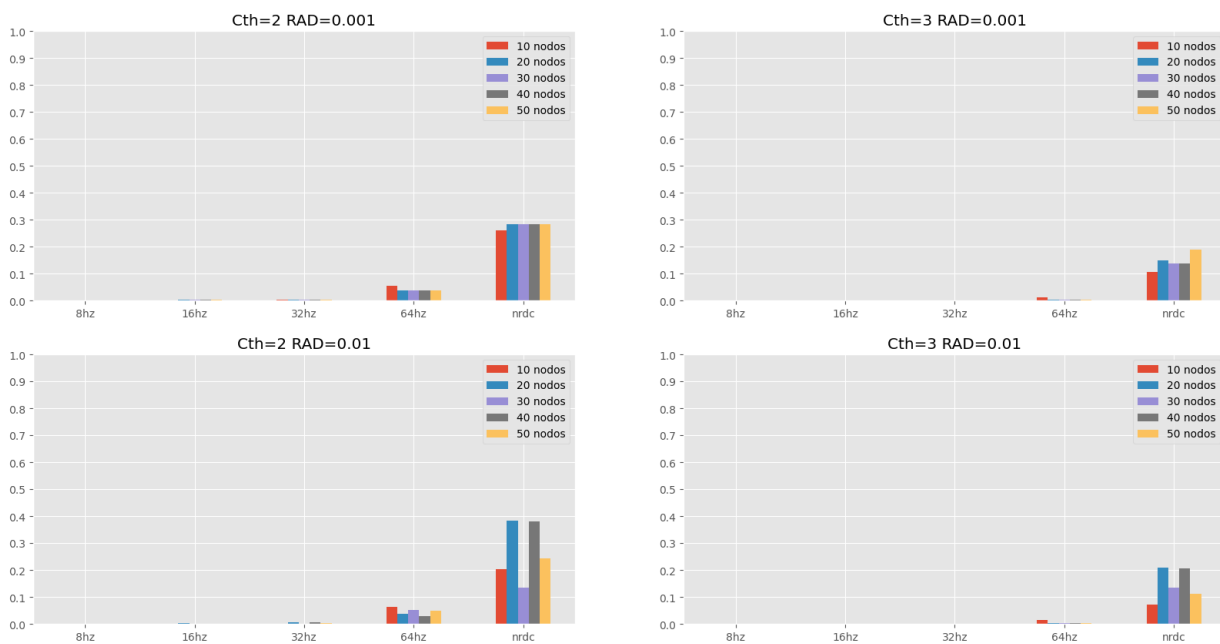
## Anexo I

Saved ReBroadcast CBS Dense Scenery



**Figura 56.** Resultado de la métrica *Saved Rebroadcast* para el esquema CBS para Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente para escenarios densos.

Saved ReBroadcast CBS Sparse Scenery



**Figura 57.** Resultado de la métrica *Saved Rebroadcast* para el esquema CBS para Cth=2 RAD=0.001, Cth=3 RAD=0.001, Cth=2 RAD=0.01, Cth=3 RAD=0.01 respectivamente para escenarios densos.